Novi model prepoznavanja ljudskih aktivnosti u proizvodnim procesima primjenom računalnoga vida

Gudlin, Mihael

Sveučilište u Zagrebu

FAKULTET STROJARSTVA I BRODOGRADNJE

Mihael Gudlin

NOVI MODEL PREPOZNAVANJA LJUDSKIH AKTIVNOSTI U PROIZVODNIM PROCESIMA PRIMJENOM RAČUNALNOGA VIDA

DOKTORSKI RAD

Zagreb, 2021.

Sveučilište u Zagrebu

FAKULTET STROJARSTVA I BRODOGRADNJE

Mihael Gudlin

NOVI MODEL PREPOZNAVANJA LJUDSKIH AKTIVNOSTI U PROIZVODNIM PROCESIMA PRIMJENOM RAČUNALNOGA VIDA

DOKTORSKI RAD

Mentor:

Prof. dr. sc. Nedeljko Štefanić

Zagreb, 2021.

Sveučilište u Zagrebu

FAKULTET STROJARSTVA I BRODOGRADNJE

Mihael Gudlin

A NEW MODEL FOR DETECTION OF HUMAN ACTIVITIES IN MANUFACTURING PROCESSES USING COMPUTER VISION

DOCTORAL DISERTATION

Supervisor:

Prof. Nedeljko Štefanić, PhD

Zagreb, 2021.

Podatci za bibliografsku karticu

UDK:	67.02:004
Ključne riječi:	Studij vremena, računalni vid, duboko strojno učenje, prepoznavanje aktivnosti, vremenska segmentacija aktivnosti, proizvodni procesi.
Znanstveno područje:	Tehničke znanosti
Znanstveno polje:	Strojarstvo
Institucija:	Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje
Mentor:	Prof. dr. sc. Nedeljko Štefanić
Broj stranica:	192
Broj slika:	96
Broj tablica:	58
Broj korištenih bibliografskih izvora:	158
Datum obrane:	26.3.2021.
Povjerenstvo:	Prof. dr. sc. Dragutin Lisjak Prof. dr. sc. Nedeljko Štefanić Prof. emer. dr. sc. Ivica Veža (*Sveučilište u Splitu, Fakultet elektrotehnike, strojarstva i brodogradnje*)
Institucija u kojoj je rad pohranjen:	Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje Nacionalna i sveučilišna knjižnica u Zagrebu

Zahvala

Zahvaljujem se mentoru prof. dr. sc. Nedeljku Štefaniću na podršci, razumijevanju i savjetima kod izrade disertacije. Također, zahvaljujem mu na tome što je dopustio da se učim samostalnosti, odgovornosti i važnosti donošenja odluka, ali isto tako i na brzim reakcijama u trenutcima kada sam znao zalutati s puta. Ovo su bile lekcije za cijeli život.

Zahvaljujem se članovima povjerenstva, prof. dr. sc Dragutinu Lisjaku i prof. emer. dr. sc. Ivici Veži, na odvojenom vremenu za konzultacije i čitanje rada te savjetima koji su doprinijeli tome da disertacija bude bolja. Konačno, hvala im i na razumijevanju u završnim fazama izrade rada.

Nadalje, zahvaljujem i poduzeću Klimaoprema d.d., a posebno dr. sc. Robertu Obrazu i Tinu Dorotiću mag. ing. mech za svu pomoć u fazi prikupljanja podataka. Hvala i zaposlenicama koje su nesebično sudjelovale u kreiranju uzorka, bez vas ovo istraživanje ne bi bilo moguće.

Zahvaljujem se svim kolegicama i kolegama sa Zavoda za industrijsko inženjerstvo na podršci i pomoći tijekom cijelog poslijediplomskog studija.

Hvala i izv. prof. dr. sc Hrvoju Cajneru na tome što je uvijek bio spreman odvojiti svoje vrijeme da bi dao odgovore na moja pitanja vezana uz statističke aspekte istraživanja.

Također, hvala kolegi i prijatelju dr. sc. Davoru Kolaru. Naše rasprave o raznim temama i tehničkim detaljima bile su od neprocjenjivoga značaja kod oblikovanja disertacije.

Posebno hvala ide mom kumu i prijatelju doc. dr. sc Miri Hegediću koji je uvijek bio velika potpora i pomoć kod svih vrsta izazova i poteškoća u procesu izrade disertacije.

Iskrenu i veliku zahvalu htio bih uputiti cijeloj svojoj obitelji i rodbini na svakom obliku pomoći tijekom mojeg školovanja. Uvijek sam osjećao vašu podršku na ovom dugačkom putovanju.

Na kraju, najveća zahvala ide mojoj supruzi Mariji te djeci Ivanu i Dori, na strpljenju, razumijevanju i požrtvovnosti. Hvala vam što ste me uvijek razveselili i nasmijali kad je bilo najpotrebnije. Bili ste i jeste, moja snaga i odgovor na pitanje zašto sve ovo ima smisla.

.

Sažetak

Učinkovitost ljudskog faktora će i u budućnosti imati značajan utjecaj na cjelokupni proizvodni sustav, što znači da će tu učinkovitost biti potrebno pratiti i kvantificirati. Metrika bazirana na vremenu izvođenja rada jedan je od važnijih pokazatelja učinkovitosti. Konvencionalni pristupi studiju vremena temeljeni su na promatranju ljudskih operacija, unaprijed definiranim vremenskim standardima i u novije vrijeme video snimkama i njihovoj ručnoj analizi. Nedostaci postojećih metoda studija vremena te tehnološki trendovi poticaj su za istraživanje tehnika umjetne inteligencije s ciljem automatizacije prepoznavanja i procjene trajanja ljudskih aktivnosti u proizvodnim procesima.

Analiza ljudskih aktivnosti iz video zapisa je upravo jedan od problema iz domene računalnog vida, a koji spada u područje umjetne inteligencije. Studij vremena zahtijeva istovremeno prepoznavanje aktivnosti koju subjekt izvodi kao i vrijeme trajanja te aktivnosti. Istovremeno rješavanje ova dva zadatka predstavlja jedan od smjerova istraživanja u području računalnog vida, a koji u kontekstu proizvodnje još uvijek nije dovoljno istražen. Moderni pristupi rješavanju problema istovremenog prepoznavanja i vremenske segmentacije aktivnosti temeljeni su na dubokom strojnom učenju, području iz domene umjetne inteligencije.

Na temelju prethodnih spoznaja kao cilj istraživanja definiran je razvoj modela dubokog strojnog učenja koji će imati sposobnost prepoznavanja i vremenske segmentacije niza ljudskih aktivnosti na temelju video zapisa prikupljenih u proizvodnim procesima. Kako bi ovaj cilj bio ostvaren prikupljen je uzorak iz realnog proizvodnog procesa koji se sastoji od devet radnih aktivnosti. Prilikom snimanja procesa, radne aktivnosti su izvodila četiri subjekta na tri različita tipa proizvoda, dok je samo snimanje izvedeno iz dva različita kadra. Razvijeno je 27 različitih modela koji se razlikuju po pitanju kadra snimanja procesa, vrste ulaznih značajki modela i arhitekture modela odgovorne za finalnu klasifikaciju aktivnosti i vremensku segmentaciju. U radu je predložena i procedura za ocjenu učinkovitosti te usporedbu novih modela. Razvijeni modeli doprinijeti će poslovima industrijskih inženjera, olakšavajući analizu produktivnosti ljudskog faktora u proizvodnim procesima.

Ključne riječi: studij vremena, računalni vid, duboko strojno učenje, prepoznavanje aktivnosti, vremenska segmentacija aktivnosti, proizvodni procesi

Summary

The performance of the human factor will continue to have a significant impact on the overall manufacturing system efficiency, which means that this performance will need to be monitored and quantified. Time-based metrics are viewed as one of the important performance indicators. Conventional approaches to the time study are based on observation of human activities, predefined time standards, and more recently videos and manual analysis of videos. The deficiencies of existing approaches to the time study and technological progress are the motivation for the research of artificial intelligence techniques with the aim of automating action detection in manufacturing processes.

Analysis of human activities from a video is one of the research problems in the domain of computer vision, which belongs to the field of artificial intelligence. Time study requires recognition of the activity performed by the subject and the duration of that activity. The simultaneous solution of these two tasks represents one of the research directions in the field of computer vision, which in the context of production is still not sufficiently investigated. Modern approaches to solving the problem of simultaneous recognition and time segmentation of activities are based on deep learning, an area from the domain of artificial intelligence.

The goal of this research was the development of a deep learning model with the capability of recognition and temporal segmentation of a series of human activities from videos collected in manufacturing processes. To achieve this goal, a sample was collected from the real manufacturing process, which consists of nine work activities. During the video recording of the process, the work activities were performed by four subjects on three different types of products, while the recording itself was performed from two different view positions. 27 different models have been developed which differ with respect to recording viewpoint, model input features, and model architecture responsible for activity classification and time segmentation. In the dissertation, a procedure for evaluating efficiency and comparison of new models is proposed. The developed models will contribute to the work of industrial engineers, facilitating the analysis of human factor productivity in manufacturing processes.

Keywords: time study, computer vision, deep learning, action segmentation, action detection, manufacturing processes

Sadržaj

Popis slika

Popis tablica

Popis oznaka

Oznaka	Značenje oznake
x	Skalar, pisan malim slovom u kurzivu
$\boldsymbol{x}$	Vektor, pisan malim zadebljanim slovom u kurzivu
$\boldsymbol{X}$	Matrica, pisana velikim zadebljanim slovom u kurzivu
$\mathbf{X}$	Tenzor, (3. i višeg reda), pisan velikim zadebljanim slovom
$\boldsymbol{X}^T$	Transponirana matrica $\boldsymbol{X}$
$\boldsymbol{X} \odot \boldsymbol{Y}$	Produkt po elementima, Hadamardov produkt
$\boldsymbol{X} * \boldsymbol{Y}$	Operator konvolucije
$diag(\boldsymbol{x})$	Dijagonalna matrica, elementi dijagonale su određeni s $\boldsymbol{x}$
$\|\boldsymbol{x}\|_p$	L^p norma vektora
$f(\boldsymbol{x}; \boldsymbol{W})$	Funkcija od $\boldsymbol{x}$ parametrizirana s $\boldsymbol{W}$
$J(\boldsymbol{W})$	Funkcija gubitka (engl. cost function)
$L(\boldsymbol{W})$	Funkcija gubitka po primjeru (engl. loss function)
$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{y}}$	Parcijalna derivacija od $\boldsymbol{x}$ s obzirom na $\boldsymbol{y}$, Jacobijeva matrica
$\frac{\partial x}{\partial \boldsymbol{y}}$	Gradijent od x s obzirom na vektor $\boldsymbol{y}$
$\frac{\partial x}{\partial \boldsymbol{Y}}$	Gradijent od x s obzirom na matricu $\boldsymbol{Y}$
$\mathbb{R}$	Skup realnih brojeva
$\{1, 2, \ldots, \mathrm{C}\}$	Skup svih cijelih brojeva od 1 do C
$\mathcal{P}$	Skup uređenih parova primjera i oznaka $(\boldsymbol{x}_i, y_i)$
$\mathbb{E}_{\mathcal{P}}[\boldsymbol{x}]$	Očekivanje od $\boldsymbol{x}$ na skupu podataka $\mathcal{P}$

Napomena: Svi vektori su predstavljeni kao stupac vektori.

Derivacija skalara po tenzoru uvijek poprima dimenziju tenzora. Za sve ostale derivacije pretpostavlja se raspored elemenata po brojniku (engl. *numerator layout*).

Svi ostali simboli i oznake objašnjeni su unutar teksta na mjestu pojavljivanja.

Popis kratica

Kratica	Opis
ADAM	Adaptivna procjena momenta (engl. *adaptive moment estimation*)
ANN	Umjetna neuronska mreža (engl. *artificial neural network)*
biLSTM	Dvosmjerna mreža s dugom kratkoročnom memorijom (engl. *bidirectional long short term memory*)
BP	Algoritam unatražnog prostiranja pogreške (engl. *backpropagation)*
BPTT	Algoritam unatražnog prostiranja pogreške kroz vrijeme (engl. *backpropagation through time*).
CNN	Konvolucijska neuronska mreže (engl. *convolutional neural network*)
CPS	Kibernetsko-fizički sustav (engl. *cyber-physical system*)
DL	Duboko strojno učenje (engl. *Deep learning)*
F1@IoU	Segmentacijska F1 metrika uz minimalan prag preklapanja (engl. *segmental F1 score with IoU*)
FNN	Unaprijedna neuronska mreža (engl. *feedforward neural network)*
FPS	Broj sličica po sekundi kod video zapisa (engl. *frames per second)*
GRU	Povratna ćelija s mehenizmima vrata (engl. *gated recurrent unit*)
HMM	Skriveni Markovljev model (engl. *Hidden Markov models*)
HOG	Histogram orijentacije gradijenata (engl. *Histogram of oriented gradients)*
IID	Pretpostavka nezavisnosti i identične distribucije primjera (engl. *independent and identically distributed)*
IoU	Omjer presjeka i unije (engl. *Intersection over Union*)
LSTM	Mreža s dugom kratkoročnom memorijom (engl. *long short-term memory*)
MLE	Procjena kriterijem najveće izglednosti (engl. *maximum likelihood estimation)*
MLP	Višeslojni perceptron (engl. *multi-layer perceptron)*

Kratica	Opis
OF	Optički tok (engl. *optical flo*w)
PTS	Tehnike bazirane na unaprijed definiranim vremenskim standardima (engl. *predetermined time standards)*
ReLU	Zglobna aktivacijska funkcija (engl. *rectified linear unit)*
RGB	Slikovni kanali tri osnovne boje („RedGreenBlue“)
RMSProp	Algoritam prostiranja korijena srednje vrijednosti kvadrata pogreške (engl. *root mean square propagation*)
RNN	Povratna neuronska mreže (engl. *recurrent neural network*)
SGD	Stohastička gradijentna metoda (engl. *stochastic gradient descent)*
SIFT	Transformacija značajki invarijantna na skaliranje (engl. *scale invariant feature transform)*
STU	Standardizirana učinkovitost modela
SURF	Ubrzane robusne značajke (engl. *speeded-up robust features)*
SVM	Stroj potpornih vektora (engl. *support vector machine)*

1. UVOD

Pojava novih proizvodnih i poslovnih paradigmi ispunjena je očekivanjima o postizanju viših razina operativne učinkovitosti i produktivnosti temeljenih na visokom stupnju digitalizacije i automatizacije. Pitanje koje se nameće kao posljedica takvih predviđanja je kakva će biti pozicija čovjeka u proizvodnim procesima, konkretnije, hoće li njegova uloga biti marginalizirana do te mjere da će biti isključen iz procesa. Koncept Industrije 4.0 je istaknuti predstavnik novih paradigmi, a temeljen je na integraciji strojeva i ljudi unutar složenih procesa podržanih kibernetsko – fizičkim sustavom (CPS) koji je pokretan velikom količinom podataka prikupljenih putem mreže senzora [1,2]. Agilnost tehnološki naprednog sustava ovisit će o vještim i obučenim zaposlenicima [3], pri čemu će doći do promjene u načinu interakcije između strojeva i ljudi [1,4]. Zaključak je da će fleksibilnost koju donosi čovjek i dalje ostati bitan faktor u proizvodnim operacijama.

U nastavku uvodnog dijela rada objašnjena je motivacija za primjenu tehnika računalnog vida temeljenih na dubokom strojnom učenju kako bi se unaprijedio pristup studiju vremena. Napravljen je inicijalni osvrt na otvorene probleme iz područja računalnog vida, a koji su relevantni u kontekstu studija vremena. Postavljena je hipoteza rada te istraživački ciljevi zajedno s očekivanim doprinosima istraživanja. Obrazložene su faze primijenjene metodologije istraživanja te je u konačnici dana struktura doktorskog rada.

1.1 Motivacija

Uvažavajući spomenute pretpostavke o utjecaju ljudskog faktora na proizvodnju, i u budućnosti će biti potrebno kvantificirati ljudsku učinkovitost u različitim procesima, pa tako i onim manualnima. Metrika temeljena na vremenu izvođenja definiranih zadataka dobar je pokazatelj učinkovitosti [5]. Tradicionalni pristupi studiju vremena uključuju primjenu štoperica, ili tehnika baziranih na unaprijed definiranim vremenskim standardima (engl. *Predetermined Time Standards-PTS*) [5], dok se za bilježenje aktivnosti koriste alati poput radnih lista ili video zapisa [6]. Ovi pristupi su neefikasni iz više razloga. U slučaju da je analiza temeljena na manualnom procesu promatranja operacija, ne postoji mogućnost analize u stvarnom vremenu što bi potencijalno moglo pomoći kod planiranja proizvodnih aktivnosti i balansiranja opterećenja. Manualni zapis ograničava obujam (veličina uzorka) i učestalost provedbe analize [7]. U slučaju primjene video zapisa, pokazalo se da je za naknadnu obradu jednoga sata snimke potrebno od dva do pet sati analitičara [6]. Osim što je postupak obilježavanja vremenski

intenzivan, podložan je subjektivnosti istraživača zbog razlika u načinu obilježavanja [6]. Primjena PTS tehnika rješava dio problema, ali zbog primjene uprosječenih vrijednosti [5], ista nije izravna i personalizirana mjera učinkovitosti promatranog subjekta.

Automatizirani pristup prepoznavanju i procjeni trajanja ljudskih aktivnosti u proizvodnim procesima moguće je rješenje nedostataka postojećih metoda. Tehnološki razvoj koji uključuje CPS i automatsko prikupljanje podataka s različitih senzora stvara preduvjete za personalizirano praćenje učinkovitosti resursa [2]. Većina kompanija iz proizvodnog sektora ne koristi podatke na optimalan način, a posebice one nestrukturirane poput video zapisa te im nedostaju rješenja i modeli koji bi olakšali njihovu analizu i interpretaciju [8]. Analiza ljudskih aktivnosti iz video zapisa predstavlja jedan od problema kojim se bavi područje računalnog vida (engl. *computer vision*) koje spada u domenu umjetne inteligencije i strojnog učenja [9,10]. Problem prepoznavanja kontinuiranih ljudskih aktivnosti je temeljni zadatak u mnogim realnim primjenama računalnog vida od video nadzora, zdravstva do interakcije između ljudi i strojeva [11]. Iako određeni istraživači naglašavaju potrebu za primjenom tehnika računalnog vida u kontekstu nadzora i praćenja ljudi u proizvodnji [1,4], prisutan je nedostatak istraživanja koja govore o ovoj problematici u proizvodnji [6].

Do unazad nekoliko godina, klasični algoritmi strojnog učenja imali su dominantan status u domeni prepoznavanja ljudskih aktivnosti [9,10], pri čemu je njihov temeljni nedostatak činjenica da je za postizanje dobrih performansi ključna kvalitetna priprema ulaznih značajki modela, što iziskuje visoku razinu domenskog znanja, ali i značajan ulog vremenskog resursa [12,13]. Nadalje, ovakav pristup je ograničen na specifični problem, odnosno nema mogućnost generalizacije na neki novi problem [9,12]. Trendovi poput povećanja računarskih resursa te velike količine nestrukturiranih podataka omogućili su primjenu dubokih hijerarhijskih modela iz područja koje je danas poznato kao duboko strojno učenje [13]. Prednost ovih modela je u zaobilaženju manualnog izvlačenja ulaznih značajki algoritma kroz hijerarhijsko učenje sve složenijih reprezentacija sirovih ulaznih podataka, dok se naučene reprezentacije nižih razina mogu koristiti kod novih problema [13,14]. Reprezentacija podataka se u ovom kontekstu odnosi na transformaciju podataka, tj. preslikavanje podataka iz jednog prostora u drugi (npr. iz dvodimenzionalnog u trodimenzionalni). Temeljni nedostatak ovakvih modela je potreba za velikom količinom podataka i računalnim resursima te pojava problema ne-konveksne optimizacije i problema razlučivanja faktora varijacije tj. interpretacije modela [14]. Uspjeh metoda dubokog učenja kod prepoznavanja aktivnosti u određenim domenama [11] poticaj je

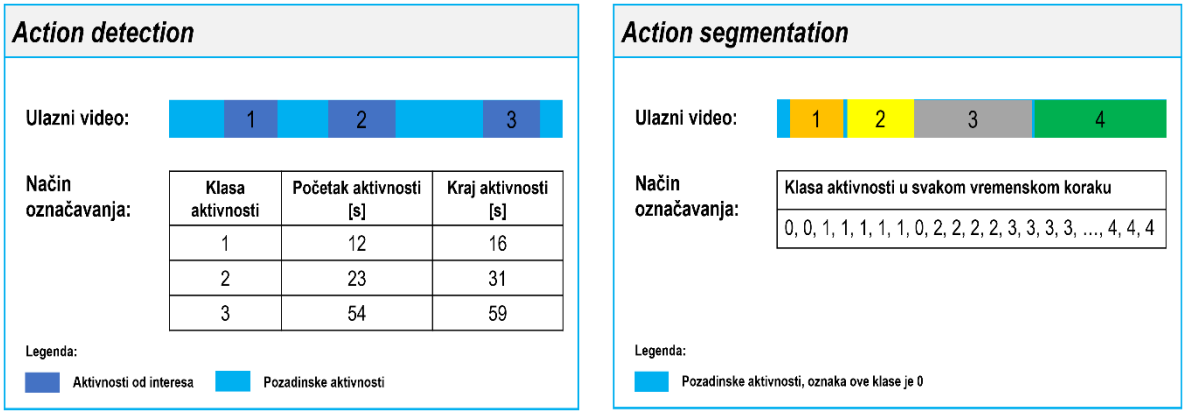
da se istraži primjenjivost i učinkovitost te vrste modela u realnim proizvodnim uvjetima, što je ujedno korak u smjeru automatiziranog praćenja učinkovitosti ljudskog faktora.

1.2 Inicijalne spoznaje o domeni problema

Analiza ljudskih aktivnosti primjenom računalnog vida aktivno je istraživačko područje iz domene umjetne inteligencije, s naglaskom na nekoliko problema. Problem koji je dosad najčešće bio istraživan odnosi se na prepoznavanje aktivnosti (engl. *action recognition / action classification*) [15] – gdje je cilj klasificirati prepoznatu aktivnost u neku od definiranih klasa, dok je ulaz u algoritam unaprijed vremenski segmentiran video (engl. *trimmed video*), koji sadrži samo jednu od ciljanih klasa [16]. Također, kod ovog problema ne razmatra se slučaj da video ne sadrži niti jednu od poznatih klasa tj. da sadrži pozadinsku klasu (engl. *background class*) [17]. Ovaj problem predstavlja zahtjevan zadatak zbog različitih izazova s kojima se robusni algoritmi moraju nositi, poput primjerice: vizualnih smetnji, varijacija u kadru, razlikama u složenosti pozadine, promjenama u osvjetljenju, snimkama iz udaljenosti i onima loše kvalitete, varijacijama unutar iste klase aktivnosti i sličnosti između različitih klasa aktivnosti te interakcijama između više ljudi i objekata [10,11]. Prije pojave dubokog učenja uobičajen tijek prepoznavanja ljudskih aktivnosti sastojao se od koraka detekcije područja interesa primjenom fiksnog algoritma nakon čega su prepoznata područja interesa kombinirana u značajke koje su bile ulaz klasifikacijskih algoritama [9]. U realnim uvjetima rijetko kad je moguće unaprijed znati koje značajke su bitne za promatrani problem, što dovodi u pitanje manualnu pripremu značajki, posebice kod različitih aktivnosti koje mogu varirati u izgledu i vremenu izvođenja, te je stoga prirodnije da se koriste naučene reprezentacije sirovih podataka [18]. Prethodni argument je jedan od razloga zašto je primjena dubokog učenja počela imati sve važniju ulogu u ovoj domeni, pri čemu su u istraživanjima predložene različite arhitekture modela [19,20] i načini reprezentiranja ulaznih podataka [21]. Nužno je napomenuti da je u literaturi prepoznato više načina prikupljanja podataka. Tri glavna načina su: klasične RGB kamere, senzori dubine ili kombinacija ovih dvaju sustava (npr. Microsoft Kinect) [22]. Način prikupljanja podataka izravno utječe na reprezentaciju sirovih podataka, a svaki od njih ima svoje prednosti i nedostatke [10,22] te je u optimalnom slučaju podatke potrebno prikupljati kombinacijom sustava. U literaturi je RGB zapis podataka najčešća pojava [11] te je zbog toga donesena odluka da se i u ovom radu koristi ovaj modalitet prikupljanja podataka. Iako je ovo veoma aktivno područje istraživanja, primjena postojećih modela u realnim uvjetima ostaje problem. U većini istraživanja fokus je na individualnim aktivnostima, a ne interakcijama s drugim ljudima ili objektima [10]. Istraživanja su obično usmjerena na rješavanje problema s

obzirom na određeni skup podataka te uslijed toga ne postoji jedinstveni pristup rješenju problema raspoznavanja aktivnosti [9,10]. Podatci su obično prikupljeni u kontroliranim uvjetima kako bi se izbjegle bilo kakve vizualne smetnje [6]. Glavni nedostatak istraživanja leži u činjenici da je sam problem prepoznavanja aktivnosti umjetno postavljen, jer podrazumijeva postojanje eksternog procesa koji radi vremensku segmentaciju video isječaka tako da oni sadrže samo pojedinačne akcije od interesa za sustav [15,17].

Drugi problem koji je znatno manje zastupljen u literaturi, a s aspekta primjenjivosti u realnim uvjetima je puno važniji, je problem istovremenog prepoznavanja i vremenske segmentacije aktivnosti, u literaturi nazivan kao *„action detection“* ili *„action segmentation“*. Cilj istraživača koji se bave ovim problemom je izgraditi model koji je sposoban na vremenski ne-segmentiranom video zapisu *(*engl. *untrimmed video*), koji u općem slučaju sadrži više od jedne klase aktivnosti (od kojih neke aktivnosti predstavljaju pozadinsku klasu), prepoznati sve aktivnosti i što preciznije definirati početak i završetak pojedine aktivnosti. U slučaju obrade video zapisa koji većinski sadrže pozadinsku klasu cilj je otkrivanje rijetkih segmenata u video zapisu koji sadrže aktivnosti od interesa te se uobičajeno koristi naziv *„action detection“*. S druge strane, za video zapise koji su gusto označeni, odnosno većinu njihova sadržaja čine klase od interesa, zadatak je klasificirati svaku sličicu video zapisa te se u literaturi obično koristi naziv *„action segmentation“* [23].



Slika 1.1 Usporedba zadatka „action detection“ i „action segmentation“

Drugi razlog razlike u nazivima obično je posljedica korištene metrike za ocjenu modela [24]. Zaključak autora ovog istraživanja je da postoji i treći mogući razlog razlika u nazivu, a on je povezan s načinom na koji je formuliran problem nadziranog strojnog učenja, tj. da li je potrebno riješiti problem regresije ili klasifikacije što je ujedno uvjetovano načinom označavanja aktivnosti koje su prisutne u video zapisu. Kod *„action detection“-a* istodobno se rješavaju oba spomenuta problema, jer su početni i završni trenutak pojedine aktivnosti

eksplicitno naznačeni te je dana oznaka klase aktivnosti unutar granica segmenata. *„Action segmentation"* je obično formuliran samo kao problem klasifikacije pri čemu je svaka sličica u video zapisu označena s odgovarajućom klasom aktivnosti te je vrijeme trajanja aktivnosti moguće odrediti implicitno. Do sada su već provođena istraživanja koja se bave primjenom dubokog učenja s ciljem rješavanja problema prepoznavanja i vremenske segmentacije aktivnosti u različitim domenama ljudske djelatnosti [15,16,24]. Analiza literature ukazuje na to da i dalje postoji potreba za novim istraživanjima po pitanju kombiniranja različitih arhitektura te učinkovitosti i posebice točnosti modela u realnim uvjetima.

Pretragom pojmova poput „studij vremena", „analiza radnog tijeka", „strojno učenje" i „video snimka" u kontekstu proizvodnje, pronađen je manji broj istraživanja, pri čemu su izdvojena istraživanja [6,7,18,25,26]. Izdvojeni radovi su analizirani s obzirom na problem koji je u fokusu, vrstu primijenjenog modela i značajki te svojstva korištenih podataka prilikom učenja modela. Sva spomenuta istraživanja osim rada [7], bave se isključivo problemom prepoznavanja aktivnosti, dok Jiang i ostali [7] demonstriraju i način na koji bi se moglo izračunati vrijeme trajanja aktivnosti, iako je i u tome istraživanju glavni fokus prepoznavanje aktivnosti. Primjena klasičnih modela strojnog učenja prevladava u istraživanjima gdje modeli poput skrivenih Markovljevih model (engl. *Hidden Markov models-HMM*) [6,25,26] ili stroja potpornih vektora (engl. *Support vector machine-SVM*) [7] slijede nakon ručnog izvlačenja značajki. Makantasis i ostali [18] su u svom istraživanju primijenili model dubokog učenja temeljen na 2D konvolucijskoj neuronskoj mreži (engl. *convolutional neural network-CNN*) i višeslojnom perceptronu (engl. *multi-layer perceptron-MLP*), međutim zbog toga su kao ulaz korištene manualno kreirane značajke, a ne sirovi video zapisi. Zanimljivo je da se samo rad [6] bavio problemom nenadziranog i polu-nadziranog učenja iako je automatizirano označavanje podataka iznimno bitno za realnu primjenu. Sva istraživanja tvrde da su ulazni podatci bili vremenski ne-segmentirani video zapisi, pri čemu su Jiang i ostali [7] prikupili podatke u laboratorijskim uvjetima, a radovi [18,25,26] su provedeni na skupu podataka opisanom u [27], koji više nije javno dostupan. Rude i ostali [6] su podatke prikupljali isključivo senzorom dubine, dok su svi ostali istraživači koristili RGB kamere.

1.3 Cilj i hipoteza rada

Iz inicijalnih spoznaja o trenutnom stanju područja istraživanja proizašli su sljedeći ciljevi:

1. Na temelju dostupne literature analizirati modele dubokog strojnog učenja koji su dosad primijenjeni u domeni prepoznavanja i segmentacije ljudskih aktivnosti iz video zapisa i izdvojiti one pristupe koji imaju potencijal za primjenu u kontekstu proizvodnih procesa.
2. Prikupiti i obilježiti dovoljnu količinu podataka o izvođenju aktivnosti u okviru realnog proizvodnog procesa, u obliku RGB video zapisa, koja će omogućiti razvoj i testiranje modela dubokog strojnog učenja.
3. Izraditi model dubokog strojnog učenja sposoban za prepoznavanje i vremensku segmentaciju niza ljudskih aktivnosti te analizirati njegovu primjenjivost u realnim proizvodnim uvjetima.

Hipoteza istraživanja je:

Na temelju podataka prikupljenih iz realnog proizvodnog procesa moguće je razviti model temeljen na računalnom vidu i dubokom strojnom učenju koji ima sposobnost prepoznavanja i vremenske segmentacije niza ljudskih aktivnosti te će se njegovom primjenom unaprijediti postupci studija vremena i analize produktivnosti ljudskog faktora.

Očekivani znanstveni doprinosi istraživanja su :

1. Izrada novog modela za istovremeno prepoznavanje i vremensku segmentaciju niza ljudskih aktivnosti u realnom proizvodnom procesu temeljenog na dubokom strojnom učenju.
2. Razvijena procedura za testiranje učinkovitosti modela na prikupljenom skupu podataka koja će omogućiti usporedbu novo razvijenih modela.

1.4 Metodologija i plan istraživanja

U prvoj fazi istraživanja istražena je dostupna literatura usmjerena na problem istovremenog prepoznavanja i vremenske segmentacije ljudskih aktivnosti kako bi se dobio detaljan uvid u postojeće modele iz domene. Kriterij kod odabira članaka za analizu bio je da se za rješavanje problema koristio pristup temeljen na dubokom strojnom učenju te da je skup podataka na kojem je razvijen model bio prikupljen u obliku RGB video zapisa. Iznimka od definiranog kriterija su članci koji se bave istraživanim problemom u kontekstu proizvodnih procesa iz razloga što je broj članaka ovog tipa izrazito malen. Izabrani članci su analizirani na način da

ih se prvo grupiralo s obzirom na značajke rješavanog problema u *„action detection“* ili *„action segmentation“* skupinu pristupa. U nastavku analize svaki članak je klasificiran s obzirom na vrstu korištenog algoritma za pripremu ulaznih podataka te generiranje izlazne predikcije. Analizirane su prednosti i nedostaci pojedinih pristupa sa svrhom donošenja zaključaka o primjenjivosti u kontekstu proizvodnje.

U drugoj fazi definiran je plan prikupljanja uzorka koji je uključivao odabir konkretnog proizvodnog procesa. U sklopu odabranog procesa određen je broj ljudskih subjekata koji će izvoditi aktivnosti iz procesa te broj ponavljanja aktivnosti, kako bi unutar uzorka bila obuhvaćena varijabilnost kod izvođenja aktivnosti uzrokovana ljudskim faktorom. Specificirani broj ponavljanja aktivnosti povezan je s količinom podataka potrebnom za izgradnju modela dubokog strojnog učenja, pri čemu su u obzir uzeta saznanja na temelju analize literature. Podatci su prikupljani u obliku video zapisa primjenom dvije fiksirane RGB kamere odgovarajućih karakteristika prema preporukama iz literature. Razlozi za ovakav postav leže u činjenici da je u fazi modeliranja cilj bio istražiti i utjecaj ulaznih podataka prikupljenih iz različitih kadrova snimanja na učinkovitost modela.

U trećoj fazi provedena je inicijalna priprema i obrada prikupljenih podataka. Ova faza uključivala je manualno označavanje podataka po pitanju tipa prepoznate aktivnosti te trenutka u kojem aktivnost počinje i završava. Nadalje, video zapisi su formatirani u pogledu dimenzija pojedinih sličica (engl. *frame*), broja sličica po sekundi video zapisa (engl. *frames per second-FPS*) i vremena trajanja pojedinačnih video zapisa kako bi bili pogodni za ulaz u modele. Priprema podataka nije uključivala podjelu video zapisa u posebne grupe aktivnosti iz razloga što je cilj bio utvrditi učinkovitost modela na ne-segmentiranim podatcima. Također, iz skupa prikupljenih podataka nisu isključeni podatci koji sadrže vizualne smetnje kako bi se za modeliranje osigurali uvjeti koji su slični stvarnim uvjetima. Napravljena je i statistička analiza cjelokupnog uzorka kako bi se dobio detaljan uvid u karakteristike izabranog procesa i potencijalne izazove u fazi razvoja modela.

U četvrtoj fazi istraživanja razvijeno je 27 različitih tipova modela koji su bili podvrgnuti validaciji i potrebnim korekcijama s aspekta odabira hiperparametara. Svaki model zasebno je naučen na jednom od tri tipa ulaznih podataka s obzirom na kadar snimanja: podatcima prikupljenim kamerom postavljenom iznad radnog mjesta, podatcima prikupljenim s fokusom na ruke izvoditelja aktivnosti te na podatcima koji predstavljaju fuziju video zapisa iz oba kadra. Kako je jedan od ciljeva bio analizirati učinkovitost modela u slučaju da ne koristimo manualno razvijene ulazne značajke bilo je potrebno proces razvoja modela podijeliti u dvije faze: fazu

izvlačenja značajki i fazu provedbe finalne klasifikacije aktivnosti i vremenske segmentacije. Faza izvlačenja značajki napravljena je na pojedinačnim sličicama primjenom tri različita pristupa: korištenjem tehnike prijenosa znanja (engl. *transfer learning*) iz prethodno naučenog modela za obradu slika, sa i bez finog podešavanja (engl. *fine tuning*) na vlastitim podatcima, te primjenom novo razvijenog modela za obradu slika na vlastitim podatcima. Za finalnu klasifikaciju i vremensku segmentaciju također su korištene tri različite vrste modela temeljene na povratnim neuronskim mrežama (engl. *recurrent neural network–RNN*) i 1D konvolucijskim neuronskim mrežama. Ocjena učinkovitosti modela napravljena je na temelju razvijene procedure koja uključuje metriku preporučenu u literaturi i funkcionalna svojstva modela poput broja parametara i vremena potrebnog za konvergenciju modela.

1.5 Struktura rada

U uvodnom dijelu disertacije dana je motivacija za provedeno istraživanje, definirana su otvorena pitanja u domeni istraživanja kroz inicijalni pregled literature. Završno je postavljena hipoteza istraživanja i očekivani znanstveni doprinosi te korištena metodologija.

U drugom poglavlju predstavljena su postojeća saznanja iz analizirane literature. Ukratko je opisan razvoj područja studija vremena i računalnoga vida. Nakon toga slijedi pregled dubokog strojnog učenja koji između ostalog sadrži pregled procesa učenja izabranih arhitektura. Finalno je dana opsežna analiza literature usmjerene na problem istovremenog prepoznavanja i vremenske segmentacije ljudskih aktivnosti iz RGB video zapisa kod kojih se koristio pristup temeljen na dubokom strojnom učenju.

U trećem poglavlju opisane su karakteristike izabranog proizvodnog procesa i načina prikupljanja podataka. Pojašnjeni su kriteriji pripreme i formatiranja video zapisa u uzorku te kriteriji kod označavanja aktivnosti u uzorku.

U četvrtom poglavlju napravljena je statistička analiza prikupljenog uzorka. Početno su prikazani razni deskriptivni statistički pokazatelji, dok su u nastavku provedeni statistički testovi po pitanju razlika u trajanju radnih aktivnosti zbog utjecaja ljudskog faktora i tipa proizvoda.

Peto poglavlje sadrži detaljan opis razvoja modela za prepoznavanje i vremensku segmentaciju aktivnosti te rezultate evaluacije modela. U prvom dijelu objašnjeni su modeli za izvlačenje značajki, a u drugom dijelu modeli odgovorni za finalnu klasifikaciju i segmentaciju. Na kraju je objašnjena procedura za ocjenu učinkovitosti i usporedbu modela.

U šestom poglavlju izneseni su zaključci doneseni na temelju provedenog istraživanja te smjernice za buduća istraživanja.

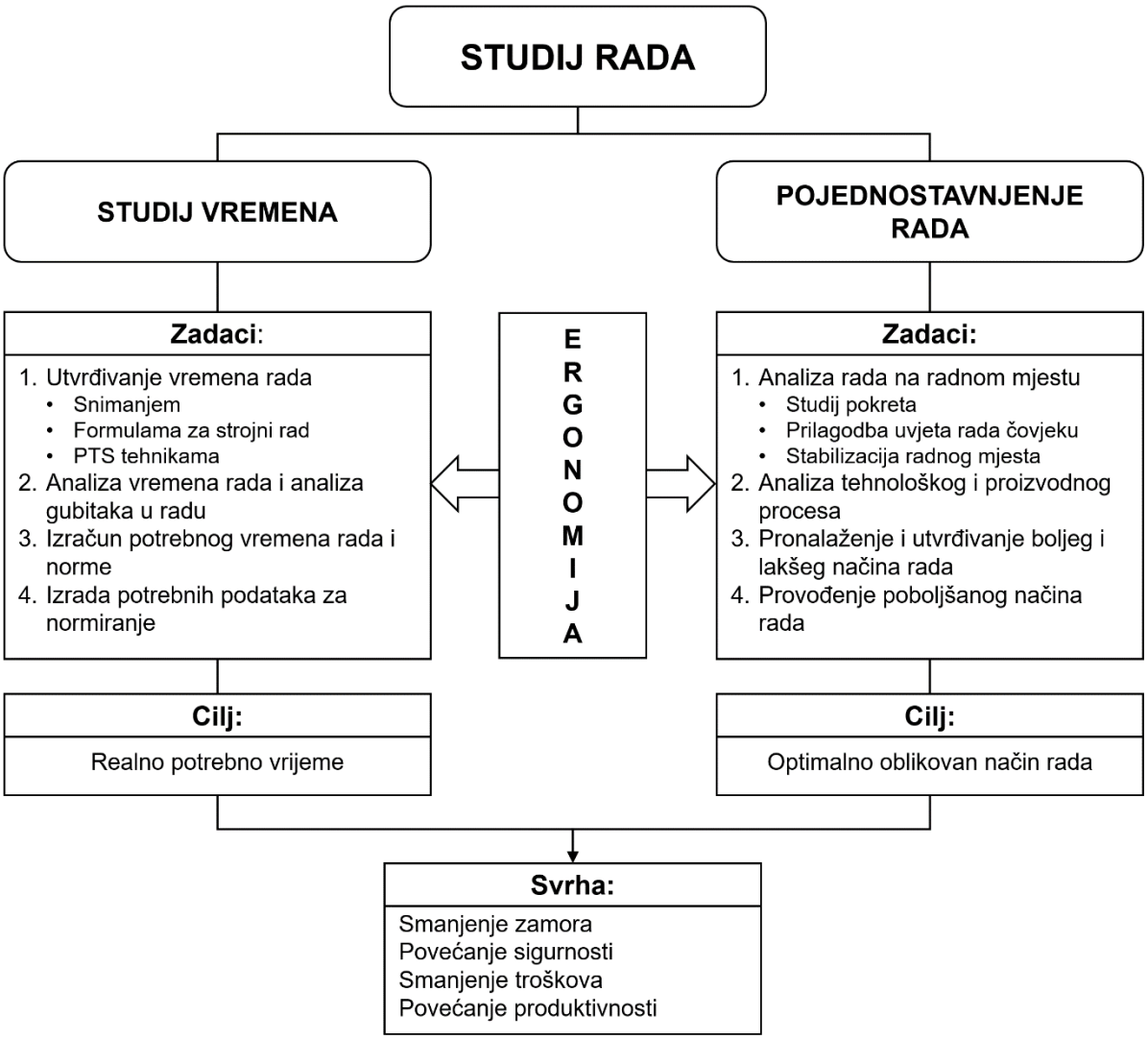
2. PREGLED DOSADAŠNJIH ISTRAŽIVANJA

U području upravljanja i organizacije proizvodnje postoje zahtjevi za povećanjem produktivnosti, smanjenjem proizvodnih troškova te povećanjem brzine procesa i kvalitete. Kako bi bilo moguće povećati efikasnost proizvodnih procesa potrebno je raspoložive resurse koristiti na optimalan način te smanjiti razinu ljudskog zamora. Jedan od najvažnijih resursa je raspoloživo vrijeme za rad, stoga je točno i brzo određivanje vremena potrebnog za rad bitna pretpostavka u organizaciji proizvodnje. Disciplina koja se bavi prethodno spomenutim izazovima je studij rada čiji je zadatak prema [28] da primjenom znanstvenih metoda i sustavnom analizom rada dođe do optimalno oblikovanog načina rada kroz prilagođavanje radnih mjesta, metoda i uvjeta rada čovjeku te do realno potrebnog vremena rada i ispravno izračunate norme koja mora biti organizacijsko mjerilo humano oblikovanog rada.

2.1 Studij vremena

Dvije temeljne komponente studija rada su studij i analiza vremena te pojednostavnjenje rada (vidi sliku 2.1). Pojednostavnjenje rada je usmjereno na unapređenje i olakšanje rada sudionika proizvodnih procesa kroz konkretne promjene operativnih koraka u procesu, pri čemu je fokus na uspostavi najprikladnijeg takta rada [28]. S druge strane, studij vremena predstavlja sustavno praćenje, mjerenje i analizu vremena pojedinih koraka radnog procesa s ciljem određivanja standardnog vremena rada, utvrđivanja svih gubitaka vremena te finalno postizanja poboljšanja kroz uklanjanje gubitaka iz definiranih radnih procedura [29]. Nažalost, uz studij vremena često se vežu negativne konotacije u vidu njegove svrhe koja se pogrešno povezuje samo uz formiranje normativa koji služe kao osnovica za plaćanje radnika te destimulacijom u slučaju da normativi nisu ostvareni [28].

Precizna procjena standardnog vremena rada važna je za operativno planiranje proizvodnje i donošenje ispravnih odluka vezanih za organizaciju proizvodnje. Za ispravnu provedbu studija vremena u obzir moraju biti uzeti okolni faktori poput metoda i sredstava rada te radnih uvjeta u kojima se aktivnosti provode, ali i faktori izravno povezani uz izvršitelja rada poput uvježbanosti, normalnog zalaganja i zamora. Bitna postavka prije provedbe studija vremena je i stabilizacija radnog mjesta koje će biti promatrano, što konkretno znači da je potrebno odrediti način rada te raspored alata i predmeta rada na radnom mjestu. U slučaju da radno mjesto nije stabilizirano, varijacija unutar prikupljenih podataka bit će posljedica različitih faktora, a ne samo ljudskog, stoga će takvi podatci biti neupotrebljivi za izračun standardnog vremena.



Slika 2.1 Elementi studija rada prema [28]

Sustavni pristup praćenju i analizi vremena rada u industriji veže se uz F.W. Taylora koji je u knjizi *„Scientific Management“* [30] definirao temeljne smjernice studija vremena navedene u nastavku:

- Istražiti koji je najbolji način izvršenja rada, podijeliti ga na pojedinačne aktivnosti i za nje odrediti vrijeme.
- Bez proučavanja vremena rada nije moguće dati jasne upute radnicima i uspostaviti organizaciju temeljenu na znanstvenim načelima.
- Grubo mjerenje vremena zasnovano na iskustvenim procjenama potrebno je zamijeniti egzaktnim postupcima.
- Potrebno je utvrditi gubitke u vremenu rada te one koji se mogu ukloniti, a one koji ne mogu uračunati u opravdane gubitke kod definiranja standardnih vremena.

Moguće je kronološki navesti metode za određivanje vremena rada počevši od najstarijih metoda zasnovanih na manualnom snimanju rada (npr. štopericom), nakon kojih su se pojavile metode temeljene na formulama za strojni rad, dok se u novije vrijeme koriste tehnike bazirane na unaprijed definiranim vremenskim standardima u nastavku teksta nazivane PTS tehnike.

Motivacija za razvoj PTS tehnika bio je problem procjene zalaganja izvoditelja rada koji ovisi o subjektivnoj procjeni analitičara pri čemu je najveća teškoća u tome što ne postoje dva radnika koja rade potpuno istom brzinom što znatno otežava procjenu. Kroz analizu rada i izvedenih pokreta istraživači su došli do zaključka da ljudi za izvođenje radnih aktivnosti upotrebljavaju uvijek nekolicinu istih jednostavnih pokreta [31], a kombiniranjem tih osnovnih pokreta mogu obavljati različite poslove. Iz ovog zaključka proizlaze dvije pretpostavke na kojima se temelje PTS tehnike. Prva pretpostavka je da je vrijeme potrebno za izvršenje osnovnih pokreta od strane uvježbanog radnika konstantno, a druga je da ukupno vrijeme rada može biti izračunato sumiranjem vremena svih izvedenih osnovnih pokreta [32]. U literaturi [28] je prepoznato više PTS tehnika poput:

- MTM – *Methods Time Measurement*
- MOST – *Maynard Operation Sequence Technique*
- MODAPTS – *Modular Arrangement of Predetermined Time Standards*
- RTM – *Robot Time Motion*
- WF – *Work Factor System*
- BMT – *Basic Motion Time Study*
- DMT –*Dimensional Motion Times*
- MTA – *Motion Time Analysis*
- MCD – *Master Clerical Data*

Prva poznata metoda koja i danas nalazi široku primjenu je MTM [33], koja se primjenjuje za procjenu manualnih aktivnosti te standardizaciju vremena rada pri čemu se razlikuju tri osnovne varijante ove metode [34]. MTM-1 je namijenjena za analizu radova u masovnoj velikoserijskoj proizvodnji, pri čemu je vrijeme potrebno za analizu oko 300 do 400 puta duže od vremena promatrane operacije. MTM-2 je modifikacija prvog sustava koju je moguće primijeniti na manje repetitivne operacije koje imaju kratka vremena ciklusa, dok je vrijeme analize od 120 do 150 puta duže od vremena promatrane operacije. MTM-3 je razvijen s ciljem brže analize za koju je potrebno oko 50 do 70 puta duže vrijeme od vremena promatrane operacije, međutim ta brzina je ostvarena uz smanjenje točnosti u procjeni u odnosu na MTM-2 metodu. Druga

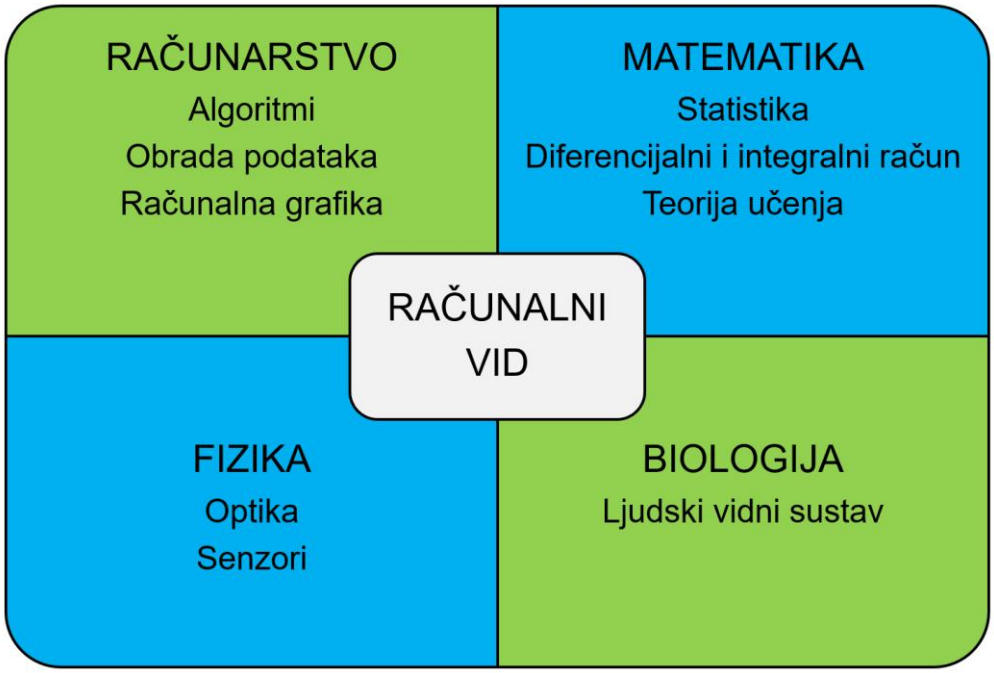
poznata metoda je MOST koja je razvijena od strane „*H.B. Maynard and Company Inc"* još 1960. godine [35]. Motivacija za razvoj ove metode bila je problematika prikupljanja i obrade velike količine podataka potrebne za uspostavu standardnih vremena kod MTM metode [36]. Ključna spoznaja kod razvoja ove metode je bila da kretanje predmeta rada i alata slijedi određene ustaljene obrasce kretanja koji čine nizove radnih pokreta koji su temeljni koncept MOST metode. Prepoznata su tri niza pokreta: osnovni pokreti, kontrolni pokreti i upotreba alata. Ova metoda je deset puta brža od MTM-2 metode. U praksi postoji mnogo prikrivenih i teško uočljivih pokreta koje je potrebno zabilježiti da bi analiza rezultirala ispravnim vremenom, stoga je potrebna dugotrajna edukacija analitičara te praktično iskustvo za korektnu primjenu PTS tehnika. Uz nedostatke navedene u uvodnom dijelu rada, ovo saznanje poslužilo je kao motivacija za istraživanje alternativnih pristupa studiju vremena temeljenih na računalnom vidu.

2.2 Računalni vid

Razvoj sustava koji je sposoban oponašati funkcije ljudskog vida već šezdesetak godina predstavlja jedan od glavnih izazova za istraživače u području umjetne inteligencije. U današnje doba kada glavninu sadržaja na Internetu čine podatci u obliku slika i video zapisa, rješenje prethodnog problema postaje imperativ. Iako je intuicija ranih istraživača bila kako se radi o jednostavnom problemu, te čak postoje anegdote kako je ovaj problem bio zadan kao studentski zadatak tijekom ljetne prakse [37], ovaj problem još uvijek nije ni blizu svog rješenja. Razlozi za pogrešnu intuiciju o težini problema mogu se tražiti u činjenici da ljudi ovaj zadatak ne izvode svjesno te stoga on i ne djeluje pretjerano teško, međutim istraživanja su pokazala da pola od ukupne količine neurona u kori velikog mozga (lat. *Cortex cerebri)* sudjeluje u ljudskoj vizualnoj percepciji [38,39].

Računalni vid predstavlja znanstveno područje koje se bavi rješavanjem prethodno opisanog problema. Konkretnije definirano, uloga ovog područja je razvoj teorijskog znanja o umjetnim sustavima koji imaju sposobnost obrade, analize i razumijevanja slika ili slijeda slika te automatizacija svih zadataka koje može izvoditi ljudski vidni sustav kroz razvoj praktičnih sustava računalnog vida [40]. Računalni vid je interdisciplinarno područje koje obuhvaća znanja iz različitih disciplina, kako je prikazano na slici 2.2, te ga je moguće smjestiti u domenu područja umjetne inteligencije [41]. Nužno je razlikovati područje računalnog vida i područje obrade slika (engl. *image processing*), iako ova područja dijele određene zajedničke metode. Ciljevi područja obrade slika su transformacija slika sa svrhom pripreme za daljnju obradu ili

poboljšanje sadržaja slike kroz npr. uklanjanje šuma, obrezivanje, primjenu rotacija i sličnih operacija koje se obično provode na razini pojedinačnih piksela, pri čemu nije potrebno razumijevanje sadržaja slike [42]. Stoga, područje obrade slika se može smatrati komplementarnim području računalnog vida jer produkt obrade slika može biti ulaz sustava računalnog vida.



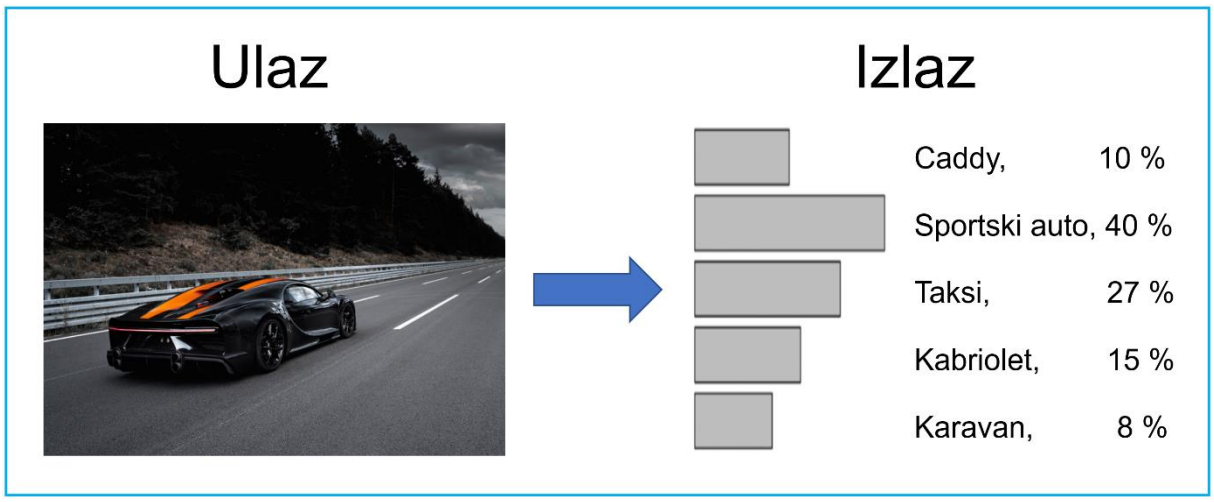
Slika 2.2 Interdisciplinarnost područja računalnog vida

Distribucija slika s kojom se mora nositi ljudski vidni sustav je veoma raznolika uslijed utjecaja različitih faktora poput orijentacije objekata u vidnom polju, osvjetljenja ili određenih vizualnih smetnji, stoga bi se i umjetni sustav vida morao moći nositi s ovim izazovima. Kako bi ovaj problem bio ublažen, u području računalnog vida definiran je čitav niz pojedinačnih zadataka koji odgovaraju zadatcima koje obavlja ljudski vid. Svaki od tih zadataka moguće je pokušati riješiti nezavisno od ostalih, pri čemu je dugoročni cilj kombiniranjem tako razvijenih pojedinačnih komponenti doći do funkcionalnosti koja je bliska ljudskom vidu. U nastavku rada bit će dan pregled najznačajnijih zadataka uz reference na odabrane znanstvene radove te će biti vizualno prikazani očekivani izlazi iz sustava koji rješavaju specifični zadatak. Određene zadatke moguće je riješiti analizirajući pojedinačne sličice, dok je kod drugih nužno analizirati niz slika jer je potreban vremenski kontekst kako bi bilo moguće razumjeti sadržaj. S obzirom na prethodni navod, pregled je grupiran na zadatke koji se izvode na slikama i na zadatke koji se izvode na video zapisima.

2.2.1 Pregled odabranih radova iz područja računalnog vida u obradi slika

Klasifikacija slika (engl. *Image classification*) [43–45]:

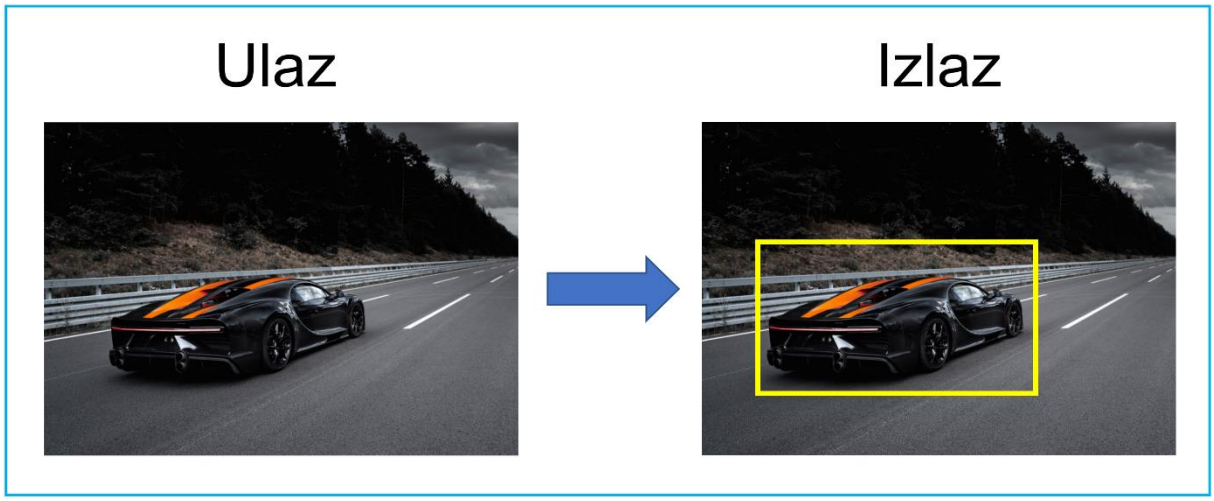
Cilj je dodijeliti odgovarajuću oznaku promatranoj slici iz skupa mogućih oznaka, pri čemu slika obično sadrži jedan objekt na temelju kojeg je moguće razumjeti sadržaj slike.



Slika 2.3 Zadatak klasifikacije slika

Lokalizacija objekata (engl. *Object localization*) [46–48]:

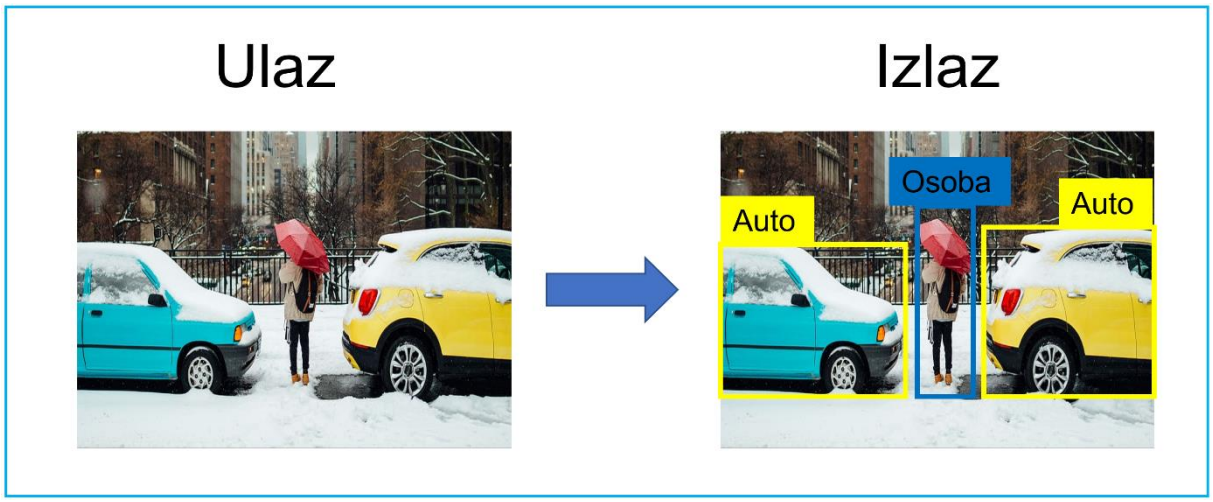
Kod ovog zadatka slike obično sadrže jedan objekt (ili **istu količinu** objekata na **svakoj slici** u skupu podataka) čiju je lokaciju potrebno odrediti. Osim toga, obično je potrebno odrediti i oznaku klase objekta te se tada zadatak naziva klasifikacija i lokalizacija objekta. Problem lokalizacije moguće je izraziti kao regresijski zadatak gdje je cilj procijeniti koordinate granične kutije (engl. *bounding box*) koja je opisana oko objekta.



Slika 2.4 Zadatak lokalizacije objekata

Detekcija objekata (engl. *Object detection*) [49–51]:

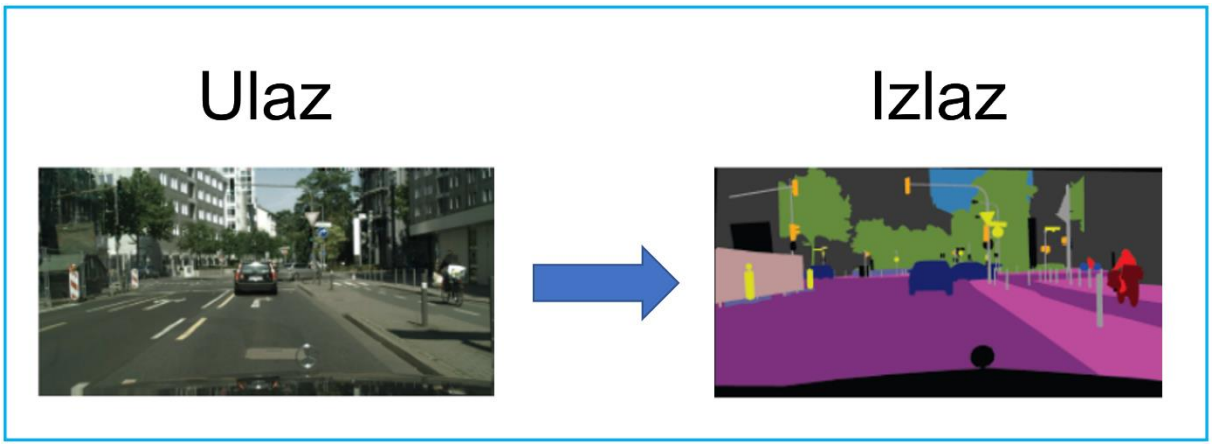
Cilj je istodobna lokalizacija i klasifikacija većeg broja objekata prisutnih na slici, pri čemu broj objekata može varirati na slikama iz promatranog skupa podataka. Također, moguće su situacije kada na slikama nema niti jednog objekta od interesa.



Slika 2.5 Zadatak detekcije objekata

Semantička segmentacija (engl. *Semantic segmentation*) [52–54]:

Cilj je svakom pikselu na slici dodijeliti odgovarajuću oznaku iz skupa mogućih oznaka, tj. svaki piksel svrstati u odgovarajući semantički razred. Ovaj zadatak moguće je shvatiti kao napredniju vrstu detekcijskog problema.



Slika 2.6 Zadatak semantičke segmentacije

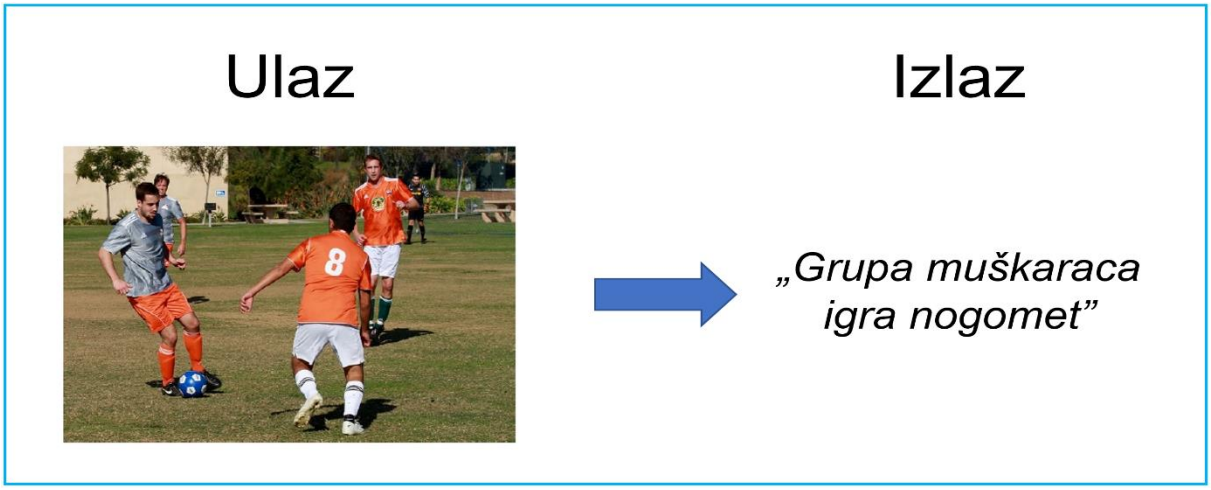
Procjena poze (engl. *Pose estimation*) [55–57]:

Cilj zadatka se može razlikovati ovisno o tome što se promatra. U slučaju da je u pitanju procjena poze čovjeka, tada je cilj odrediti pozicije ljudskih zglobova, a kod krutih objekata cilj je odrediti poziciju i orijentaciju objekta.

Slika 2.7 Zadatak procjene poze

Prevođenje slike u tekst (engl. *Image Captioning*) [19,58,59]:

Cilj zadatka je tekstualno opisati sadržaj prisutan na slici.

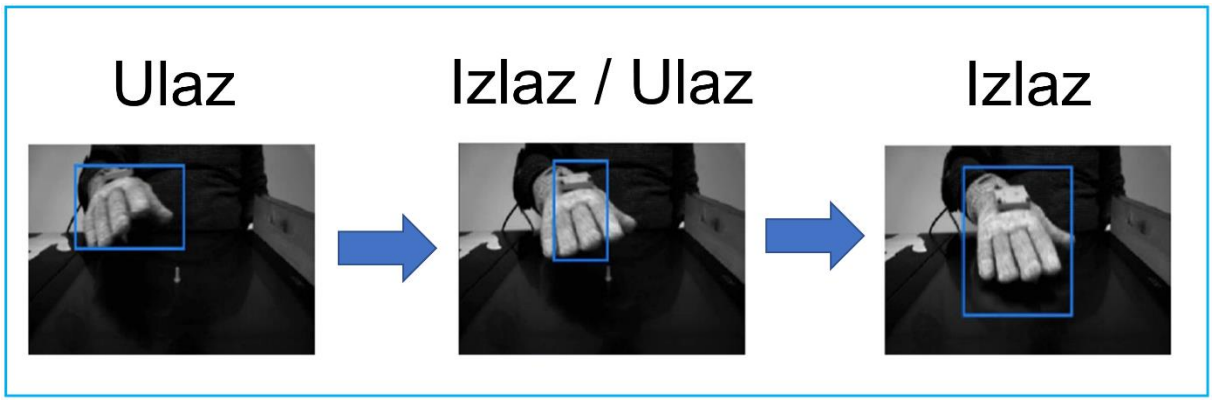


Slika 2.8 Zadatak prevođenja slike u tekst

2.2.2 Pregled odabranih radova iz područja računalnog vida u obradi video zapisa

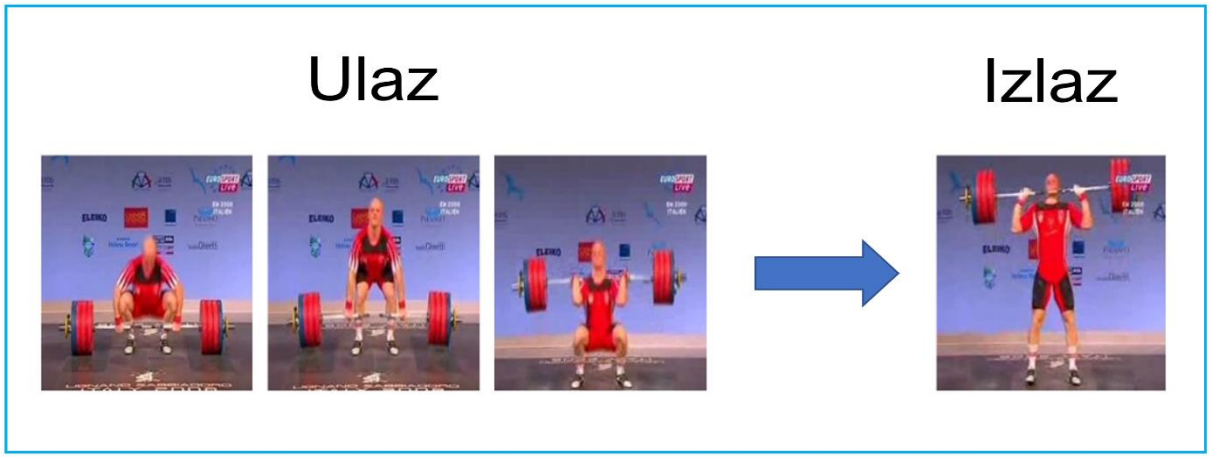
Praćenje objekata (engl. *Object tracking*) [60–62]:

Cilj je pratiti objekt ili više objekata koji obavljaju kretanje u određenom vremenskom intervalu koristeći uzastopne sličice kao ulaz sustava. Ovaj zadatak moguće je rješavati i kao problem detekcije objekata, međutim to je veoma neefikasan pristup. Bolji pristup je koristiti rezultate o lokacijama objekata na prethodnim sličicama kako bi se djelomično predvidjela njihova lokacija na budućim sličicama.



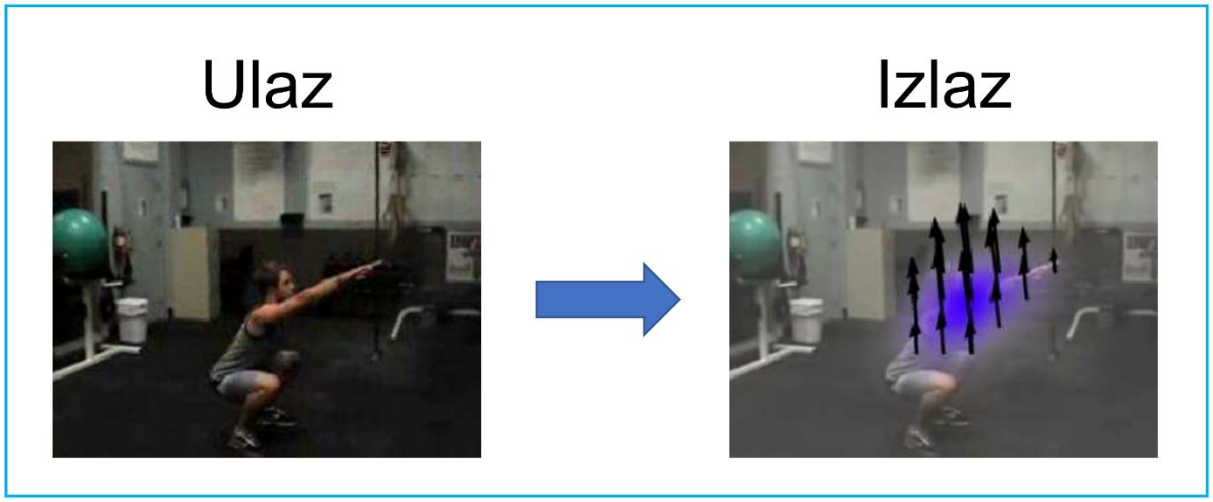
Slika 2.9 Zadatak praćenja objekata

Predikcija sljedeće sličice (engl. *Video prediction*) [63–65]:
Kao što sam naziv kaže, cilj je napraviti predikciju sadržaja budućih sličica u video zapisu. Ovaj zadatak usko je povezan sa zadatkom praćenja objekata. Ovo je korisno kada se npr. želi planirati putanja autonomnog vozila.



Slika 2.10 Zadatak predikcije sljedeće sličice

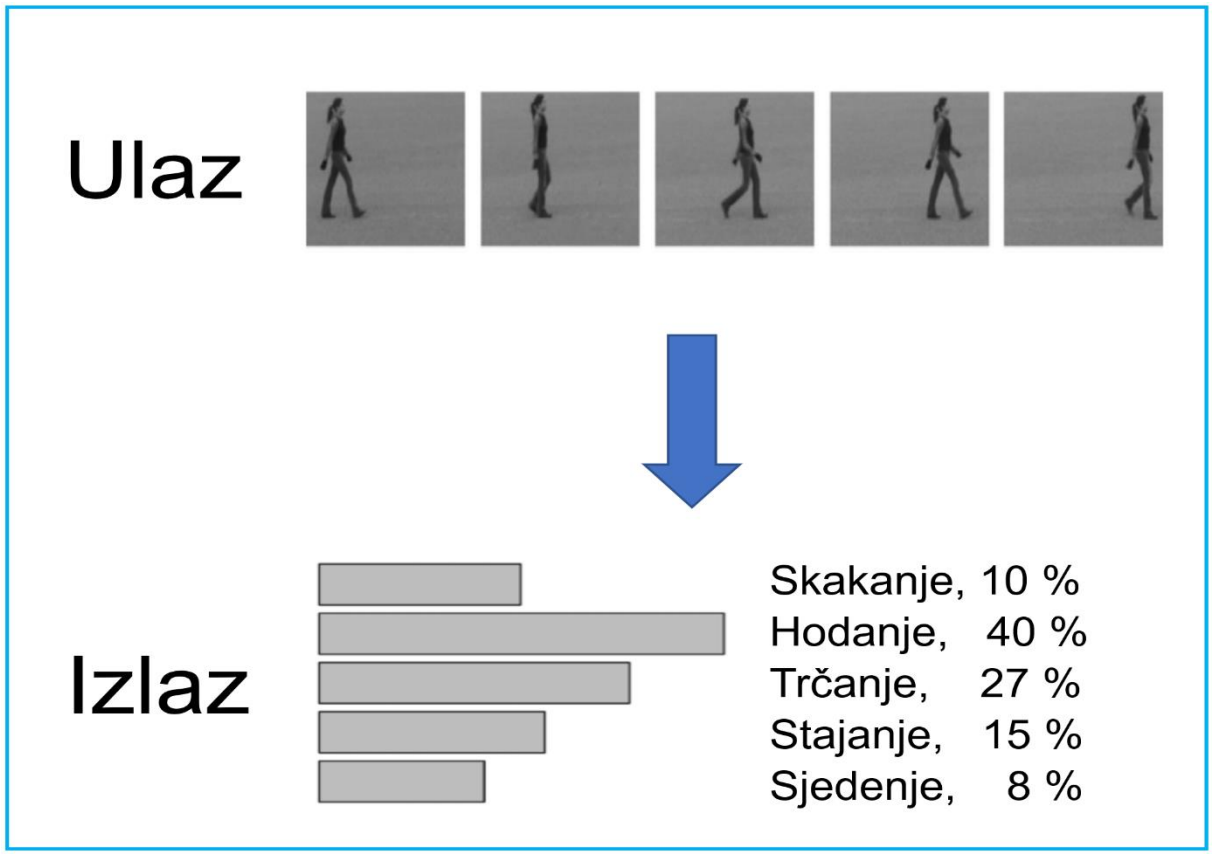
Procjena kretanja (engl. *Motion estimation*) [66–68]:
Za razliku od zadatka praćenja objekata, ovdje nije cilj prepoznati elemente koji se kreću, već je cilj procijeniti brzinu ili putanju objekata. Konkretnije, cilj je odrediti vektore kretanja koji opisuju transformaciju između susjednih sličica. Radi se o teškom problemu jer slike predstavljaju projekciju 3D sadržaja u 2D prostor.



Slika 2.11 Zadatak procjene kretanja

Prepoznavanje aktivnosti (engl. *Action recognition*) [9,10,69]:

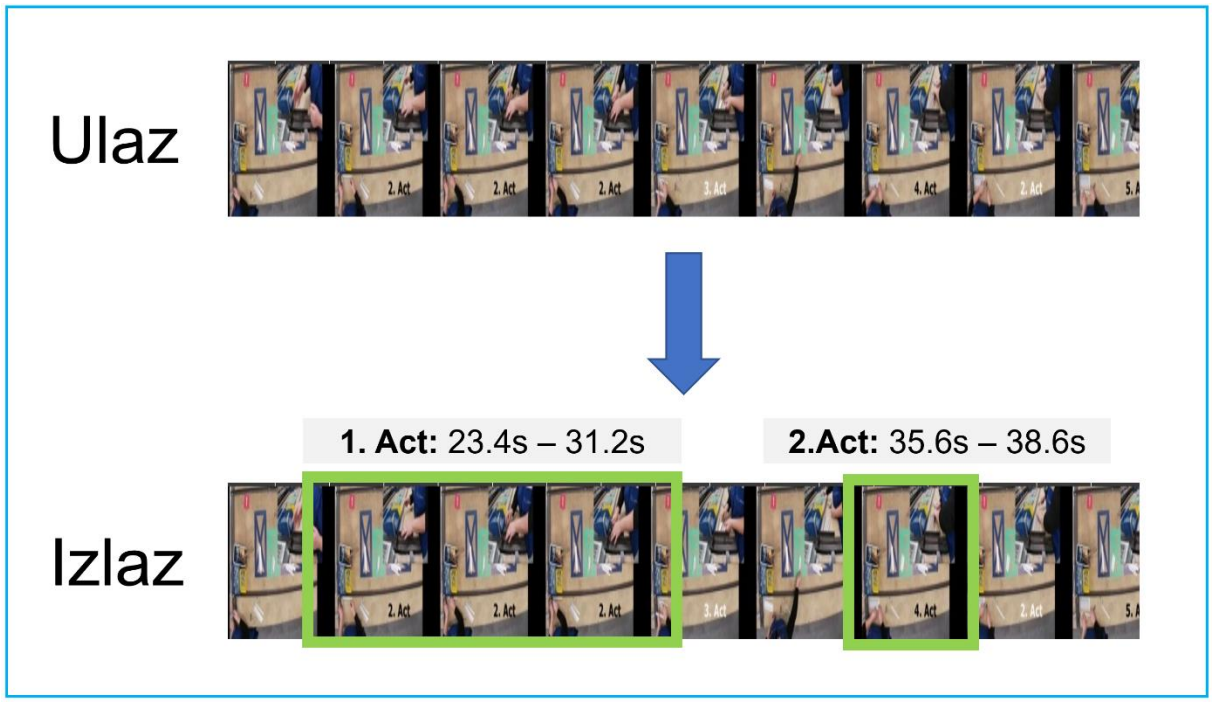
Cilj zadatka je klasificirati prepoznatu aktivnost u neku od definiranih klasa, dok je ulaz u algoritam unaprijed vremenski segmentiran video, koji sadrži samo jednu od ciljanih klasa.



Slika 2.12 Zadatak prepoznavanja aktivnosti

Istovremeno prepoznavanje i vremenska segmentacija aktivnosti (engl. *Action detection/segmentation*) [70–72]:

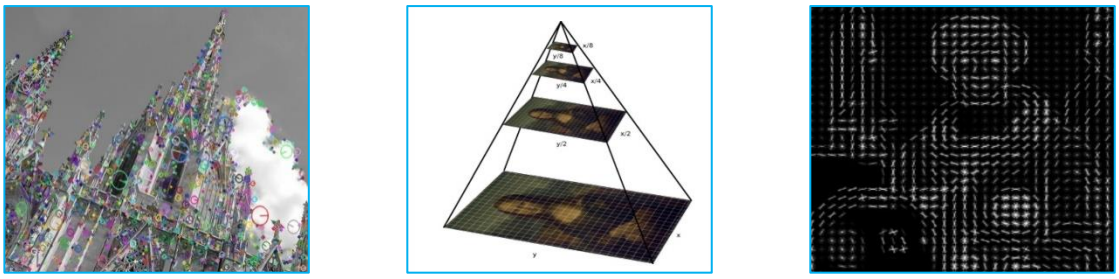
Cilj zadatka je u vremenski ne-segmentiranom video zapisu prepoznati sve aktivnosti i što preciznije definirati početak i završetak pojedine aktivnosti.



Slika 2.13 Zadatak istovremenog prepoznavanja i vremenske segmentacije aktivnosti

Rani pristupi rješavanju ovih zadataka temeljeni su na fiksnim algoritmima. Glavna ideja kod ove vrste pristupa leži u definiranju i izračunu značajki na temelju vrijednosti piksela prisutnih na slici [73]. Sljedeći korak kod ovih pristupa je korištenje izračunatih značajki kao predloška (engl. *template*) za određene vrste objekata od interesa koji mogu biti prisutni na slikama (npr. čovjek, auto, itd.) [10]. Konkretno, u slučaju zadatka klasifikacije slika, na novim slikama se traže značajke koje odgovaraju postojećim predlošcima te u slučaju da je prepoznata dovoljna količina značajki određene vrste objekta, algoritam će zaključiti da je specifičan objekt prisutan na slici i dodijeliti joj odgovarajuću oznaku. Značajke u ovom kontekstu predstavljaju ključne informacije koje je moguće izvući iz sirovih podataka, u konkretnom slučaju slika, te ih je matematički moguće zapisati kao npr. vektor ili matricu. Kvalitetno definirane značajke moraju biti dovoljno deskriptivne da je na temelju njih moguće riješiti odgovarajući zadatak računalnog vida, a istodobno postupak njihovog izvlačenja iz slika mora biti jednostavan i brz [73]. Primjeri osnovnih informacija koje je moguće koristiti kod definiranja značajki u obradi slika su pozicije rubova i kutova te gradijenti teksture. Istraživači su osmislili razne tipove algoritama za definiranje značajki pri čemu su neki od povijesno najznačajnijih:

- Transformacija značajki invarijantna na skaliranje (engl. *Scale invariant feature transform-SIFT*) [74]
- Ubrzane robusne značajke (engl. *Speeded-up robust features-SURF)* [75]
- Histogram orijentacije gradijenata (engl. *Histogram of oriented gradients-HOG*) [76]



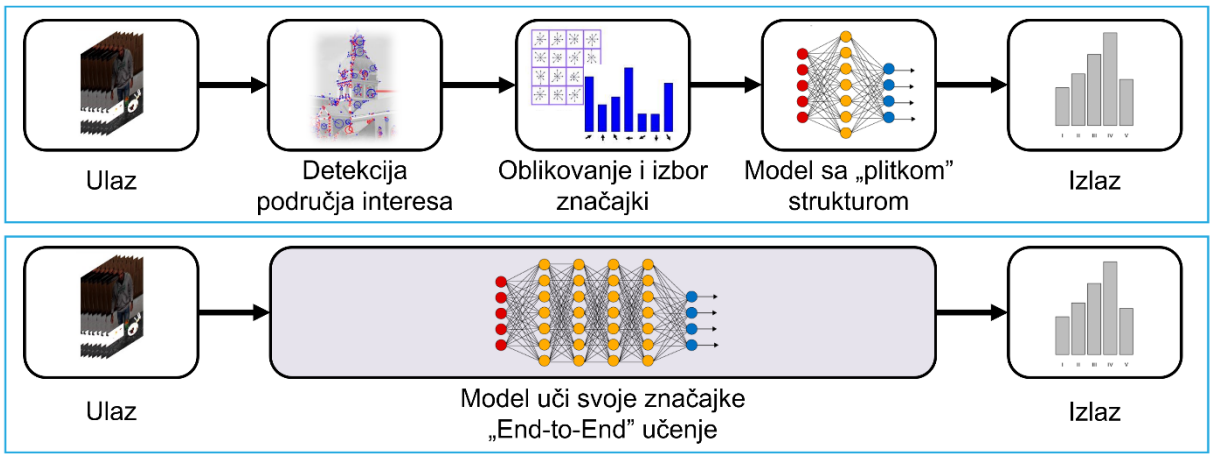
Slika 2.14 Ilustracija izlaza SIFT algoritma[77], piramide slika i izlaza HOG algoritma

Kod SIFT algoritma [74] vizualni objekti su predstavljeni sa značajkama koje su invarijantne na promjene izazvane skaliranjem dimenzija i rotacijama te su robusne na afine transformacije i promjene u kadru. Srž algoritma je u traženju ključnih točaka (engl. *key points*) na slici koje je moguće prepoznati na temelju značajnih promjena u gradijentu, nakon čega se radi izvlačenje okoline (engl. *patch*) svake ključne točke i za nju se računa vektor značajki. Vektor značajki služi za pronalazak sličnih vizualnih objekata na drugim slikama. SURF algoritam [75] slijedi načela SIFT algoritma, npr. primjenjuje strategiju analize slike na različitim dimenzijama - piramidu slika (engl. *multi-resolution image pyramid*), ali je brža i jednostavnija metoda uslijed korištenja aproksimacije Hesseove matrice za traženje ključnih točaka. Histogram orijentacije gradijenata [76] pokazao se kao učinkovita lokalna značajka, pri čemu je osnovna ideja podijeliti sliku u dijelove te izraditi histograme koji govore o učestalosti smjera gradijenata vizualnih elemenata unutar svakog dijela slike. Ovi pristupi rješavanju zadataka računalnog vida imaju nekoliko nedostataka. Prvi nedostatak je potreba određivanja važnih značajki za svaki tip vizualnog objekta od interesa što postaje veoma problematično s rastom broja različitih klasa objekata, pri čemu je odgovornost inženjera koji razvija sustav da odabere broj različitih značajki te njihove karakteristike [73]. Osim toga, vizualni objekti iste klase mogu izgledati sasvim različito na različitim slikama te uslijed toga dijele mali broj zajedničkih značajki što otežava rješavanje složenih zadataka računalnog vida jednostavnim prebrojavanjem zajedničkih značajki između predloška i nove slike.

Korištenje fiksnih pravila onemogućuje skaliranje sustava računalnog vida s rastom količine podataka koja uzrokuje sub-optimalna vremena obrade sustava. Strojnim učenjem nastoje se riješiti navedeni nedostatci, na način da se izbjegne potreba za definiranjem fiksnih pravila, kroz učenje statističkih pravilnosti iz definiranih značajki [78]. Ova paradigma temelji se na

automatiziranom pronalaženju optimalnih parametara matematičke funkcije koja povezuje ulazne podatke u obliku značajki izvučenih iz slika s odgovarajućom odzivnom vrijednosti koja ovisi o zadatku koji je potrebno riješiti. U literaturi [73,78] je navedeno da su izračunate značajke najčešće kombinirane s jednostavnijim modelima strojnog učenja poput linearne regresije, logističke regresije, stabala odluke (*engl. decision trees*) i strojem potpornih vektora za rješavanja problema računalnog vida. Ovaj pristup problemima računalnog vida, iako bolji od pristupa temeljenog na fiksnima pravilima, i dalje nije optimalan jer ovisi o kvaliteti ručno definiranih značajki [13,14] za čiji je razvoj je potrebna znatna količina vremenskog resursa i dobro poznavanje domene problema.

Prijelomni trenutak u domeni računalnog vida najčešće se veže uz uspjeh duboke konvolucijske neuronske mreže AlexNet [79] na ImageNet skupu podataka [80]. Spomenuti model bio je dokaz da je moguće naučiti veći broj hijerarhijski uređenih značajki iz slika te riješiti problem klasifikacije slika na velikom skupu podataka. Ova paradigma nazvana je duboko strojno učenje te njena privlačnost leži upravo u činjenici da omogućava zaobilaženje faze ručnog definiranja značajki koja je bila jedan od nedostataka ranijih pristupa [13,14](slika 2.15). Razlozi uspjeha pristupa temeljenih na dubokom strojnom učenju vezani su uz pojavu dovoljno snažnih računarskih resursa koji omogućuju konvergenciju prihvatljivim rješenjima u razumnom vremenu, raspoloživost dovoljne količine podataka za učenje s kraja na kraj modela (engl. *end-to-end learning*) i razvoj algoritamskih tehnika koje olakšavaju konvergenciju [14]. Kako su i modeli razvijeni u sklopu ove disertacije predstavnici ovog pristupa, u nastavku rada bit će detaljnije opisano područje dubokog strojnog učenja.

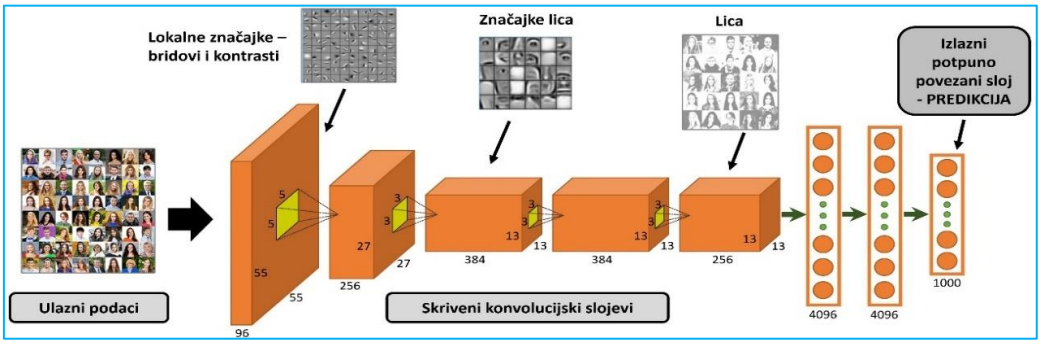


Slika 2.15 Usporedba pristupa strojnom (A) i dubokom učenju (B)

2.3 Duboko strojno učenje

U ovom dijelu rada cilj je uvesti terminologiju koja će biti korištena u nastavku rada te razjasniti zašto se javlja potreba za primjenom dubokog strojnog učenja (engl. *Deep learning-DL*) kod rješavanja problema ljudske percepcije. Namjera je ukazati na razlike u odnosu na druge pristupe strojnom učenju, te strukturirano objasniti temeljne elemente i njihovu poziciju u cjelokupnom algoritmu. Finalno će biti predstavljene i tri najučestalije arhitekture DL-a u primjeni i način odabira pojedinih elemenata algoritma. Kako je područje DL-a veoma opsežno, fokus pregleda bit će samo na nadziranim pristupima učenju (engl. *supervised learning*). Kod ovog pristupa dostupni su podatci za učenje koji se sastoje od primjera (ulaza, opažanja) zapisanih u obliku vektora značajki $\boldsymbol{x}$ te ciljanih izlaza modela y koji se nazivaju oznake (odzivi), koji zajedno čine uzorak $\mathcal{P} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$. U domeni nadziranog učenja problemi su obično formulirani kao zadatci regresije gdje je cilj napraviti predikciju numeričke vrijednosti iz skupa $\mathbb{R}$ za svaki primjer, ili klasifikacije gdje je cilj dodijeliti ispravnu klasu $\mathcal{C} = \{1, 2, \ldots, \mathrm{C}\}$ svakom primjeru. Bitno je naglasiti da je ovo poglavlje inspirirano saznanjima na temelju izvora poput [14,81–84] te se za detaljniju elaboraciju pojedinih termina moguće referirati na ove izvore.

Duboko učenje je grana strojnog učenja koja omogućava istodobno učenje ekstrakcije značajki iz sirovih podataka primjenom niza jednostavnih nelinearnih transformacija te učenje regresije ili klasifikacije koja se provodi na naučenim značajkama. Učenje značajki temeljeno je na pretpostavci da podatci imaju kompozitnu strukturu tj. pretpostavlja se postojanje hijerarhije koncepata (faktora varijacija) u podatcima, pri čemu je svaki koncept definiran pomoću veze s jednostavnijim konceptima [13]. Dobar primjer iz domene računalnog vida je problem klasifikacije slika (vidi sliku 2.16) gdje su početni elementi modela, koje nazivamo slojevi, odgovorni za ekstrakciju jednostavnijih koncepata poput bridova, dok dublji slojevi pokušavaju naučiti apstraktnije koncepte poput npr. objekata koji su kombinacija bridova.



Slika 2.16 Ilustracija učenja hijerarhije koncepata prisutne u podatcima

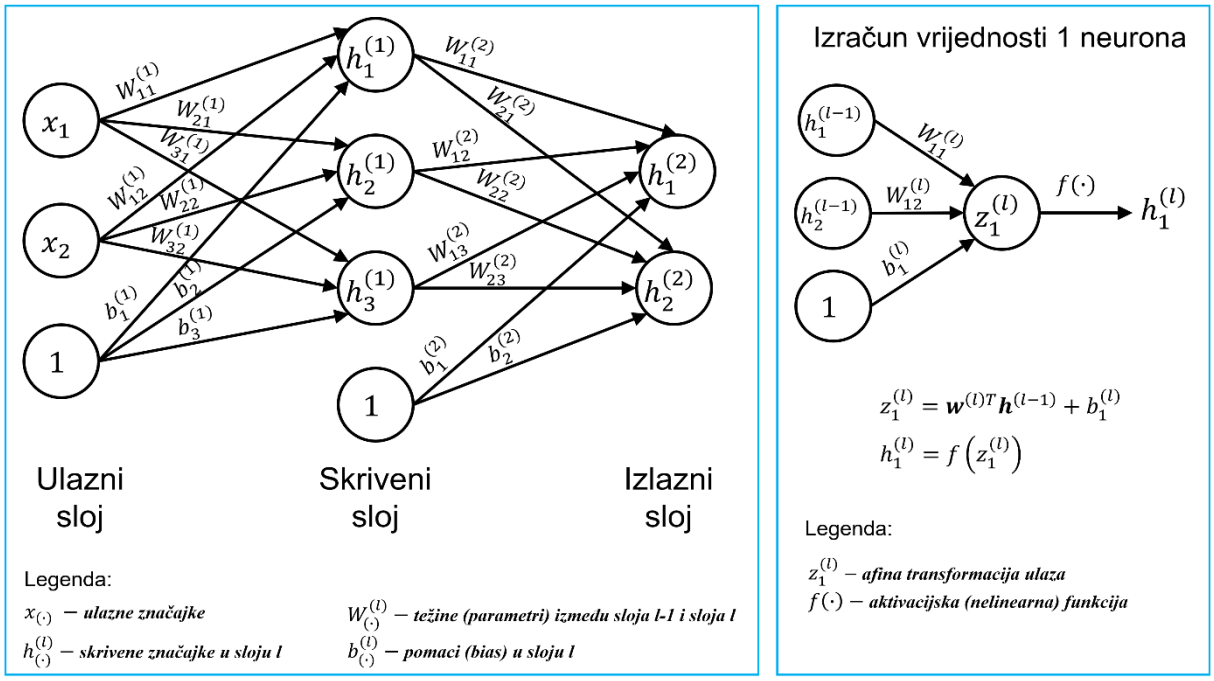
Na temelju napisanog može se zaključiti da se riječ „duboko“ u nazivu „duboko strojno učenje“ odnosi na činjenicu da algoritam koristi veći broj slojeva odgovornih za učenje značajki iz ulaznih podataka. Drugi bitan zaključak vezan je uz pretpostavku kompozitne strukture podataka koja je ispravna u slučaju problema iz domene računalnog vida ili obrade prirodnog jezika, dvije domene na kojima su se algoritmi DL pokazali izuzetno uspješnima. Kod problema iz ovih domena, naučeni modeli moraju biti sposobni zanemariti nevažne varijacije kod ulaznih podataka, npr. kod problema detekcije objekata na slici to može biti promjena u osvjetljenju, orijentaciji, poziciji ili pozadini objekta, a s druge strane moraju biti jako osjetljivi na veoma specifične varijacije kako bi mogli razlikovati slične objekte (npr. knjiga i tablet). Ako se i dalje poslužimo primjerom detekcije objekata, dva slična objekta, ovisno o svojoj poziciji i okolini, mogu izgledati i veoma slično i veoma različito što znatno otežava analizu i obradu na razini pojedinačnih piksela. Upravo ovo je razlog zašto tradicionalni algoritmi strojnog učenja, koji imaju „plitku“ strukturu s obzirom na broj slojeva, ne rade sa sirovim ulaznim podatcima već s kvalitetnim manualno definiranim značajkama koje naglašavaju bitne faktore varijacije, a prigušuju irelevantne. Treći važan zaključak koji proizlazi iz definicije DL-a povezan je s važnosti **istodobnog** učenja niza povezanih koraka ekstrakcije značajki. Kada bismo pokušali naučiti niz reprezentacija ulaznih podataka primjenom većeg broja pojedinačnih plitkih modela strojnog učenja, pri čemu bi izlaz prethodnog modela bio ulaz sljedećeg modela, efekti ostvareni ovakvim pohlepnim algoritmom (engl. *greedy algorithm*) učenja ne bi bili slični onima ostvarenima primjenom algoritma DL-a. Razlog je da optimalna reprezentacija koju je naučio prvi pojedinačni model, ne mora nužno biti i optimalna prva reprezentacija u modelu koji ima primjerice tri ili četiri sloja, stoga je ključno u procesu učenja istovremeno podešavati sve povezane značajke koje se mogu nalaziti u različitim slojevima modela.

S obzirom da je duboko učenje grana strojnog učenja, njeni algoritmi mogu biti analizirani s aspekta tri temeljne komponente algoritama strojnog učenja [85], a to su: model, funkcija gubitka (engl. *cost function*) i optimizacijska metoda. Ukratko, modelom se ograničava skup funkcija koje algoritam može pokušati naučiti, funkcija gubitka usmjerava algoritam u postupku traženja optimalnih parametara modela, dok je za postupak traženja odgovorna optimizacijska metoda.

2.3.1 Model

Model koji se koristi kod algoritama DL-a je umjetna neuronska mreža (engl. *artificial neural network–ANN)*, koja se pod ovim nazivom spominje u ranim istraživanjima [86–88] koja su imala ambiciozan cilj pronalaska matematičke reprezentacije obrade podataka koja je slična

onima bioloških sustava. U svojoj suštini neuronske mreže moguće je interpretirati kao proširenje poopćenih linearnih modela (engl. *generalized linear models*) (npr. logistička regresija), kod kojih je bazne funkcije moguće naučiti iz podataka [83]. Iz ovoga je jasno da ideja dubokog učenja nije nova te je kroz povijest bila prezentirana u okviru različitih paradigmi. Konekcionizam (engl. *connectionism*) [89] je bio naziv paradigme koji se koristio u eri koja je prethodila aktualnoj eri dubokog učenja te je ovaj pokret zaslužan za nekoliko velikih otkrića [90]. Razlog zašto se ovi modeli nazivaju „mrežama" je zato što ih se može prikazati u obliku usmjerenih acikličkih grafova (engl. *directed acyclic graph*) (slika 2.17). Osnovni elementi ANN-a su „neuroni" (u slici 2.17 prikazani kao čvorovi) te oni predstavljaju pojedine značajke izračunate kod učenja modela. Neuroni zatim bivaju organizirani u slojeve, pri čemu se sloj koji sadrži neurone koji predstavljaju ulazne podatke naziva ulaznim slojem. Sljedeći slojevi, koji sadrže značajke dobivene postupkom nelinearnih transformacija neurona prethodnih slojeva (slika 2.17, desno), nazivaju se skrivenim slojevima. Završni sloj u kojem se provodi regresija ili klasifikacija naziva se izlaznim slojem. Veze između neurona u susjednim slojevima (u slici 2.17 prikazani kao bridovi) nazivaju se težine (engl. *weights*), odnosno parametri modela, koji se podešavaju u procesu učenja. Struktura mreže koja je određena brojem neurona i slojeva te načinom povezivanja neurona naziva se arhitekturom modela. O arhitekturama će biti više riječi u kasnijim odjeljcima gdje će biti razrađene specifične arhitekture.



Slika 2.17 Jednostavna potpuno povezana neuronska mreža

Svaki sloj ($l = 1,2, \dots, L$) predstavlja funkciju koja je kombinacija afine transformacije i nelinearne (aktivacijske) funkcije, pri čemu je u općem slučaju model DL-a „duboka“ kompozicija funkcija prema izrazu (2.1).

$$f(\boldsymbol{x}, \boldsymbol{W}) = f^{(L)}\left(f^{(L-1)}\left(\cdots\left(f^{(1)}\left(\boldsymbol{x}, \boldsymbol{W}^{(1)}\right), \cdots\right), \boldsymbol{W}^{(L-1)}\right), \boldsymbol{W}^{(L)}\right) \tag{2.1}$$

Klase funkcija koje algoritam može pokušati naučiti predstavljaju reprezentacijski kapacitet[1] modela, koji je u slučaju dubokog učenja izravno povezan s arhitekturom neuronske mreže. Iako su teorijska istraživanja pokazala da dovoljno velika neuronska mreža može reprezentirati bilo koju funkciju [91], to ne znači da je algoritam može i naučiti jer efektivni kapacitet algoritma ovisi i o funkciji gubitka i o optimizacijskoj metodi te stoga postupak traženja optimalnih parametara može zaglaviti i prije njihovog pronalaska. Pretpostavka je da postoji funkcija $y = f^*(\boldsymbol{x})$ kojom je objašnjena stvarna veza između $\boldsymbol{x}$ i y stoga je cilj učenja pronaći funkciju $\hat{y} = f(\boldsymbol{x}, \boldsymbol{W}^*)$ koja će biti najbolja aproksimacija stvarne funkcije za svaki mogući primjer $\boldsymbol{x}$, što je ekvivalentno traženju optimalnih parametara jer oni u potpunosti određuju funkciju.

2.3.2 Funkcija gubitka

Informaciju o razlikama između izlaza modela $\hat{y}$ i stvarnih oznaka y moguće je dobiti na temelju funkcije gubitka. Prvotno se kod modela neuronskih mreža za funkciju gubitka koristilo isključivo srednje kvadratno odstupanje (engl. *mean squared error*). Danas se funkcije gubitka kod algoritama DL-a uglavnom izvode na temelju kriterija najveće izglednosti (engl. *maximum likelihood estimation-MLE*) parametara modela. Razlog za ovu algoritamsku promjenu je otkriće da modeli koji u izlaznom sloju kao aktivacijsku funkciju koriste sigmoidnu ili softmax funkciju imaju problema sa sporim učenjem u slučaju korištenja srednjeg kvadratnog odstupanja kao funkcije gubitka [14]. S obzirom da se kod algoritama DL najčešće radi o minimiziranju funkcije numeričkim postupcima, funkcija gubitka se definira na temelju negativne log-izglednosti $\mathcal{L}$ (2.2).

$$\mathcal{L}(\boldsymbol{W}; \mathcal{P}) = -\sum_{i=1}^{N} \ln p(y_i | \boldsymbol{x}_i; \boldsymbol{W}) \tag{2.2}$$

U izrazu (2.2) $p(y_i | \boldsymbol{x}_i; \boldsymbol{W})$ je funkcija uvjetne distribucije vjerojatnosti oznake y_i uz zadani primjer $\boldsymbol{x}_i$ za model parametriziran težinama $\boldsymbol{W}$, pri čemu je pretpostavljeno da su podatci

[1] Sinonim za riječ „kapacitet“ u ovom kontekstu je složenost, u smislu da je složeniji model onaj koji ima veći kapacitet

$\mathcal{P} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ međusobno nezavisni i identično distribuirani. Funkcija distribucije vjerojatnosti p u jednadžbi (2.2) usko je povezana s odabirom aktivacijske funkcije izlaznog sloja. Kod regresije se u izlaznom sloju nalaze značajke izračunate afinom transformacijom bez dodatne aktivacijske funkcije. U slučaju klasifikacije, a ovisno o tome da li se radi o binarnoj ili višeklasnoj, nalaze se značajke na koje je primijenjena sigmoidna funkcija (2.3) ili njena generalizacija – softmax funkcija (2.4), iz razloga što ove funkcije preslikavaju ulaz u raspon (0, 1) što se može interpretirati kao vjerojatnost pojedine klase.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.3}$$

$$softmax(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_{k=1}^{C} e^{z_k}}, i \in \{1, 2, \dots, \mathrm{C}\}, \boldsymbol{z} \in \mathbb{R}^C \tag{2.4}$$

Kada se rješava problem regresije, pretpostavka je da se izlazne vrijednosti modela pokoravaju normalnoj (Gaussovoj) distribuciji uslijed čega se korištenjem jednadžbe (2.2), te zanemarivanjem aditivnih i multiplikativnih konstanti, kao gubitak dobije kvadratno odstupanje (2.5) između oznaka y_i i izlaza modela $\hat{y}_i$.

$$J(\boldsymbol{W}; \mathcal{P}) = \frac{1}{2}\sum_{i=1}^{N}\big(y_i - f(\boldsymbol{x}_i, \boldsymbol{W})\big)^2 = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.5}$$

Za problem višeklasne klasifikacije pretpostavlja se kategorijska razdioba preko izlaznih vrijednosti što rezultira gubitkom koji se kolokvijalno naziva gubitak unakrsne entropije (2.6), gdje su oznake u obliku binarnih vektora $\boldsymbol{y}_i$ koji sadrže C elemenata $y_i^{(k)} \in \{0,1\}$ uz $\sum_{k=1}^{C} y_i^{(k)} = 1$, a za vektor izlaznih vrijednosti modela $\hat{\boldsymbol{y}}_i \in \mathbb{R}^C$ vrijedi $\sum_{k=1}^{C} \hat{y}_i^{(k)} = 1$.

$$J(\boldsymbol{W}; \mathcal{P}) = -\sum_{i=1}^{N}\sum_{k=1}^{C} y_i^{(k)} \ln \hat{y}_i^{(k)} \tag{2.6}$$

Notacija $J(\boldsymbol{W}; \mathcal{P})$ ukazuje da je gubitak funkcija parametara pri čemu su podatci fiksirani iz čega slijedi da je problem učenja modela, tj. traženja optimalnih parametara $\boldsymbol{W}^*$, moguće definirati kao minimizaciju funkcije gubitka (2.7), pri čemu funkcija argmin vraća vrijednost argumenata koji minimiziraju funkciju.

$$\boldsymbol{W}^* = \underset{\boldsymbol{W}}{\operatorname{argmin}} J(\boldsymbol{W}; \mathcal{P}) \tag{2.7}$$

Iz razloga što množenje funkcije gubitka s konstantom ne mijenja rješenje optimizacijskog problema iz (2.7), funkcija gubitka obično se množi s $1/N$ (N je veličina uzorka) pa je onda

funkcija gubitka, koristeći jednadžbu negativne log-izglednosti (2.2), ekvivalentna unakrsnoj entropiji između distribucije oznaka i distribucije izlaza modela prema jednadžbi (2.8).

$$J(\boldsymbol{W};\mathcal{P}) = -\frac{1}{N}\sum_{i=1}^{N}\ln p(y_i|\boldsymbol{x}_i;\boldsymbol{W}) = \frac{1}{N}\sum_{i=1}^{N}L(y_i, f(\boldsymbol{x}_i,\boldsymbol{W})) = \mathbb{E}_{\mathcal{P}}[L(y, f(\boldsymbol{x},\boldsymbol{W}))] \qquad (2.8)$$

U jednadžbi (2.8) $L(y_i, f(\boldsymbol{x}_i,\boldsymbol{W}))$ je gubitak po pojedinačnim primjerima (engl. *loss function*), stoga slijedi da je vrijednost $J(\boldsymbol{W};\mathcal{P})$ jednaka očekivanom gubitku $\mathbb{E}_{\mathcal{P}}$ na skupu podataka $\mathcal{P}$.

2.3.3 Optimizacijska metoda

Zbog cjelovitosti pregleda bitno je ukazati na dva svojstva koja razlikuju problem učenja od standardnog problema optimizacije, iako ona nisu specifična za algoritme dubokog učenja već vrijede za bilo koji algoritam strojnog učenja. Prethodno je spomenuto da je cilj učenja pronaći najbolju aproksimaciju stvarne funkcije $y = f^*(\boldsymbol{x})$ za svaki mogući $\boldsymbol{x}$, a ne samo za one koji su u skupu $\mathcal{P}$ koji je korišten za učenje modela. Sposobnost algoritma da odredi ispravnu oznaku y za primjer $\boldsymbol{x}$ iz skupa neviđenih primjera naziva se generalizacija. Pretpostavka generalizacije je da su primjeri korišteni za učenje i novi neviđeni primjeri proizašli iz iste distribucije vjerojatnosti te da su međusobno nezavisni jedni od drugih (engl. *independent and identically distributed–IID*). Činjenica zbog koje se učenje razlikuje od optimizacije je da spomenuta zajednička distribucija vjerojatnosti nije poznata, ali uz IID pretpostavku moguće je procijeniti očekivani gubitak na novim primjerima na temelju gubitka izračunatog na poznatim podatcima $\mathcal{P}$, rječnikom statistike, aritmetička sredina uzorka prema izrazu (2.8) dobra je procjena nepoznatog parametra populacije. U domeni strojnog učenja ova procjena se obično provodi na način da se skup podataka $\mathcal{P}$ podjeli u dva zasebna skupa, skup za učenje $\mathcal{P}_{train}$ na kojem se traže optimalni parametri modela i skup za testiranje $\mathcal{P}_{test}$ kojim se procjenjuje sposobnost generalizacije. Druga bitna razlika između optimizacije i učenja je ta da su problemi strojnog učenja često povezani s određenom metrikom učinkovitosti koja predstavlja stvarni cilj učenja, ali tu metriku nije moguće izravno optimirati, npr. zato jer funkcija kojom je opisana metrika nije derivabilna ili je njena optimizacija težak problem. Primjer metrike je točnost klasifikacije (engl. *accuracy*), koja predstavlja proporciju ispravno klasificiranih primjera u cijelom uzorku, umjesto koje se optimira zamjenska funkcija gubitka na temelju jednadžbe (2.8) uz pripadajuću distribuciju vjerojatnosti $p(y|\boldsymbol{x})$ stvarnih oznaka uz zadane primjere. Ranije spomenuta derivabilnost funkcije gubitka izuzetno je bitna u kontekstu optimizacije algoritama DL-a zato jer se ona provodi primjenom različitih varijanti gradijentnog spusta (engl. *gradient descent*). Gradijentni spust je iterativna optimizacijska metoda prvog reda koja koristi negativni

gradijent funkcije s obzirom na odabrane parametre kako bi pronašla vrijednost parametara u kojima se postiže minimum funkcije. Metoda radi na način da se proizvoljno izabere početna vrijednost parametara, konkretno kod algoritama DL-a slučajno se inicijaliziraju početne vrijednosti matrice težina $\boldsymbol{W}$, a zatim se za tu točku ($\boldsymbol{W}$) izračuna vrijednost gradijenta funkcije gubitka J nakon čega se radi mali pomak u smjeru suprotnom od gradijenta, pri čemu je veličina pomaka određena pozitivnom konstantom α koja se u kontekstu strojnog učenja naziva stopa učenja (engl. *learning rate*). Stopa učenje ima značajan utjecaj na konvergenciju metode, u slučaju da je premala model sporo uči, a ako je prevelika metoda može divergirati. Ažuriranje vrijednosti težina između iteracija, označenih s τ, pokazano je u jednadžbi (2.9).

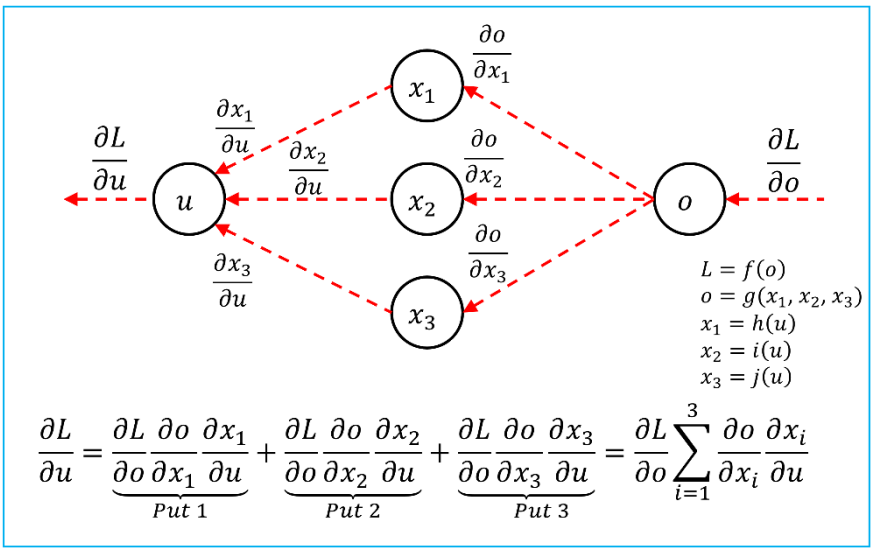
$$\boldsymbol{W}^{(\tau+1)} = \boldsymbol{W}^{(\tau)} - \alpha \frac{\partial J}{\partial \boldsymbol{W}} \tag{2.9}$$

U općem slučaju prethodno opisani postupak se ponavlja sve dok vrijednost gradijenta nije jednaka nuli ili barem blizu nule, tj. dok vrijednost gubitka nije jako mala. U slučaju konveksnih funkcija metoda garantira pronalazak globalnog optimuma, dok u općem slučaju barem lokalnog minimuma. Funkcija gubitaka dubokih neuronskih mreža je nekonveksna zbog korištenja niza nelinearnih transformacija, uslijed čega postoji velik broj lokalnih minimuma, sedlastih točaka (engl. *saddle points*) i platoa[2] zbog čega je teško očekivati pronalazak globalnog optimuma. Ovo je jedan od razloga zašto su algoritmi DL dugo bili ignorirani od strane industrije i većinskog dijela znanstvene zajednice, te su modeli s plitkim strukturama povezani s konveksnim funkcijama gubitka dominirali. Međutim, teorijska i praktična istraživanja pokazala su da je pronalazak lošeg lokalnog optimuma rijetko problematičan jer postoji jako velik broj lokalnih optimuma, ali i sedlastih točaka, u kojima je vrijednost funkcije gubitka približno ista tako da je svejedno u kojem od njih algoritam zaglavi [92]. Nadalje, postupak učenja obično se zaustavlja i prije nego gradijent bude blizu nule, što je još jedna razlika u odnosu na klasičnu optimizaciju. Spomenuta razlika može biti posljedica činjenice da se kod učenja koristi metrika učinkovitosti koja će postići zadovoljavajuću vrijednost prije nego li je zamjenska funkcija gubitka postigla optimum, ili zato jer je učenje zaustavljeno nakon što je utvrđena prenaučenost modela. Termin prenaučenosti bit će objašnjen kasnije.

Još jedan od razloga zašto je područje dubokog učenja, odnosno njene ranije verzije, dugo stagniralo povezano je s time da nije postojala efikasna metoda za izračun gradijenta funkcije gubitka dubokih modela. Algoritam koji se dominantno koristi za pronalazak gradijenta

[2] Područje funkcije gdje je gradijent gotovo jednak nuli

funkcije gubitka je algoritam unatražnog prostiranja pogreške (engl. *backpropagation–BP*) [90] koji se zasniva na lančanom pravilu deriviranja i algoritmu dinamičkog programiranja. Primjena lančanog pravila je logična s obzirom da je model kompozicija funkcija te se cjelokupni gradijent funkcije gubitka s obzirom na parametre, može dobiti kroz izračun suma produkta lokalnih derivacija pojedinih slojeva, pri čemu računanje produkta derivacija kreće od izlaznog sloja prema ulazu. Ovo je moguće prikazati grafom sa slike 2.18, gdje je ukupni gradijent funkcije L s obzirom na čvor u, dobiven sumiranjem produkata lokalnih derivacija pri čemu se sumira po putevima između ulaznog čvora u i izlaznog čvora o.



Slika 2.18 Ilustracija izračuna gradijenta kroz jedan sloj

U općem slučaju, kada model ima više od jednog sloja, broj puteva između promatranog čvora i izlaza modela raste eksponencijalno s dubinom modela. Konkretno, u slučaju potpuno povezanih slojeva, a uz pretpostavku jednakog broja čvorova u svakom sloju, broj puteva između čvora x i y jednak je izrazu $(broj\ čvorova)^{(broj\ slojeva\ između\ čvora\ x\ i\ čvora\ y)}$. Zbog navedenih svojstava, naivna primjena multivarijantnog lančanog pravila deriviranja nije adekvatna. Opisani problem u okviru BP algoritma riješen je primjenom optimizacijske metode dinamičkog programiranja [93]. Dva svojstva koja opravdavaju primjenu dinamičkog programiranja na neki problem su postojanje optimalne podstrukture i preklapajućih podproblema. Optimalna podstruktura znači da je zadani optimizacijski problem moguće riješiti kombiniranjem optimalnih rješenja podproblema, pri čemu je ovo svojstvo obično moguće opisati rekurzijom. Ovo svojstvo je prisutno kod modela DL zbog kompozicijske strukture jer se kombiniranjem lokalnih derivacija radi izračun ukupnog gradijenta po pojedinom parametru. Preklapajući problemi se mogu vidjeti na primjeru istih lokalnih derivacija koji se ponavljaju u

različitim produktima (isti član na različitim putevima kroz graf), pa umjesto da ih se svaki puta iznova računa, moguće ih je pohraniti u memoriju. Ovaj se pristup u okviru dinamičkog programiranja naziva memoizacija (engl. *memoization*). Iako dio znanstvene zajednice smatra da je potrebno pronaći algoritam koji ima bolje biološko opravdanje te je istodobno i efikasniji od BP-a [94], a već i postoje alternative koje pokazuju kompetitivne rezultate [95–97], gradijentni spust s BP algoritmom je trenutno dominantan pristup učenja modela DL-a.

Velik broj parametara (obično više od $1 \cdot 10^6$) koji je prisutan u modelima DL-a nije jedini izazov kod učenja gradijentnom metodom. Izračun gradijenta funkcije gubitka s obzirom na težine ovisi i o broju primjera N u skupu podataka $\mathcal{P}$ (2.10), pri čemu je ovaj broj obično veći od $1 \cdot 10^5$.

$$\frac{\partial J}{\partial \boldsymbol{W}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\partial L(y_i, f(\boldsymbol{x}_i, \boldsymbol{W}))}{\partial \boldsymbol{W}} = \mathbb{E}_{\mathcal{P}}\left[\frac{\partial L}{\partial \boldsymbol{W}}\right] \tag{2.10}$$

Izračun gradijenta na svakom primjeru prema izrazu (2.10) je računarski intenzivna operacija zbog algebarske složenosti funkcije gubitka, te je u praktičnom smislu uglavnom neizvediva zbog memorijskih ograničenja računala. Moguće rješenje je stohastička gradijentna metoda (engl. *stochastic gradient descent–SGD*), koja koristi činjenicu da je gradijent iz izraza (2.10) očekivanje koje je moguće procijeniti i na manjem slučajno uzorkovanom skupu primjera $\mathcal{B} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{B}$, za koji vrijedi $\mathcal{B} \subset \mathcal{P}$ i $B \ll N$, prema izrazu (2.11).

$$\left(\frac{\partial J}{\partial \boldsymbol{W}}\right)_{Batch} = \frac{1}{B}\sum_{i=1}^{B}\frac{\partial L(y_i, f(\boldsymbol{x}_i, \boldsymbol{W}))}{\partial \boldsymbol{W}} \tag{2.11}$$

Skup primjera $\mathcal{B}$ obično se naziva grupa (engl. *batch*) te se ažuriranje definirano u izrazu (2.9) radi nakon svake grupe opažanja. Određeni autori koriste nazivi grupni gradijentni spust (engl. *batch gradient descent*) kada je veličina grupe $\mathcal{B}$ veća od jedan. Izbor veličine grupe $\mathcal{B}$ utječe na preciznost procjene gradijenta, što je grupa veća procjena je bolja, pri čemu izbor najčešće ovisi o memorijskom ograničenju. Dobro svojstvo SGD pristupa je prisutnost varijabilnosti koja je posljedica procjene gradijenta što omogućava bijeg iz lošeg lokalnog optimuma. S druge strane, ta varijabilnost dovodi do toga da kod SGD metode, za razliku od obične gradijentne metode, gradijent nikada ne pada na nulu, zato je potrebno kroz iteracije smanjivati stopu učenja kako bi metoda konvergirala. Primjer linearnog smanjenja stope učenja prikazan je u (2.12).

$$\alpha_\tau = \alpha_0\left(1 - \frac{\tau}{T}\right) \tag{2.12}$$

U jednadžbi (2.12) α_0 je početna stopa učenja, α_τ je stopa učenja u trenutnoj iteraciji, τ je oznaka trenutne iteracije, T je ukupan broj iteracija. Učenje primjenom dosad navedenih inačica gradijentne metode može biti veoma sporo. Kao odgovor na ovaj problem u literaturi su predložene naprednije metode. Dio predloženih metoda nastoji problem spore konvergencije riješiti kroz uvođenje algoritma momenta (engl. *momentum*) u gradijentnu metodu [98]. Metode s momentom ubrzavaju učenje na način da algoritam akumulira informacije o vrijednosti gradijenta u prethodnim iteracijama, npr. primjenom eksponencijalno smanjujućeg pomičnog prosjeka (engl. *exponentially decaying moving average*), te na temelju dodatnog parametra povezuju vrijednost gradijenta u aktualnoj iteraciji s akumuliranim vrijednostima. Druga grupa metoda pokušava adaptivno upravljati stopom učenja iz razloga što se radi o komponenti koja ima najveći utjecaj na uspjeh učenja, a predstavnici ovih metoda su RMSProp (engl. *root mean square propagation*) i ADAM (engl. *adaptive moment estimation*) metoda [99]. Uzimajući u obzir sve opisane izazove učenja modela DL s gradijentnim metodama, nameće se pitanje zašto ne koristiti metode optimizacije drugog reda? Metode drugog reda koriste i informacije o zakrivljenosti funkcije gubitka na temelju Hesseove matrice, međutim spomenuta matrica je razlog zašto takve metode ne skaliraju dobro s veličinom modela jer je njena veličina m^2, gdje je m broj parametara modela, a invertiranje takve matrice ima složenost m^3.

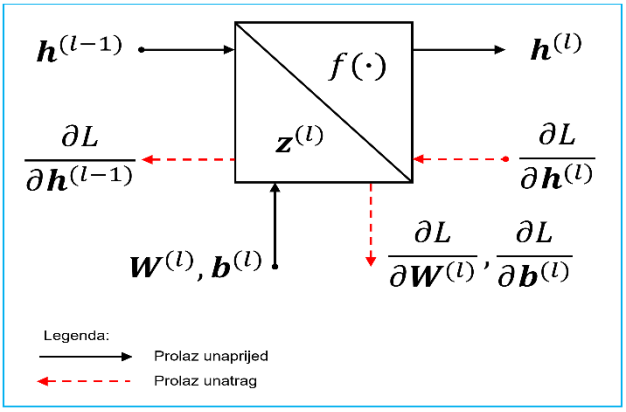
2.3.4 Izabrane arhitekture modela dubokog strojnog učenja

Kako je struktura modela upravo ono što najočitije izdvaja algoritme dubokog strojnog učenja od ostalih algoritama strojnog učenja, u sljedećem odlomku bit će prezentirane tri najčešće spominjane arhitekture u literaturi. Radi se o unaprijednim, konvolucijskim i povratnim neuronskim mrežama. Opisana su njihova temeljna svojstva i područja primjene, uz matematički izvod prolaza unaprijed i unatrag pojedinačnog primjera kroz jedan sloj arhitekture.

Unaprijedna neuronska mreža

Opći oblik dubokih neuronskih mreža naziva se unaprijedna neuronska mreža (engl. *feedforward neural network-FNN)*, te kako je već spomenuto može se tumačiti kao proširenje poopćenih linearnih modela, pri čemu su bazne funkcije naučene iz podataka. Drugi nazivi pod kojima se ova arhitektura javlja su potpuno povezana neuronska mreža (engl. *fully connected neural network*), iz razloga što je svaki neuron u sloju l povezan sa svim neuronima iz prethodnog sloja $l-1$ i sljedećeg sloja $l+1$, te višeslojni perceptron (engl. *multi-layer perceptron*) iz povijesnih razloga. U općoj formulaciji ovaj model se može shvatiti kao

kompozicija funkcija što je prethodno prikazano u jednadžbi (2.1), te model može biti predstavljen grafom (vidi sliku 2.17) u kojem informacije teku od ulaznih značajki do izlaza, bez povratnih petlji, iz čega dolazi i naziv „unaprijedna“ mreža. U praksi se ovakve arhitekture obično primjenjuju na podatke u obliku matrica (2D tenzora). Osnovni element ove arhitekture je potpuno povezani sloj (engl. *fully connected layer*), prikazan vektorskim računskim grafom na slici 2.19.



Slika 2.19 Računski graf jednog sloja unaprijedne neuronske mreže

Sloj predstavlja grupu neurona, dok neuroni predstavljaju pojedinačne značajke iz čega slijedi da je sloj zapravo grupa značajki. Povezivanjem proizvoljnog broja slojeva moguće je oblikovati model željene dubine. Svaki sloj u potpunosti je opisan matematičkim operacijama koje izvodi prilikom prolaza unaprijed (engl. *forward pass*) i prolaza unatrag (engl. *backward pass*). U prolazu unaprijed sloj kao ulaz prima vektor značajki prethodnog sloja $\boldsymbol{h}^{(l-1)} \in \mathbb{R}^{D_{l-1}}$, matricu težina $\boldsymbol{W}^{(l)} \in \mathbb{R}^{D_l \times D_{l-1}}$ i vektor pomaka $\boldsymbol{b}^{(l)} \in \mathbb{R}^{D_l}$ koji su potrebni za izračun afine transformacije $\boldsymbol{z}^{(l)} \in \mathbb{R}^{D_l}$ prema izrazu (2.13).

$$\boldsymbol{z}^{(l)} = \boldsymbol{W}^{(l)} \boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)} \tag{2.13}$$

Matrica težina $\boldsymbol{W}^{(l)}$ povezuje neurone sloja $l-1$ s neuronima sloja l, dok vektor pomaka $\boldsymbol{b}^{(l)}$ osigurava dodatnu fleksibilnost transformacije, u smislu da izračunata funkcija ne mora striktno prolaziti kroz ishodište. Kapacitet modela se dodatno povećava na način da se na rezultat izraza (2.13) primjenjuje nelinearna transformacija primjenom aktivacijske funkcije f, koja je definirana tako da djeluje na svaki element vektora $\boldsymbol{z}^{(l)}$ zasebno (engl. *elementwise*), tj. vrijedi $f(\boldsymbol{z})_i = f(z_i)$. Rezultat primjene funkcije f na $\boldsymbol{z}^{(l)}$ je vektor značajki $\boldsymbol{h}^{(l)} \in \mathbb{R}^{D_l}$ prema izrazu (2.14), koji se zatim prosljeđuje sloju $l+1$.

$$\boldsymbol{h}^{(l)} = f\left(\boldsymbol{z}^{(l)}\right) \tag{2.14}$$

Ovaj postupak se ponavlja za svaki sloj pri čemu vrijedi $\boldsymbol{h}^{(0)} = \boldsymbol{x}$, tj. ulaz modela je vektor značajki primjera, te $\boldsymbol{h}^{(L)} = \hat{\boldsymbol{y}}$, odnosno izlaz zadnjeg sloja je vektor predikcija modela $\hat{\boldsymbol{y}}$. Izlaz modela se zatim uspoređuje s vektorom stvarnih oznaka $\boldsymbol{y}$ na temelju funkcije gubitka po primjeru $L \in \mathbb{R}$. U prolazu unatrag zadatak algoritma unatražnog prostiranja pogreške je izračun gradijenta funkcije gubitka L po matrici težina $\boldsymbol{W}^{(l)}$ i vektoru pomaka $\boldsymbol{b}^{(l)}$ koji su potrebni za učenje vrijednosti težina i pomaka u optimizacijskom postupku. S obzirom da prolaz unatrag kreće iz zadnjeg sloja prema ulazu modela, sloj l će u prolazu unatrag iz sloja $l + 1$ kao ulaz dobiti gradijent gubitka po izlaznom vektoru $\partial L/\partial \boldsymbol{h}^{(l)}$, iz čega primjenom lančanog pravila deriviranja, ali uz napomenu da je iskorištena struktura jednadžbi kako bi se izbjeglo eksplicitno formiranje Jacobijeve matrice[3] i tenzora 3. reda, slijedi gradijent gubitka po matrici težina (2.15) te pomaku (2.16).

$$\frac{\partial L}{\partial \boldsymbol{W}^{(l)}} = \left(\frac{\partial L}{\partial \boldsymbol{h}^{(l)}} \odot f'\left(\boldsymbol{z}^{(l)}\right) \right) \boldsymbol{h}^{(l-1)\mathrm{T}} \tag{2.15}$$

$$\frac{\partial L}{\partial \boldsymbol{b}^{(l)}} = \frac{\partial L}{\partial \boldsymbol{h}^{(l)}} \odot f'\left(\boldsymbol{z}^{(l)}\right) \tag{2.16}$$

U izrazima (2.15) i (2.16), operator $\odot$ predstavlja množenje po pojedinačnim elementima (Hadamardov produkt). $f'\left(\boldsymbol{z}^{(l)}\right)$ je rezultat derivacije $\partial \boldsymbol{h}^{(l)}/\partial \boldsymbol{z}^{(l)}$, i općem slučaju to je Jacobijeva matrica, ali s obzirom na to da je aktivacijska funkcija f definirana tako da djeluje na pojedinačne elemente vektora $\boldsymbol{z}^{(l)}$, ta matrica je dijagonalna te se može zamijeniti vektorom i ulančiti s ostalim izrazima primjenom Hadamardovog produkta. Korištenje strukture problema je vidljivo i kod zadnjeg člana izraza (2.15) gdje je $\boldsymbol{h}^{(l-1)\mathrm{T}}$ dobiven iz $\partial \boldsymbol{z}^{(l)}/\partial \boldsymbol{W}^{(l)}$, koji je u općem slučaju tenzor 3. reda.

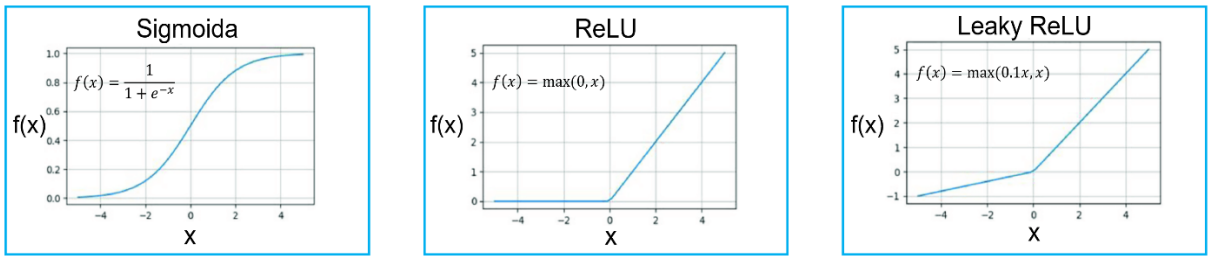
Osim navedenih gradijenata, potrebno je izračunati i gradijent gubitka po ulazu u sloj l, $\partial L/\partial \boldsymbol{h}^{(l-1)}$ prema izrazu (2.17) koji se zatim prosljeđuje sloju $l - 1$, kako bi algoritam BP-a mogao nastaviti s izračunom gradijenta za sve prethodne slojeve.

$$\frac{\partial L}{\partial \boldsymbol{h}^{(l-1)}} = \boldsymbol{W}^{(l)\mathrm{T}} \left(\frac{\partial L}{\partial \boldsymbol{h}^{(l)}} \odot f'\left(\boldsymbol{z}^{(l)}\right) \right) \tag{2.17}$$

Postupak izračuna gradijenata nastavlja se sve dok se ne stigne do ulaza modela, nakon čega optimizacijski postupak provodi ažuriranje svih težina i pomaka. Prolazi unaprijed i unazad

[3] Jacobijeva matrica dimenzija $m \times n$ rezultat je derivacije funkcije $f: \mathbb{R}^n \to \mathbb{R}^m$

izmjenjuju se sve do postizanja uvjeta zaustavljanja algoritma koji može biti uvjetovan ostvarenim rezultatom na metrici učinkovitosti ili ograničenjem u vidu maksimalnog broja iteracija učenja. Jedan od problema koji se javlja u procesu učenja dubokih modela je nestanak gradijenta, što onemogućava učenje. Kao jedan od uzroka ovog problema prepoznata je primjena sigmoidne aktivacijske funkcije (slika 2.20, lijevo) koja ulazi u područje zasićenja (izrazito mali gradijent) kada su njen ulaz jako negativne ili jako pozitivne vrijednosti. Danas se zato kao nelinearnost koristi zglobna aktivacijska funkcija (engl. *rectified linear unit–ReLU*) (slika 2.20, sredina), ili njene generalizacije (slika 2.20, desno) s obzirom da je ReLU funkcija jednaka nuli u negativnom dijelu domene, što opet može dovesti do problema kod učenja. Rezultati su pokazali [100] da primjena ReLU funkcije znatno ubrzava konvergenciju kod dubokih modela.



Slika 2.20 Primjeri aktivacijskih funkcija

Osim izmjene s aspekta aktivacijskih funkcija, za rješavanje problema nestanka gradijenta predložena je i tehnika normalizacije grupe (engl. *batch normalization*) [101] koja se u praksi implementira kao specijalni sloj koji se može dodati nakon afine transformacije ili nakon aktivacijske funkcije. Ova tehnika razvijena je s ciljem smanjenja kovarijantnog pomaka, koji je definiran kao razlika u distribuciji izlaza sloja između dvije iteracije procesa učenja uzrokovana procesom ažuriranja težina modela, a smatra se da usporava konvergenciju modela. Glavna ideja je normalizirati ulaze sloja te nakon toga naučiti optimalnu aritmetičku sredinu i standardnu devijaciju za te iste ulaze. Prvi korak algoritma je procijeniti vektor aritmetičkih sredina $\boldsymbol{\mu}_B \in \mathbb{R}^D$ i vektor varijance $\boldsymbol{\sigma}_B^2 \in \mathbb{R}^D$ (2.18) ulaznih vrijednosti određenog sloja, pri čemu je pojedini vektor ulaznih vrijednosti $\boldsymbol{h}_i \in \mathbb{R}^D$ dobiven na temelju i-tog primjera iz grupe primjera veličine B.

$$\boldsymbol{\mu}_B = \frac{1}{B}\sum_{i=1}^{B} \boldsymbol{h}_i \qquad \boldsymbol{\sigma}_B^2 = \frac{1}{B}\sum_{i=1}^{B} (\boldsymbol{h}_i - \boldsymbol{\mu}_B)^2 \tag{2.18}$$

Korištenjem rezultata iz (2.18) moguće je dobiti normalizirane vektore ulaznih vrijednosti $\widehat{\boldsymbol{h}}_i$ prema jednadžbi (2.19), gdje je ε konstanta koja osigurava da ne dođe do dijeljenja s nulom.

$$\widehat{\boldsymbol{h}}_i = \frac{\boldsymbol{h}_i - \boldsymbol{\mu}_B}{\sqrt{\boldsymbol{\sigma}_B^2 + \varepsilon}} \tag{2.19}$$

Izlazi iz sloja za normalizaciju grupe su vektori $\boldsymbol{z}_i$ iz jednadžbe (2.20), dobiveni primjenom naučenih vektora parametra $\boldsymbol{\beta} \in \mathbb{R}^D$, koji određuje optimalne aritmetičke sredine, i vektora parametara $\boldsymbol{\gamma} \in \mathbb{R}^D$, koji određuje optimalne standardne devijacije ulaza.

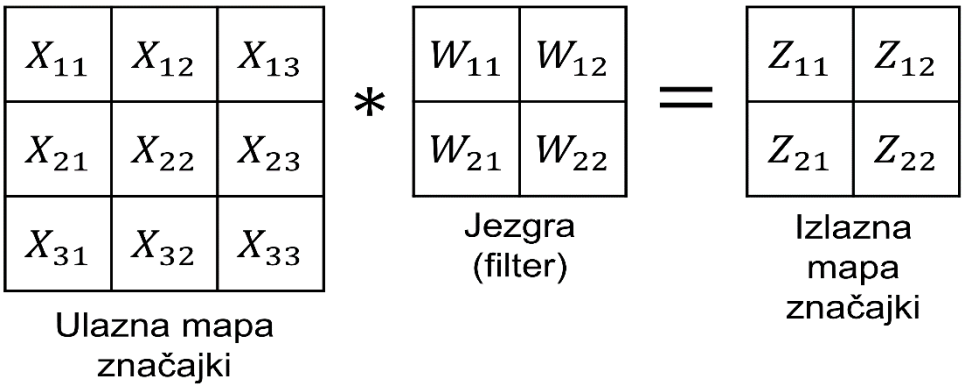
$$\boldsymbol{z}_i = \boldsymbol{\gamma} \odot \widehat{\boldsymbol{h}}_i + \boldsymbol{\beta} \tag{2.20}$$

Osim spomenutih tehnika, postoji čitav niz drugih tehnika za sprječavanje problema nestanka gradijenta, a jedna od njih je i primjena preskočnih veza (engl. *skip connections*) [102] u arhitekturi. Preskočne veze će biti detaljnije objašnjene kod modela za izvlačenje značajki korištenih u okviru disertacije. Primjena preskočnih veza nalazi čestu primjenu kod dubokih konvolucijskih modela.

Konvolucijska neuronska mreža

Konvolucijske neuronske mreže (engl. *convolutional neural network-CNN*) najbolji su primjer iskorištavanja kompozicijske strukture prisutne u podatcima, a njihov izum je inspiriran biološkim vidnim sustavom. U istraživanju [103] pokazano je da pojedini neuroni biološkog vidnog sustava reagiraju na podražaje locirane u malom dijelu vidnog polja, drugim riječima imaju malo lokalno receptivno polje. Nadalje, receptivna polja različitih neurona mogu se preklapati, a sva ona zajedno pokrivaju cijelo vidno polje. Drugi bitan zaključak spomenutog istraživanja je da pojedini neuroni reagiraju samo na određene jednostavne vizualne objekte, primjerice neki neuroni reagiraju na vertikalne bridove, drugi na horizontalne linije itd.. Osim toga, otkriveno je da neki neuroni imaju veća receptivna polja i da reagiraju na složenije vizualne objekte koji se mogu dobiti povezivanjem jednostavnijih vizualnih objekata, tj. neuroni viših razina produkt su izlaza neurona nižih razina. Pokazalo se da je ovu biološku inspiraciju u kontekstu dubokog strojnog učenja moguće formalizirati primjenom matematičke operacije konvolucije. Konvolucija je linearna operacija kojom je moguće kombinirati dvije funkcije kako bi se dobila nova funkcija, a konvolucijska neuronska mreža je ona koja koristi diskretnu konvoluciju nad značajkama i težinama, umjesto matričnog množenja. Terminologija koja se koristi kod dubokog strojnog učenja za elemente operacije konvolucije prikazana je slikom 2.21, pri čemu je bitno primijetiti da se parametri modela u kontekstu konvolucije nazivaju jezgrama (engl. *kernels*) ili filterima te da su njihove dimenzije manje od dimenzija

ulaznih značajki. Druga bitna napomena je da se u kontekstu DL-a zapravo ne koristi operacija konvolucije u pravom matematičkom smislu, već povezana operacija koja se zove unakrsna korelacija (engl. *cross-correlation*), pri čemu je razlika između njih da se kod konvolucije radi rotacija jezgre za 180°. Praktično, ova razlika nije važna jer će model naučiti iste vrijednosti parametara samo na obrnutim pozicijama u jezgri, stoga će se u nastavku rada i dalje koristiti naziv konvolucija za operaciju nad značajkama i jezgrama kao što je i uobičajeno u literaturi.

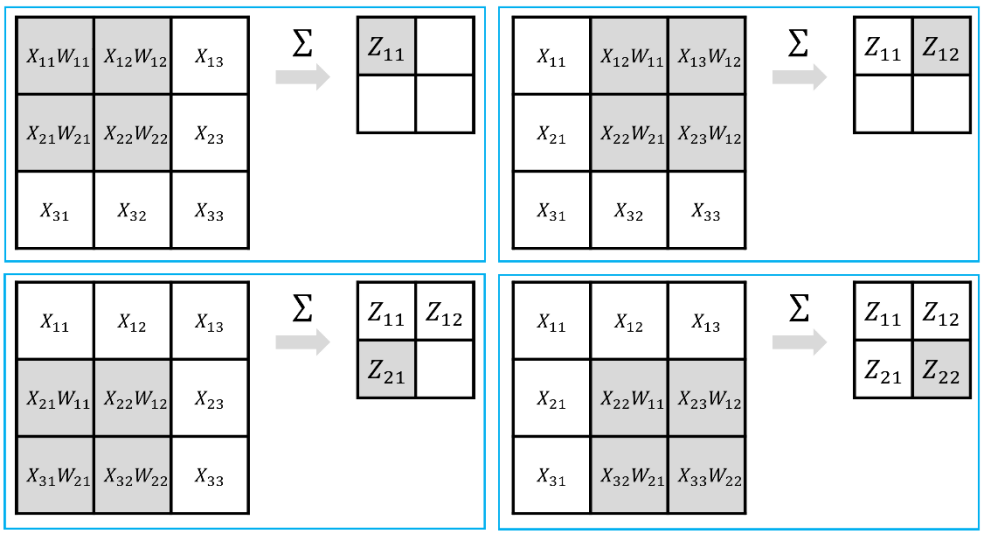


Slika 2.21 Elementi operacije konvolucije

Konvolucijske neuronske mreže spadaju u grupu unaprijednih mreža, jer i u njihovom slučaju podatci prolaze kroz model od ulaza do izlaza bez ikakvih petlji. Ovi modeli posjeduju određena svojstva koja ih čine pogodnima za obradu podataka koji imaju strukturu rešetke poput npr. vremenskih nizova (1D), slika (2D) ili video zapisa (3D). Prvo svojstvo su lokalne veze između neurona u susjednim slojevima, a ono proizlazi iz intuicije koja vrijedi za podatke poput npr. slika, da su lokacijski bliski pikseli često u korelaciji te zajedno formiraju određene vizualne objekte. Ovo svojstvo osigurano je korištenjem jezgre koja je puno manjih dimenzija u odnosu na ulaznu mapu značajki, što znači da će vrijednost pojedine značajke u izlaznoj mapi biti funkcija malog dijela susjednih neurona u ulaznoj mapi, što je ekvivalentno konceptu receptivnog polja biološkog vidnog sustava. Drugo svojstvo je korištenje dijeljenih težina (engl. *shared/tied weights*), koje potaknuto opažanjem da se jedan te isti vizualni objekt može pojaviti na različitim lokacijama u slici, što je formalizirano na način da su neuroni na različitim pozicijama u izlaznoj mapi funkcija jednih te istih jezgri. Upravo zato što su jezgre odgovorne za učenje „izgleda“ vizualnih objekata ovim pristupom se oponaša druga karakteristika bioloških vidnih sustava, a to je da određeni neuroni reagiraju samo na jednu vrstu vizualnih objekata. Treće svojstvo je učenje hijerarhije koncepata kroz duboki model, što zadovoljava pretpostavku da su kod biološkog sustava neuroni viših razina rezultat kombinacija neurona nižih razina, odnosno da se povezivanjem primitivnih vizualnih objekata poput linija i rubova

mogu oblikovati apstraktniji vizualni objekti. U matematičkom smislu ovo je postignuto na način da je svaka izlazna mapa funkcija svih ulaznih mapa, te je zbog toga receptivno polje „dubljih“ mapa veće od onih ranijih mapa. Radi navedenog, modeli s povećanjem dubine mogu raspoznavati složenije i veće vizualne objekte. Ova svojstva CNN-u daju prednost nad FNN modelima u domeni modeliranja podataka sa strukturom rešetke. Prva prednost je u vidu toga da za ulazne podatke istih dimenzija CNN modeli imaju obično puno manji broj parametara od FNN modela, zato što koriste dijeljene težine i jezgre malih dimenzija. Razlog zašto FNN modeli moraju imati puno veći broj parametara od CNN-a kod obrade podataka u obliku slika je taj da FNN modeli uče globalne karakteristike podataka dok CNN modeli uče lokalne. Konkretnije, FNN model mora naučiti posebne skupove parametara da bi raspoznao isti vizualni objekt na različitim lokacijama. Također, CNN modeli su zbog korištenja iste jezgre za izračun različitih izlaznih značajki fleksibilniji po pitanju dimenzija ulaznih podataka, tj. mogu prepoznati određene vizualne objekte neovisno od dimenzije ulazne slike.

Prolaz unaprijed kroz jedan konvolucijski sloj modela, za primjer u obliku slike (2D), ilustriran je na slici 2.22. Na slici je pokazano kako se jezgra **W**, dimenzija 2×2, pomiče po ulaznoj mapi **X**, dimenzija 3×3, što rezultira izlaznom mapom **Z** dimenzija 2×2. Na slici je moguće primijetiti koncept lokalnih receptivnih polja, npr. izlazna značajka Z_{11} ovisi o ulaznim značajkama $X_{11}, X_{12}, X_{21}, X_{22}$. Vidljiv je i koncept preklapanja receptivnih polja, npr. izlazne značajke Z_{11} i Z_{12} su funkcija istih ulaznih značajki X_{12} i X_{22}. Koncept dijeljenih težina jasan je iz toga da su sve izlazne značajke dobivene primjenom iste jezgre **W**.



Slika 2.22 Prolaz unaprijed kroz jedan konvolucijski sloj za dvije dimenzije

Ovo je pojednostavnjeni primjer jer su u općem slučaju ulazne i izlazne mape tenzori 3. reda, konkretno $\mathbf{X} \in \mathbb{R}^{H\times W\times D}$ i $\mathbf{Z} \in \mathbb{R}^{I\times J\times K}$, a jezgra je tenzor 4. reda $\mathbf{W} \in \mathbb{R}^{M\times N\times K\times D}$. U praksi za ulazne mape obično vrijedi $H = W$, za jezgre $M = N$ i posljedično za izlazne mape $I = J$. Primjerice, ako je ulazna mapa neka slika, onda dimenzije H i W predstavljaju visinu i širinu slike, a dimenzija D predstavlja kanale triju osnovnih boja. U općem slučaju izračun pojedinačnih elemenata izlazne mape značajki moguće je prikazati jednadžbom (2.21).

$$Z_{i,j}^{(k)} = \sum_{d,m,n} X_{i-1+m,j-1+n}^{(d)} \cdot W_{m,n}^{(k,d)} + b^{(k)} \tag{2.21}$$

U izrazu (2.21) $W_{m,n}^{(k,d)}$ je težina na poziciji (m,n) u jezgri koja povezuje d-tu ulaznu mapu s k-tom izlaznom mapom. $b^{(k)}$je element vektora pomaka $\boldsymbol{b} \in \mathbb{R}^K$ povezan s k-tom izlaznom mapom. $X_{i-1+m,j-1+n}^{(d)}$ je element d-te ulazne mape povezan s elementom na poziciji (i,j) svake izlazne mape $k = 1, 2 \ldots K$. $Z_{i,j}^{(k)}$ je element k-te izlazne mape na poziciji (i,j). Izračun svih elemenata u k-toj izlaznoj mapi moguće je prikazati u kompaktnom obliku primjenom operatora konvolucije $(*)$ prema (2.22).

$$\boldsymbol{Z}^{(k)} = \sum_{d} \boldsymbol{X}^{(d)} * \boldsymbol{W}^{(k,d)} + b^{(k)} \tag{2.22}$$

U jednadžbi (2.22) $\boldsymbol{Z}^{(k)} \in \mathbb{R}^{I\times J}$ je k-ta izlazna mapa, $\boldsymbol{X}^{(d)} \in \mathbb{R}^{H\times W}$ je d-ta ulazna mapa, $\boldsymbol{W}^{(k,d)} \in \mathbb{R}^{M\times N}$ je matrica težina jezgre $\mathbf{W}$ koja povezuje d-tu ulaznu mapu s k-tom izlaznom mapom. Rezultati primjene konvolucije prikazani u jednadžbama (2.21) i (2.22) mogu se praktično tumačiti na način da se kaže kako izlazna mapa pokazuje gdje se koja značajka nalazi u ulaznoj mapi, a svaka k-ta komponenta jezgre[4] $\mathbf{W}$ odgovorna je za prepoznavanje određenog tipa značajke. Drugi bitan zaključak je da su dimenzije I i J izlazne mape manje u odnosu na dimenzije H i W ulazne mape, a smanjenje slijedi formulu (2.23).

$$(I, J) = (H - M + 1), (W - N + 1) \tag{2.23}$$

Prolaz unaprijed nastavlja se prosljeđivanjem rezultata konvolucijskog sloja u aktivacijsku funkciju, pri čemu se kao i kod FNN modela danas najčešće koristi ReLU funkcija i njene generalizacije. Uz ova dva koraka, unaprijedni prolaz CNN modela može sadržavati sloj

[4] U literaturi se često za komponentu jezgre $\boldsymbol{W}^{(k)} \in \mathbb{R}^{M\times N\times D}$ isto koristi termin „jezgra“, u smislu da se uči k-različitih jezgri, a svaka uči određeni tip značajke. Iz ovog se jasno vidi da svaka $\boldsymbol{W}^{(k)}$ mora imati istu treću dimenziju (D) kao ulazna mapa.

sažimanja (engl. *pooling*). Sloj sažimanja ima dvostruku ulogu, povećati receptivno polje i reducirati broj parametara. Postupak sažimanja provodi se izračunom statističkog pokazatelja za svaku malu lokalnu regiju unutar mape značajki, pri čemu je uobičajeno da se te lokalne regije ne preklapaju. Statistički pokazatelji koji se obično koriste su aritmetička sredina ili maksimalna vrijednost. Izlaz iz sloja sažimanja je mapa značajki čije dimenzije odgovaraju broju definiranih regija, gdje će svaka regija biti reprezentirana izračunatim statističkim pokazateljem. Primjena sažimanja povećava invarijantnost izlazne mape značajki na male pomake u ulaznoj mapi. Ovo znači da će većina vrijednosti u mapi značajki dobivenoj sažimanjem biti nepromijenjena uslijed malih promjena pozicija vrijednosti u ulaznoj mapi značajki. Ovo svojstvo je korisno ako je bitno prepoznati da određena značajka postoji u ulaznim podatcima, ali nije toliko bitno na kojoj se točno prostornoj poziciji nalazi.

U izrazima (2.21) i (2.22) dane su formule za osnovni oblik operacije konvolucije koja se koristi u CNN modelima, međutim potrebno je navesti da postoje razna proširenja i specijalizirani oblici konvolucije. Jedan od osnovnih dodataka konvoluciji je dopunjavanje ulazne mape značajki nulama (engl. *zero padding*). Svrha ovog dodatka je spriječiti efekt smanjenja dimenzija izlazne mape značajki s dubinom modela. Drugi mogući dodatak konvoluciji je korak pomicanja jezgre (engl. *stride*). Uz korak veći od jedan, jezgra se više neće pomicati po svim mogućim pozicijama ulazne mape, što se matematički može prikazati modificiranjem jednadžbe (2.21) u jednadžbu (2.24), gdje s predstavlja veličinu koraka.

$$Z_{i,j}^{(k)} = \sum_{d,m,n} X_{(i-1)\cdot s+m,(j-1)\cdot s+n}^{(d)} \cdot W_{m,n}^{(k,d)} + b^{(k)} \tag{2.24}$$

Primjenom koraka s većeg od 1 može se postići povećanje receptivnog polja neurona izlazne mape, što može biti alternativa sloju sažimanja. Povećanje receptivnog polja moguće je napraviti i primjenom dilatirane konvolucije (engl. *dilated convolution*). Radi se o specijaliziranom obliku konvolucije koji čestu primjenu pronalazi u obradi vremenskih nizova. Dilatirana konvolucija radi na principu povećanja jezgre kroz dodavanje praznina između pojedinih elemenata. Ovaj oblik će biti detaljnije opisan kod finalnih modela izrađenih u okviru disertacije. U ovom dijelu prikazan je prolaz unaprijed kroz konvolucijski sloj za 2D podatke, na sličan način moguće je definirati prolaz unaprijed za 1D i 3D podatke. U nastavku će biti predstavljen prolaz unatrag kroz konvolucijski sloj za 2D podatke.

Prolazom unatrag kroz konvolucijski sloj, slično kao i kod FNN modela, žele se odrediti gradijenti funkcije gubitka po parametrima potrebni za učenje modela te gradijenti funkcije gubitka po ulazu u sloj. Kako bi ovi gradijenti mogli biti izračunati, trenutni konvolucijski sloj

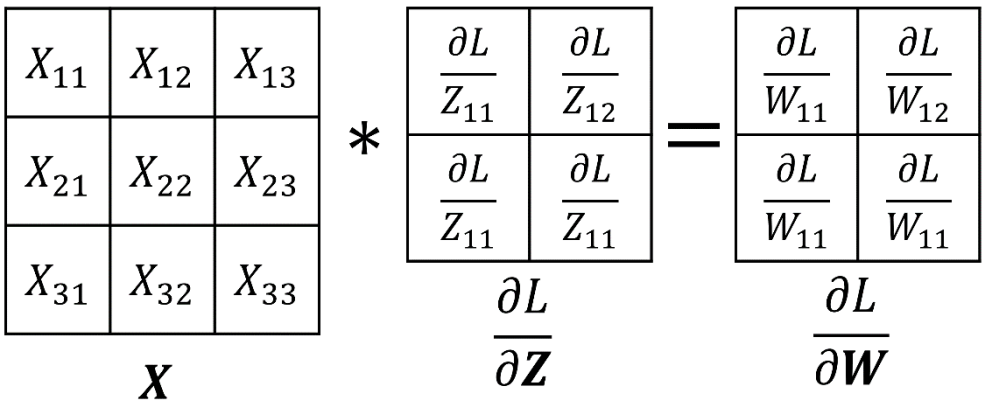
l prima gradijent funkcije gubitka po izlaznoj mapi značajki $\partial L/\partial \mathbf{Z}$ izračunat od strane konvolucijskog sloja $l+1$. Derivacija funkcije gubitka L po elementu $W_{m,n}^{(k,d)}$ jezgre $\mathbf{W}$ prikazana je u jednadžbi (2.25).

$$\frac{\partial L}{\partial W_{m,n}^{(k,d)}} = \sum_{i,j} \frac{\partial L}{Z_{i,j}^{(k)}} \cdot X_{i-1+m,j-1+n}^{(d)} \tag{2.25}$$

Gradijent funkcije gubitka L po svim elementima matrice $\boldsymbol{W}^{(k,d)}$ jezgre $\mathbf{W}$ moguće je izračunati korištenjem operatora konvolucije na temelju izraza (2.26).

$$\frac{\partial L}{\partial \boldsymbol{W}^{(k,d)}} = \frac{\partial L}{\partial \boldsymbol{Z}^{(k)}} * \boldsymbol{X}^{(d)} \tag{2.26}$$

Interpretacija koja slijedi iz jednadžbe (2.26) je da jezgra $\mathbf{W}$ s težinama u matrici $\boldsymbol{W}^{(k,d)}$ utječe na k-tu izlaznu mapu $\boldsymbol{Z}^{(k)}$ preko d-te ulazne mape $\boldsymbol{X}^{(d)}$ te se zato gradijent računa zasebno za svaku matricu $\boldsymbol{W}^{(k,d)}$. Ova veza opisana je operacijom konvolucije gdje se $\partial L/\partial \boldsymbol{Z}^{(k)}$pomiče po $\boldsymbol{X}^{(d)}$, što je ilustrirano slikom 2.23.



Slika 2.23 Računanje gradijenta funkcije gubitka s obzirom na jezgru za dvije dimenzije

Izračun gradijenta funkcije gubitka po pomaku moguće je napraviti na sličan način. Finalno, derivacija funkcije gubitka po elementu d-te ulazne mape na poziciji (p,r) $X_{p,r}^{(d)}$ prikazana je jednadžbom (2.27).

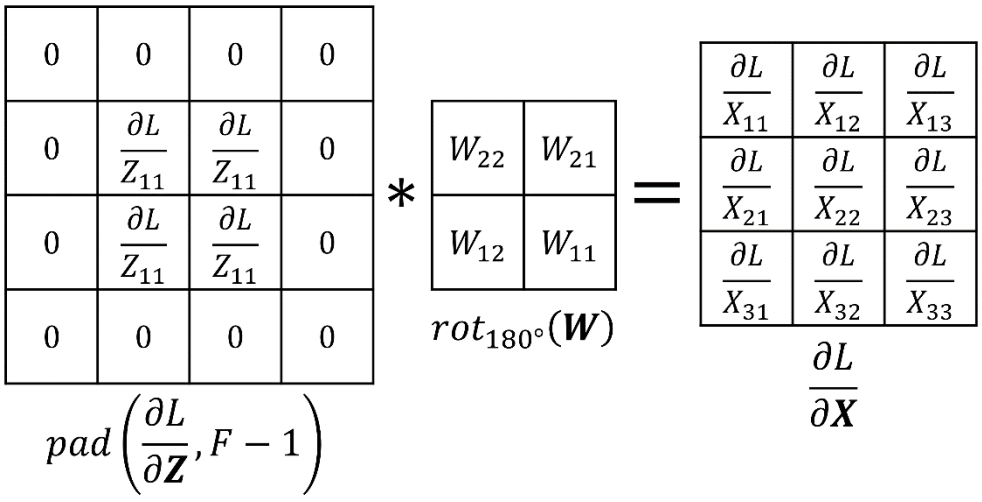
$$\frac{\partial L}{\partial X_{p,r}^{(d)}} = \sum_{\substack{i,m: \\ i-1+m=p}} \sum_{\substack{j,n: \\ j-1+n=r}} \sum_{k} \frac{\partial L}{Z_{i,j}^{(k)}} \cdot W_{m,n}^{(k,d)} \tag{2.27}$$

U izrazu (2.27), notacija " : " kod operatora sume znači da vrijednosti koje poprimaju indeksi sume moraju zadovoljavati zadani uvjet. Konkretnije, sumiraju se elementi s indeksima (i,m) za koje je ispunjen uvjet $i-1+m=p$. Elementi unutar operatora sume (npr. $W_{m,n}^{(k,d)}$) koji

poprime vrijednost indeksa koji ne zadovoljava uvjet bivaju zanemareni u izračunu. Gradijent funkcije gubitka po svim elementima matrice d-te ulazne mape $\boldsymbol{X}^{(d)}$ slijedi na temelju izraza (2.28).

$$\frac{\partial L}{\partial \boldsymbol{X}^{(d)}} = \sum_{k} pad\left(\frac{\partial L}{\partial \boldsymbol{Z}^{(k)}}, \mathrm{F}-1\right) * rot_{180°}\left(\boldsymbol{W}^{(k,d)}\right) \tag{2.28}$$

U izrazu (2.28) $pad(\cdot, F-1)$ je funkcija nadopunjavanja nulom sa svake strane prostorne dimenzije gradijenta $\partial L/\partial \boldsymbol{Z}^{(k)}$, gdje je količina nadopunjavanja određena na temelju prostorne dimenzije jezgre **W**, tj. $F = M = N$. Funkcija $rot_{180°}(\cdot)$, radi rotaciju („zrcaljenje“) matrice $\boldsymbol{W}^{(k,d)}$ za 180°. Ilustracija jednadžbe (2.28), za slučaj u dvije dimenzije, je na slici 2.24.



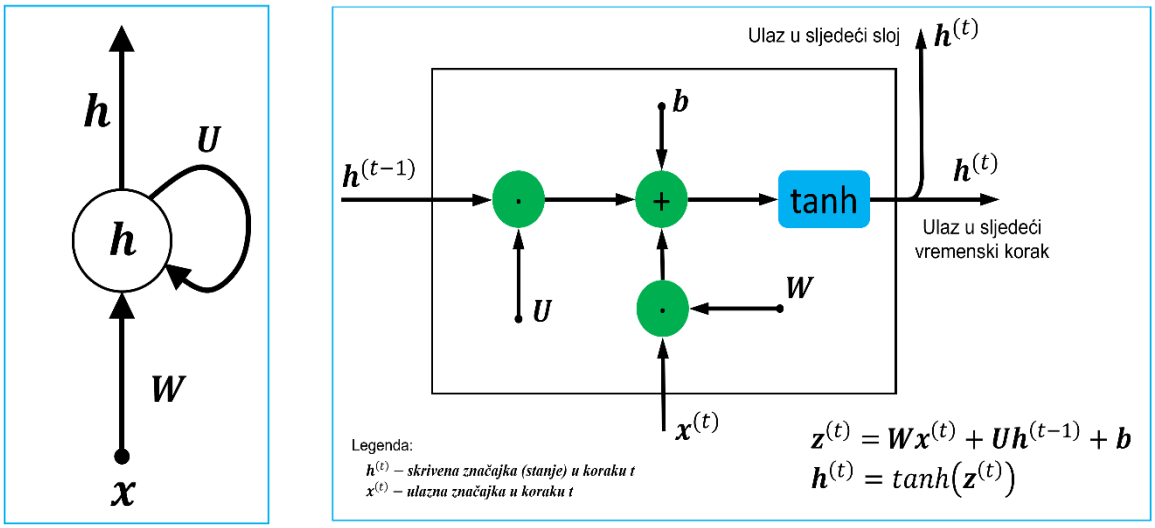
Slika 2.24 Računanje gradijenta funkcije gubitka po ulaznoj mapi značajki za dvije dimenzije

Tumačenje jednadžbe (2.28) je da d-ta ulazna mapa $\boldsymbol{X}^{(d)}$ utječe na svaku k-tu izlaznu mapu **Z**, preko konvolucije s k-tim komponentama jezgre **W**, i zato je potrebno sumirati doprinose po k-tim komponentama kako bi se dobio gradijent $\partial L/\partial \boldsymbol{X}^{(d)}$. Ovaj gradijent se prosljeđuje sloju $l-1$ te se ovaj postupak ponavlja sve dok algoritam BP-a ne dođe do ulaza modela.

Ranije je spomenuto da se za obradu vremenskih nizova mogu koristiti 1D konvolucijski slojevi. Kod modeliranja vremenskih nizova često je potrebno uhvatiti globalni kontekst u podatcima, a u tom smislu svojstvo lokalnih veza koje odlikuje konvolucijske neuronske mreže je ograničenje. Ovo ograničenje se može zaobići povećanjem receptivnog polja neurona. Povećanje receptivnog polja može biti postignuto ili definiranjem jako dubokog 1D CNN modela ili korištenjem dilatirane konvolucije. Alternativa ovakvim pristupima su povratne neuronske mreže.

Povratna neuronska mreža

Povratne neuronske mreže (engl. *recurrent neural network-RNN*) su razvijene s ciljem obrade nizova proizvoljne duljine. Riječ „povratna“ u njihovom nazivu posljedica je činjenice što ovakve mreže koriste cikluse, odnosno povratne petlje, u svojoj arhitekturi. Ovo svojstvo ih razlikuje od CNN i FNN modela. Razlog zašto se RNN modeli mogu nositi s nizovima proizvoljnih duljina je primjena istih parametara u svakom vremenskom koraku[5] niza. Ovaj koncept se kod modela strojnog učenja zove dijeljenje težina te je već spomenut kod konvolucijskih modela. Kod CNN modela, pojedina skrivena značajka je funkcija male lokalne regije neurona u prethodnom sloju. U slučaju RNN modela, skrivene značajke[6] su funkcija svih skrivenih značajki prethodnih vremenskih koraka te trenutnog ulaza modela (vidi sliku 2.25, desno). Dijeljenje parametara na ovaj način omogućava modeliranje puno dužih nizova u odnosu na ono što može jedan sloj konvolucijske mreže.



Slika 2.25 Jednostavni RNN sloj: kompaktni prikaz (lijevo), detaljni prikaz za 1 korak (desno)

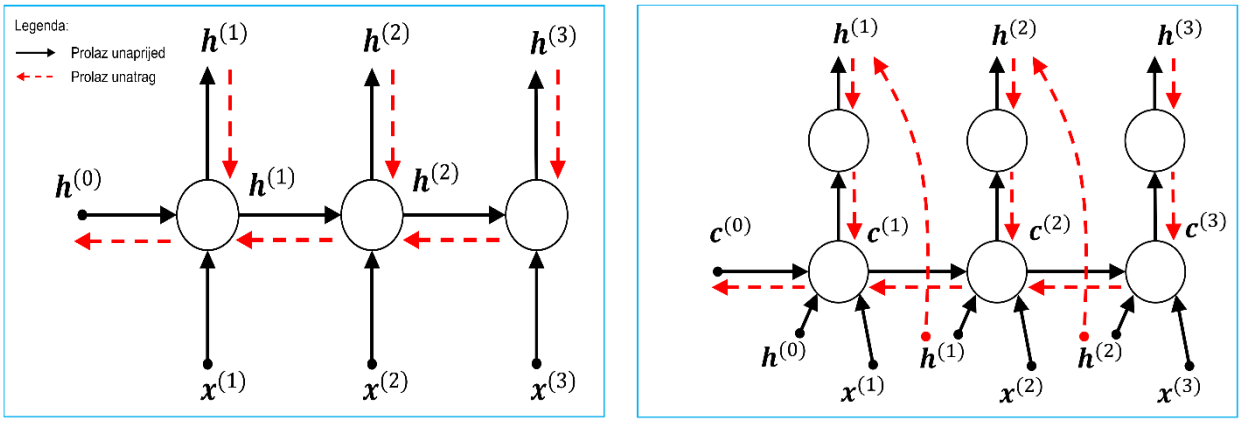
Intuicija koja je vodila istraživače u razvoju ovakve arhitekture je da su susjedni elementi niza obično povezani na neki način, a kada bi ih se tretiralo kao nezavisne izgubile bi se bitne informacije. Drugi motiv je da se određena informacija može naći na više različitih pozicija unutar niza, stoga nije potrebno učiti različite parametre ovisno od pozicije. Treći razlog je da

[5] Nizovi ne moraju nužno biti vremenski, pa tako i pojedini korak ne mora biti vezan uz vrijeme. Primjeri takvih nizova su riječi u rečenici na koje se isto primjenjuju RNN modeli. S obzirom na temu disertacije, element niza će se uglavnom uvijek povezivati s tijekom vremena.

[6] Skrivene značajke se u kontekstu RNN modela nazivaju i „stanjem“, a razlog za ovaj naziv je inspiracija dinamičkim sustavima, gdje trenutno stanje ovisi o prethodnom stanju i trenutnom ulazu sustava.

model mora uzimati u obzir poredak ulaznih podataka, tj. mora modelirati relativnu poziciju pojedine značajke u nizu. Završno, korištenje dijeljenih parametara omogućava obradu niza proizvoljne duljine i u krajnjoj liniji generalizaciju modela na podatke različitih duljina. Iz ovoga se može zaključiti zašto unaprijedne mreže nisu adekvatne za zadatke obrade nizova. FNN modeli su bazirani na pretpostavci nezavisnosti podataka, tako da ne koriste informacije o prethodnim stanjima. Također, zato što koriste potpunu povezanost između neurona, moraju imate posebne parametre za modeliranje iste značajke na različitim pozicijama u nizu te ne mogu raditi s ulaznim primjerima različitih duljina.

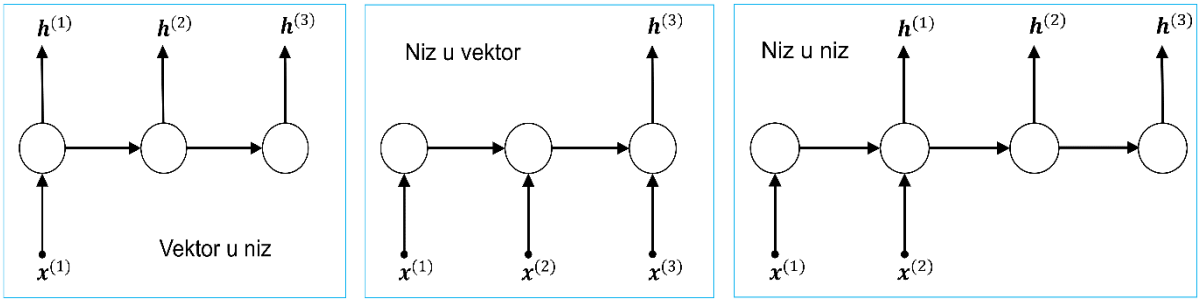
RNN modeli svoju primjenu nalaze kod problema poput predikcije vremenskih serija, obrade prirodnog jezika ili analize video zapisa. Oblik ulaznih podataka i oznaka ovisi o konkretnom zadatku, a u okviru disertacije zanimljiva je situacija gdje ulazni i izlazni niz imaju jednak broj koraka t te su koraci sinkronizirani, odnosno potrebno je napraviti predikciju u svakom vremenskom koraku. Konkretno, ulazni niz je $\mathbf{X}_1^{\mathrm{T}} = \left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(\mathrm{T})}\right)$, izlazni niz je $\mathbf{Y}_1^{\mathrm{T}} = \left(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(\mathrm{T})}\right)$, pri čemu je član ulaznog niza u koraku t vektor $\boldsymbol{x}^{(t)} \in \mathbb{R}^D$, a član izlaznog niza u koraku t je binarni vektor $\boldsymbol{y}^{(t)} \in \{0,1\}_{i=1}^{C}$ za kojeg vrijedi $\sum_{i=1}^{C} y_i^{(t)} = 1$. Ova situacija može se vizualizirati ako se graf sa slike 2.25 (lijevo) „odmota“ kroz vremenske korake. Novi graf se onda može interpretirati kao unaprijedni model koji ima onoliko slojeva[7] koliko ima vremenskih koraka, pri čemu svi slojevi dijele iste parametre. Primjer ove interpretacije za problem sinkroniziranih ulaznih i izlaznih nizova je na slici 2.26. Na razmotanom grafu prikazano je kako RNN sekvencijalno obrađuje ulazne podatke $\boldsymbol{x}^{(t)}$, pri čemu čuva stanje u vektoru $\boldsymbol{h}^{(t)}$ koji sadrži informacije o svim prošlim članovima niza.



Slika 2.26 Računski graf za tri vremenska koraka RNN sloja (lijevo) i LSTM sloja (desno)

[7] Tehnički radi se zapravo o RNN modelu koji ima samo **jedan** sloj.

Osim zadataka obrade vremenskih nizova sa sinkroniziranim ulaznim i izlaznim nizovima jednake duljine, postoje i drugi zadatci (slika 2.27).



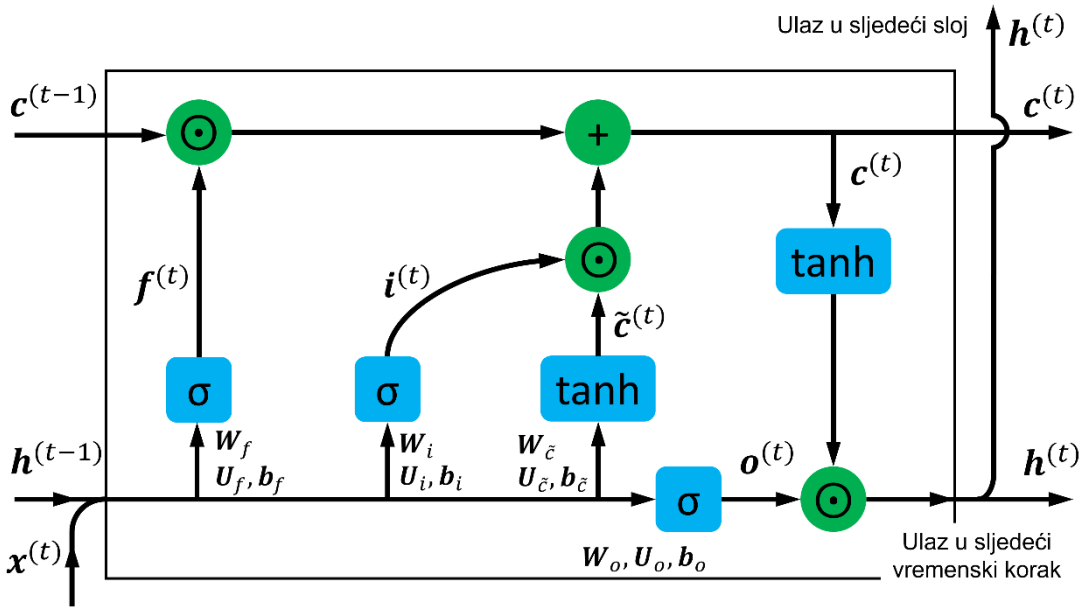
Slika 2.27 Zadatci obrade nizova s obzirom na oblik ulaza i izlaza modela

Kada su primjeri u obliku vektora[8] značajki, a izlazi u obliku nizova vektora značajki, radi se o problemu „vektora u niz“, a primjer zadatka koji spada u ovu grupu je generiranje opisa slike, gdje je ulaz slika, a izlaz su riječi koje opisuju sliku. Ako je ulaz niz, a izlaz vektor oznaka generiran u zadnjem vremenskom koraku, to je problem „niza u vektor“, primjer ove grupe problema je prepoznavanje aktivnosti u video zapisu. Problemi „niza u niz“ se razlikuju po tome da li su ulazni i izlazni niz sinkronizirani ili ne te da li su jednake duljine. Primjer zadatka gdje su nizovi sinkronizirani i jednake duljine je vremenska segmentacija aktivnosti u video zapisu, dok je zadatak vremenske detekcije aktivnosti primjer problema s nizovima različite duljine.

Iako RNN modeli teorijski mogu modelirati zavisnosti i relacije prisutne u veoma dugim nizovima, njihovo učenje se pokazalo veoma teškim zbog problema nestajanja ili eksplozije gradijenta [104]. U istraživanju [104] pokazano je da ovaj problem kod RNN modela nije moguće riješiti samo kroz promjenu aktivacijske funkcije (npr. iz sigmoidne u ReLU), jer je korijen problema u dijeljenim parametrima. Preciznije, problem je u matrici parametara koja se u svakom vremenskom koraku množi sa skrivenim značajkama (na slici 2.25 to je matrica $\boldsymbol{U}$). Analiza svojstvenih vrijednosti spomenute matrice ukazuje da će cijeli umnožak ovisiti o najvećoj svojstvenoj vrijednosti. Ako je ta vrijednost manja od jedan, s porastom broja koraka gradijent će težiti u nulu, a ako je najveća svojstvena vrijednost veća od jedan, gradijent će u slučaju dugih vremenskih nizova eksplodirati. Istraživanje [105] kao rješenje predlaže drugačiju arhitekturu povratne neuronske mreže (slika 2.28), gdje mreža koristi koncept duge kratkoročne memorije (engl. *long short-term memory-LSTM*). Glavna ideja istraživanja je bila rasteretiti skriveno stanje $\boldsymbol{h}^{(t)}$ kroz uvođenje memorijske ćelije $\boldsymbol{c}^{(t)}$ odgovorne za „pamćenje“

[8] Može biti i matrica, ili tenzor višeg reda, bitno je da nije u obliku niza.

prethodnih stanja. Iako LSTM ublažava problem eksplozije gradijenta, primarno je osmišljen da riješi problem nestajućeg gradijenta jer uvođenjem memorijske ćelije postoji još jedan dodatan put kojim gradijent može biti prosljeđen. Na slici 2.26 je upravo pokazana ta razlika u toku gradijenta između jednostavnog RNN sloja i LSTM sloja. Naziv „duga kratkoročna memorija“ dolazi iz sljedeće intuicije. Jednostavni RNN ima dugoročnu memoriju u obliku težina koje se mijenjaju sporo tijekom učenja te na taj način opisuju opće znanje u podatcima, a kratkoročnu memoriju u obliku vrijednosti skrivenog stanja koje se mijenja u svakom vremenskom koraku. LSTM produžuje upravo tu kratkoročnu memoriju kroz međuspremnik u vidu memorijske ćelije [106]. Osim LSTM-a postoje i drugi tipovi arhitektura koji su razvijeni sa sličnom motivacijom poput npr. GRU (engl. *Gated recurrent unit*) sloja [107].



Slika 2.28 Detaljan prikaz LSTM sloja

Izračun prolaza unaprijed kroz LSTM sloj u vremenskom koraku t zahtjeva ulaz $\boldsymbol{x}^{(t)} \in \mathbb{R}^D$ te vrijednost skrivenog stanja i memorijske ćelije iz prethodnog vremenskog koraka $\boldsymbol{h}^{(t-1)} \in \mathbb{R}^H$ i $\boldsymbol{c}^{(t-1)} \in \mathbb{R}^H$. Jednadžbe prolaza unaprijed od (2.29) do (2.32) moguće je tumačiti kao mehanizme sloja koji pokušavaju naučiti kako ulazni podatci $\boldsymbol{x}^{(t)}$ i prošla skrivena stanja $\boldsymbol{h}^{(t-1)}$ trebaju mijenjati informacije u memorijskoj ćeliji $\boldsymbol{c}^{(t)}$ i skrivenom stanju $\boldsymbol{h}^{(t)}$ na temelju jednadžbi (2.33) i (2.34). Ulazni podatci djeluju na memorijsku ćeliju preko matrica parametara $\boldsymbol{W}_{(\cdot)} \in \mathbb{R}^{H \times D}$, a prošla skrivena stanja preko matrica $\boldsymbol{U}_{(\cdot)} \in \mathbb{R}^{H \times H}$, pri čemu vektori pomaka $\boldsymbol{b}_{(\cdot)} \in \mathbb{R}^H$ omogućuju dodatnu fleksibilnost transformacije. Usporedbom jednadžbi za prolaz unaprijed LSTM sloja i jednostavnog RNN sloja (slika 2.25, desno) lako je zaključiti da LSTM ima četiri puta više parametara što mu daje prednost u modeliranju duljih nizova.

$$\boldsymbol{f}^{(t)} = \sigma\left(\boldsymbol{W}_f \boldsymbol{x}^{(t)} + \boldsymbol{U}_f \boldsymbol{h}^{(t-1)} + \boldsymbol{b}_f\right) = \sigma\left(\boldsymbol{z}_f^{(t)}\right) \tag{2.29}$$

$$\boldsymbol{i}^{(t)} = \sigma\left(\boldsymbol{W}_i \boldsymbol{x}^{(t)} + \boldsymbol{U}_i \boldsymbol{h}^{(t-1)} + \boldsymbol{b}_i\right) = \sigma\left(\boldsymbol{z}_i^{(t)}\right) \tag{2.30}$$

$$\tilde{\boldsymbol{c}}^{(t)} = tanh\left(\boldsymbol{W}_{\tilde{c}} \boldsymbol{x}^{(t)} + \boldsymbol{U}_{\tilde{c}} \boldsymbol{h}^{(t-1)} + \boldsymbol{b}_{\tilde{c}}\right) = tanh\left(\boldsymbol{z}_{\tilde{c}}^{(t)}\right) \tag{2.31}$$

$$\boldsymbol{o}^{(t)} = \sigma\left(\boldsymbol{W}_o \boldsymbol{x}^{(t)} + \boldsymbol{U}_o \boldsymbol{h}^{(t-1)} + \boldsymbol{b}_o\right) = \sigma\left(\boldsymbol{z}_o^{(t)}\right) \tag{2.32}$$

$$\boldsymbol{c}^{(t)} = \boldsymbol{f}^{(t)} \odot \boldsymbol{c}^{(t-1)} + \boldsymbol{i}^{(t)} \odot \tilde{\boldsymbol{c}}^{(t)} \tag{2.33}$$

$$\boldsymbol{h}^{(t)} = \boldsymbol{o}^{(t)} \odot tanh\left(\boldsymbol{c}^{(t)}\right) \tag{2.34}$$

U gornjim izrazima σ je sigmoidna funkcija, a tanh je funkcija tangens hiperbolni koja je definirana prema jednadžbi (2.35), a radi preslikavanje ulaza u raspon (-1,1).

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.35}$$

Intuitivno tumačenje rezultata jednadžbe (2.29) je da $\boldsymbol{f}^{(t)} \in \mathbb{R}^H$ predstavlja mehanizam zaboravljanja (engl. *forget gate*). Spomenuta interpretacija se oslanja na činjenicu da će vrijednost svakog elementa vektora $\boldsymbol{f}^{(t)}$ biti u rasponu (0, 1) zbog korištenja sigmoidne funkcije. Zbog toga će prema izrazu (2.33) množenje $\boldsymbol{f}^{(t)}$ sa starim stanjem memorijske ćelije $\boldsymbol{c}^{(t-1)}$ rezultirati odlukom o udjelu stare informacije, prisutne u $\boldsymbol{c}^{(t-1)}$, koja se želi zadržati u novom stanju memorije $\boldsymbol{c}^{(t)}$. To znači da se uz $\boldsymbol{f}^{(t)} = \boldsymbol{0}$ stara informacija u potpunosti odbacuje. Rezultat izraza (2.30) je $\boldsymbol{i}^{(t)} \in \mathbb{R}^H$ koji se može shvatiti kao mehanizam ulaza (engl. *input gate*). Ovaj mehanizam također koristi sigmoidnu funkciju putem koje odlučuje koliki udio nove informacije iz $\tilde{\boldsymbol{c}}^{(t)}$ treba ući u memoriju $\boldsymbol{c}^{(t)}$, što je drugi član izraza (2.33). Drugim riječima, ako je $\boldsymbol{i}^{(t)} = \boldsymbol{0}$, onda nova informacija nije relevantna za čuvanje u memoriji. Analogno tome $\tilde{\boldsymbol{c}}^{(t)} \in \mathbb{R}^H$ iz jednadžbe (2.31) interpretira se kao mehanizam modifikacije (engl. *modulation*) koji definira kako najbolje kombinirati $\boldsymbol{x}^{(t)}$ i $\boldsymbol{h}^{(t-1)}$ iz mehanizma ulaza $\boldsymbol{i}^{(t)}$. Zadatak mehanizma izlaza (engl. *output gate*) $\boldsymbol{o}^{(t)} \in \mathbb{R}^H$ iz izraza (2.32) je da uči koji dio memorije $\boldsymbol{c}^{(t)}$ je bitan za izračun izlaza $\boldsymbol{h}^{(t)}$. Ako je $\boldsymbol{o}^{(t)} = \boldsymbol{0}$, ništa iz memorije neće biti priključeno izlazu u trenutnom vremenskom koraku što se vidi iz izraza (2.34). U izrazu (2.33) za memorijsku ćeliju $\boldsymbol{c}^{(t)}$, vidljivo je da se u izračunu koriste samo linearne operacije, pa je iz toga jasno zašto gradijent može lako prolaziti kroz memorijsku ćeliju. Na temelju istog izraza moguće je zaključiti da gradijent neće eksplodirati kod memorijske ćelije, jer su mehanizmi zaboravljanja i ulaza ograničeni na raspon (0, 1) zbog sigmoidne funkcije. Međutim, problem

eksplozije gradijenta se i dalje može dogoditi kod skrivenog stanja $\boldsymbol{h}^{(t)}$, ali empirijski rezultati su pokazali da je to puno rjeđe nego kod običnog RNN sloja. Opisani koraci prolaza unaprijed se ponavljaju sekvencijalno do zadnjeg vremenskog koraka, pri čemu se u slučaju sinkroniziranih ulaznih i izlaznih nizova iste duljine računa vrijednost gubitka u svakom vremenskom koraku $L^{(t)}$, a ukupni gubitak je jednak sumi gubitaka po koracima. Moguće je kreirati i model s većim brojem LSTM slojeva, a u njemu izlazi iz skrivenih stanja $\boldsymbol{h}^{(t)}$ u sloju l predstavljaju ulazne značajke u sloju $l+1$, tj. to se može zamisliti kao $\boldsymbol{h}_l^{(t)} = \boldsymbol{x}_{l+1}^{(t)}$.

Prolaz unatrag kod povratnih neuronskih mreža provodi se modificiranim algoritmom unatražnog prosljeđivanja pogreške koji se zove algoritam unatražnog prostiranja pogreške kroz vrijeme (engl. *backpropagation through time–BPTT*)[108]. Naziv algoritma je posljedica dvije činjenice, prva je da se kod RNN modela gradijent ne računa samo kroz slojeve već i kroz vremenske korake niza, a druga je da se izračun gradijenata radi u suprotnom smjeru niza, tj. suprotno od tijeka vremenskih koraka. U suštini ovaj algoritam je u potpunosti isti kao i standardni BP, ako se vremenski koraci u sloju zamisle kao zasebni slojevi unaprijedne mreže uz dijeljene težine u vremenskim koracima, kako je prethodno pokazano na slici 2.26. U nastavku će biti prikazan prolaz unatrag kroz jedan vremenski korak LSTM sloja te izračun gradijenata za cijeli sloj kod problema sinkroniziranih ulaznih i izlaznih nizova jednake duljine. Prvi bitan rezultat za izvod prolaza unatrag vezan je uz derivaciju gubitka po gubitcima vremenskih koraka (2.36).

$$\frac{\partial L}{\partial L^{(t)}} = 1 \tag{2.36}$$

U prolazu unatrag potrebno je izračunati gradijent gubitka po parametrima trenutnog vremenskog koraka, ali i po ulaznim vrijednostima memorijske ćelije $\boldsymbol{c}^{(t-1)}$ i skrivenog stanja $\boldsymbol{h}^{(t-1)}$ prethodnog vremenskog koraka. Iz vremenskog koraka $t+1$ dolaze informacije o gradijentima $\partial L/\partial \boldsymbol{h}^{(t)}$ i $\partial L/\partial \boldsymbol{c}^{(t)}$. Gradijent $\partial L/\partial \boldsymbol{h}^{(t)}$ moguće je rastaviti na dvije komponente iz razloga što skriveno stanje $\boldsymbol{h}^{(t)}$ utječe na izlaz u trenutnom koraku t, ali i na buduće skriveno stanje $\boldsymbol{h}^{(t+1)}$, prema izrazu (2.37) gdje je druga jednakost dobivena na temelju (2.36).

$$\frac{\partial L}{\partial \boldsymbol{h}^{(t)}} = \underbrace{\frac{\partial L^{(t)}}{\partial \boldsymbol{h}^{(t)}}}_{\substack{Gubitak\ u\ trenutnom \\ vremenskom\ koraku}} + \underbrace{\frac{\partial L^{(t+1)}}{\partial \boldsymbol{h}^{(t)}}}_{\substack{Gubitak\ \boldsymbol{svih\ budućih} \\ vremenskih\ koraka}} = \frac{\partial L}{\partial \boldsymbol{h}^{(t)}} + \left(\frac{\partial \boldsymbol{h}^{(t+1)}}{\partial \boldsymbol{h}^{(t)}}\right)^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{h}^{(t+1)}} \tag{2.37}$$

U slučaju zadnjeg vremenskog koraka niza, gradijenta gubitka po skrivenom stanju $\boldsymbol{h}^{(\mathrm{T})}$ nema drugi član u jednadžbi (2.37), što je pokazano u (2.38).

$$\frac{\partial L}{\partial \boldsymbol{h}^{(t)}} = \begin{cases} \frac{\partial L^{(t)}}{\partial \boldsymbol{h}^{(t)}} + \frac{\partial L^{(t+1)}}{\partial \boldsymbol{h}^{(t)}} & za\ t\ < \mathrm{T} \\ \frac{\partial L^{(t)}}{\partial \boldsymbol{h}^{(t)}} & za\ t\ = \mathrm{T} \end{cases} \tag{2.38}$$

Analogno je moguće analizirati gradijent gubitka po memorijskoj ćeliji $\boldsymbol{c}^{(t)}$, koja istodobno utječe na skriveno stanje $\boldsymbol{h}^{(t)}$ i na memorijsku ćeliju sljedećeg koraka $\boldsymbol{c}^{(t+1)}$ što se vidi u (2.39).

$$\frac{\partial L}{\partial \boldsymbol{c}^{(t)}} = \frac{\partial L^{(t)}}{\partial \boldsymbol{c}^{(t)}} + \frac{\partial L^{(t+1)}}{\partial \boldsymbol{c}^{(t)}} = \left(\frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{c}^{(t)}}\right)^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{h}^{(t)}} + \left(\frac{\partial \boldsymbol{c}^{(t+1)}}{\partial \boldsymbol{c}^{(t)}}\right)^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{c}^{(t+1)}} \tag{2.39}$$

Slično kao i za slučaj skrivenog stanja zadnjeg vremenskog koraka, i za memorijsku ćeliju u zadnjem vremenskom koraku vrijedi da je drugi član u izrazu (2.39) jednak nuli.

$$\frac{\partial L}{\partial \boldsymbol{c}^{(t)}} = \begin{cases} \frac{\partial L^{(t)}}{\partial \boldsymbol{c}^{(t)}} + \frac{\partial L^{(t+1)}}{\partial \boldsymbol{c}^{(t)}} & za\ t\ < \mathrm{T} \\ \frac{\partial L^{(t)}}{\partial \boldsymbol{c}^{(t)}} & za\ t\ = \mathrm{T} \end{cases} \tag{2.40}$$

Koristeći rezultate iz (2.37) i (2.39) te lančano pravilo deriviranja, moguće je izračunati gradijent po ranije opisanim mehanizmima LSTM sloja, kako je pokazano u jednadžbama (2.41) do (2.44).

$$\frac{\partial L}{\partial \boldsymbol{o}^{(t)}} = \left(\frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{o}^{(t)}}\right)^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{h}^{(t)}} = diag(tanh(\boldsymbol{c}^{(t)})) \frac{\partial L}{\partial \boldsymbol{h}^{(t)}} \tag{2.41}$$

$$\frac{\partial L}{\partial \boldsymbol{i}^{(t)}} = \left(\frac{\partial \boldsymbol{c}^{(t)}}{\partial \boldsymbol{i}^{(t)}}\right)^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{c}^{(t)}} = diag\left(\tilde{\boldsymbol{c}}^{(t)}\right) \frac{\partial L}{\partial \boldsymbol{c}^{(t)}} \tag{2.42}$$

$$\frac{\partial L}{\partial \boldsymbol{f}^{(t)}} = \left(\frac{\partial \boldsymbol{c}^{(t)}}{\partial \boldsymbol{f}^{(t)}}\right)^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{c}^{(t)}} = diag\left(\boldsymbol{c}^{(t-1)}\right) \frac{\partial L}{\partial \boldsymbol{c}^{(t)}} \tag{2.43}$$

$$\frac{\partial L}{\partial \tilde{\boldsymbol{c}}^{(t)}} = \left(\frac{\partial \boldsymbol{c}^{(t)}}{\partial \tilde{\boldsymbol{c}}^{(t)}}\right)^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{c}^{(t)}} = diag\left(\boldsymbol{i}^{(t)}\right) \frac{\partial L}{\partial \boldsymbol{c}^{(t)}} \tag{2.44}$$

U gornjim izrazima $diag(\cdot)$ je dijagonalna matrica u kojoj su elementi dijagonale određeni vektorom. Uz poznate gradijente po mehanizmima sloja, slijedi izračun gradijenata po afinim transformacijama $\boldsymbol{z}_{(\cdot)}^{(t)}$ pojedinog mehanizma. S obzirom da su na afine transformacije primijenjene sigmoidna ili tanh aktivacijska funkcija, za daljnji izvod su potrebni rezultati derivacija aktivacijskih funkcija iz jednadžbi u (2.45)

$$\frac{d\,\sigma(x)}{dx} = \sigma(x)\big(1-\sigma(x)\big) \qquad \frac{d\,tanh(x)}{dx} = 1 - tanh^2(x) \tag{2.45}$$

Gradijent gubitka po afinoj transformaciji mehanizma zaboravljanja prikazan je u izrazu (2.46)

$$\frac{\partial L}{\partial \boldsymbol{z}_f^{(t)}} = \left(\frac{\partial \boldsymbol{f}^{(t)}}{\partial \boldsymbol{z}_f^{(t)}}\right)^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{f}^{(t)}} = \left(\frac{\partial \sigma\left(\boldsymbol{z}_f^{(t)}\right)}{\partial \boldsymbol{z}_f^{(t)}}\right)^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{f}^{(t)}} = diag\big(\boldsymbol{f}^{(t)}\big) diag\big(1-\boldsymbol{f}^{(t)}\big) \frac{\partial L}{\partial \boldsymbol{f}^{(t)}} \tag{2.46}$$

Na sličan način moguće je napraviti izračun gradijenta po $\boldsymbol{z}_i^{(t)}$ i $\boldsymbol{z}_o^{(t)}$, s obzirom da je na oba primijenjena sigmoidna funkcija. Gradijent po afinoj transformaciji mehanizma modulacije $\boldsymbol{z}_{\tilde{c}}^{(t)}$ slijedi iz jednadžbe (2.47).

$$\frac{\partial L}{\partial \boldsymbol{z}_{\tilde{c}}^{(t)}} = \left(\frac{\partial \tilde{\boldsymbol{c}}^{(t)}}{\partial \boldsymbol{z}_{\tilde{c}}^{(t)}}\right)^{\mathrm{T}} \frac{\partial L}{\partial \tilde{\boldsymbol{c}}^{(t)}} = \left(\frac{\partial tanh(\boldsymbol{z}_{\tilde{c}}^{(t)})}{\partial \boldsymbol{z}_{\tilde{c}}^{(t)}}\right)^{\mathrm{T}} \frac{\partial L}{\partial \tilde{\boldsymbol{c}}^{(t)}} = diag\big(1-(\tilde{\boldsymbol{c}}^{(t)})^{\boldsymbol{2}}\big) \frac{\partial L}{\partial \tilde{\boldsymbol{c}}^{(t)}} \tag{2.47}$$

Korištenjem svih prethodnih rezultata i lančanog pravila moguće je izvesti gradijent po parametrima. Zbog lakše manipulacije s različitim matricama parametara i vektorima pomaka uvedena je notacija iz izraza (2.48) i (2.49).

$$\boldsymbol{z}^{(t)} = \begin{bmatrix} \boldsymbol{z}_f^{(t)} \\ \boldsymbol{z}_i^{(t)} \\ \boldsymbol{z}_{\tilde{c}}^{(t)} \\ \boldsymbol{z}_o^{(t)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}_f \\ \boldsymbol{W}_i \\ \boldsymbol{W}_{\tilde{c}} \\ \boldsymbol{W}_o \end{bmatrix} \boldsymbol{x}^{(t)} + \begin{bmatrix} \boldsymbol{U}_f \\ \boldsymbol{U}_i \\ \boldsymbol{U}_{\tilde{c}} \\ \boldsymbol{U}_o \end{bmatrix} \boldsymbol{h}^{(t-1)} + \begin{bmatrix} \boldsymbol{b}_f \\ \boldsymbol{b}_i \\ \boldsymbol{b}_{\tilde{c}} \\ \boldsymbol{b}_o \end{bmatrix} = \boldsymbol{W}\boldsymbol{x}^{(t)} + \boldsymbol{U}\boldsymbol{h}^{(t-1)} + \boldsymbol{b} \tag{2.48}$$

$$\frac{\partial L}{\partial \boldsymbol{z}^{(t)}} = \begin{bmatrix} \dfrac{\partial L}{\partial \boldsymbol{z}_f^{(t)}} \\ \dfrac{\partial L}{\partial \boldsymbol{z}_i^{(t)}} \\ \dfrac{\partial L}{\partial \boldsymbol{z}_{\tilde{c}}^{(t)}} \\ \dfrac{\partial L}{\partial \boldsymbol{z}_o^{(t)}} \end{bmatrix} \tag{2.49}$$

U jednadžbi (2.48) sve matrice $\boldsymbol{W}_{(\cdot)}$ naslagane su u zajedničku matricu $\boldsymbol{W} \in \mathbb{R}^{4H\times D}$, sve matrice $\boldsymbol{U}_{(\cdot)}$ naslagane su u matricu $\boldsymbol{U} \in \mathbb{R}^{4H\times H}$, a vektori pomaka $\boldsymbol{b}_{(\cdot)}$ u vektor $\boldsymbol{b} \in \mathbb{R}^{4H}$. U jednadžbi (2.49) svi gradijenti po afinim transformacijama naslagani su u vektor dimenzije $4H$. Na temelju uvedene notacije izrazi za gradijente po parametrima i pomacima dani su u (2.50) do (2.52).

$$\frac{\partial L}{\partial \boldsymbol{W}^{(t)}} = \left(\frac{\partial \boldsymbol{z}^{(t)}}{\partial \boldsymbol{W}^{(t)}}\right)^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{z}^{(t)}} = \frac{\partial L}{\partial \boldsymbol{z}^{(t)}} \boldsymbol{x}^{(t)\mathrm{T}} \tag{2.50}$$

$$\frac{\partial L}{\partial \boldsymbol{U}^{(t)}} = \left(\frac{\partial \boldsymbol{z}^{(t)}}{\partial \boldsymbol{U}^{(t)}}\right)^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{z}^{(t)}} = \frac{\partial L}{\partial \boldsymbol{z}^{(t)}} \boldsymbol{h}^{(t-1)\mathrm{T}} \tag{2.51}$$

$$\frac{\partial L}{\partial \boldsymbol{b}^{(t)}} = \left(\frac{\partial \boldsymbol{z}^{(t)}}{\partial \boldsymbol{b}^{(t)}}\right)^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{z}^{(t)}} = \frac{\partial L}{\partial \boldsymbol{z}^{(t)}} \tag{2.52}$$

Algoritam BPTT-a zatim provodi izračun gradijenta po skrivenom stanju i memorijskoj ćeliji prema jednadžbama (2.53) i (2.54)

$$\frac{\partial L}{\partial \boldsymbol{h}^{(t-1)}} = \frac{\partial L}{\partial \boldsymbol{h}^{(t-1)}} + \boldsymbol{U}^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{z}^{(t)}} \tag{2.53}$$

$$\frac{\partial L}{\partial \boldsymbol{c}^{(t-1)}} = diag(1 - tanh(\boldsymbol{c}^{(t-1)})^{\boldsymbol{2}}) diag(\boldsymbol{o}^{(t-1)}) \frac{\partial L}{\partial \boldsymbol{h}^{(t-1)}} + diag(\boldsymbol{f}^{(t)}) \frac{\partial L^{(t)}}{\partial \boldsymbol{c}^{(t)}} \tag{2.54}$$

Koraci ovog postupka se ponavljaju sve dok algoritam ne dostigne početni vremenski korak. S obzirom da se radi o modelu s dijeljenim parametrima u svim koracima niza, ukupni gradijent gubitka po parametrima je jednak sumi po koracima, kao što je pokazano u (2.55).

$$\frac{\partial L}{\partial \boldsymbol{W}} = \sum_{t}^{T} \frac{\partial L}{\partial \boldsymbol{W}^{(t)}} \qquad \frac{\partial L}{\partial \boldsymbol{U}} = \sum_{t}^{T} \frac{\partial L}{\partial \boldsymbol{U}^{(t)}} \qquad \frac{\partial L}{\partial \boldsymbol{b}} = \sum_{t}^{T} \frac{\partial L}{\partial \boldsymbol{b}^{(t)}} \tag{2.55}$$

Računanje gradijenta kroz povratnu neuronsku mrežu je računalno intenzivna operacija zbog toga što se izvodi sekvencijalno. Postupak učenja moguće je ubrzati primjenom algoritma krnjeg unatražnog prostiranja pogreške kroz vrijeme (engl. *truncated backpropagation*). Ovaj algoritam ograničava broj koraka i definira razmak između dva vremenska koraka kroz koje se računa vrijednost gradijenta. Ovakvim pristupom ubrzava se proces učenja, ali učinkovitost modela ovisi o tome koliko su dobro podešene postavke algoritma.

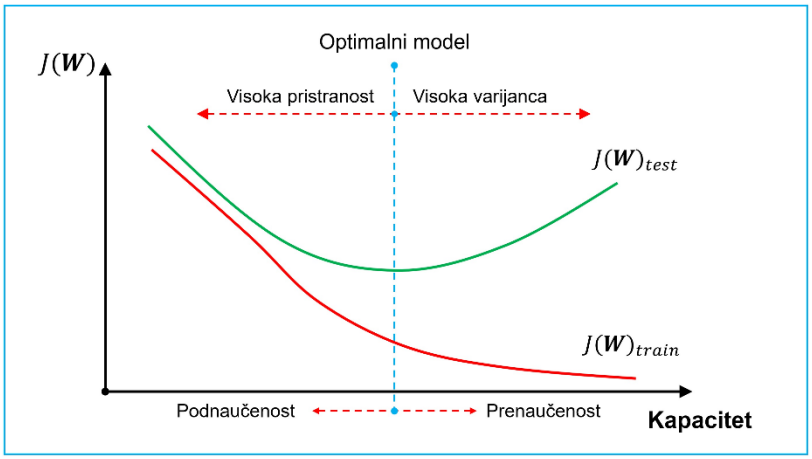
Jedno od ograničenja povratnih neuronskih mreža, bez obzira na broj slojeva, je da u određenom vremenskom koraku vrijednost izlaza ovisi samo o prethodnim vremenskim koracima i trenutnom ulazu. U slučaju kada zadatak obrade niza to dopušta, korisno je upotrijebiti informacije iz svih vremenskih koraka niza, odnosno i prošli i budući vremenski kontekst. Ova ideja istražena je u [109] u kojem je predloženo korištenje dva paralelna RNN sloja s jednakim brojem parametara, pri čemu jedan obrađuje niz u jednom smjeru, a drugi u suprotnome smjeru. Dva sloja koja rade na opisani način, omogućuju modelu da vidi cijeli niz u svakom koraku. S obzirom da su dva paralelna sloja nezavisni jedan od drugoga, za izračun gradijenta je moguće

koristiti već opisani algoritam BPTT-a. Ideja dvosmjernih mreža (engl. *bidirectional RNNs*) nije ograničena samo na jednostavni RNN, već ju je na isti način moguće koristiti i s LSTM slojem.

Nakon što su objašnjene tri glavne komponente algoritma dubokog strojnog učenja i tri tipa arhitektura, potrebno je sagledati na koji način se odabire specifična komponenta i kako se podešavaju elementi svake komponente. Svaki element algoritma DL-a, osim težina modela koje se podešavaju optimizacijskim metodama, naziva se hiperparametar (engl. *hyperparameter*). Primjeri hiperparametara su izbor arhitekture modela, npr. koliko će slojeva i neurona imati model i kako će oni biti povezani, i izbor vrste i oblika funkcije gubitka, npr. da li će funkcija imati regularizacijske članove. Odgovori na neka od ovih pitanja tema su sljedećeg odjeljka.

2.3.5 Utjecaj hiperparametara na proces učenja

Prilikom učenja modela cilj je ostvariti generalizaciju modela na nove podatke uz optimalnu razinu gubitka i rezultata na metrici učinkovitosti. Gubitak generalizacije $J(\boldsymbol{W})_{test}$ mjeri se na skupu za testiranje $\mathcal{P}_{test}$. U procesu učenja model se može naći u dva režima rada u ovisnosti od svoje složenosti, u području podnaučenosti (engl. *underfitting*) i području prenaučenosti (engl. *overfitting*) što je prikazano na slici 2.29. Za podnaučene modele se kaže da imaju visoku pristranost (engl. *bias*), što znači da je razlika između stvarne funkcije y i očekivanja modela $\mathbb{E}_{\mathcal{P}}[\hat{y}]$ velika. Do ove situacije dolazi zato jer je model prejednostavan u odnosu na stvarnu funkciju. Visoka pristranost vidljiva je na temelju visokog gubitka na skupu za učenje $J(\boldsymbol{W})_{train}$. Prenaučeni modeli su oni koji imaju veliku varijancu, koja je posljedica prevelike složenosti modela, pa osim statističkih pravilnosti koje su prisutne u podatcima za učenje, modeliraju i šum. Visoka varijanca se manifestira u vidu velike razlike između generalizacijskog gubitka $J(\boldsymbol{W})_{test}$ i gubitka učenja $J(\boldsymbol{W})_{train}$. U praksi se javljaju situacije gdje model ima i visoku pristranost i visoku varijancu. Zaključak je da se generalizacijski gubitak može rastaviti na dio uzrokovan pristranosti i dio uzrokovan varijancom, a njegova optimalna razina povezana je s odgovarajućim kapacitetom modela.



Slika 2.29 Odnos varijance i pristranosti modela s obzirom na kapacitet

Kroz optimizaciju hiperparametara se nastoji pronaći odgovarajuća složenost modela s obzirom na problem koji se rješava. Algoritmi dubokog učenja imaju pregršt hiperparametara, međutim empirijski je pokazano da nemaju svi jednak utjecaj na uspjeh učenja. U nastavku će biti napravljen pregled nekih od najvažnijih hiperparametara. Pregled je organiziran po komponentama algoritma, pri čemu će izdvojene biti regularizacijske tehnike jer one mogu biti vezane uz svaku od triju komponenti. Osim toga za svaki od hiperparametara bit će navedeno kako utječe na pristranost i varijancu modela, ovisno o vrijednostima koje poprima.

Hiperparametri modela

Broj neurona – poprima cjelobrojne vrijednosti. Veći broj neurona smanjuje pristranost, ali povećava varijancu (vidi sliku 2.29). Kada je greška učenja veća od zahtjevane, a ostale komponente modela rade ispravno, korisno je povećati broj neurona. U praksi broj neurona u početnom sloju se obično odabire tako da se ne izgube nikakve informacije uslijed premalog broja neurona u odnosu na ulaznu dimenzionalnost podataka, jer te informacije neće biti moguće povratiti. Kod potpuno povezanih slojeva, stari pristup je bio definirati različit broj neurona u svakom sloju, pri čemu su raniji slojevi imali veći broj neurona u odnosu na kasnije. Empirijski rezultati ukazuju na to da se slični ili bolji rezultati postižu uz isti broj neurona u svakom sloju. U slučaju konvolucijskih slojeva korisna praksa je imati istu veličinu jezgre u svakom sloju, ali povećavati broj jezgri s dubinom, jer se u dubljim slojevima nalaze značajke koje su bitne za klasifikaciju.

Broj slojeva – poprima cjelobrojne vrijednosti. Veći broj slojeva smanjuje pristranost i povećava varijancu (vidi sliku 2.29). Vrijede slična načela kao i za odabir broja neurona u slučaju da gubitak učenja nije na zadovoljavajućoj razini. Teorijski i empirijski rezultati

pokazali su da modeli veće dubine puno efikasnije koriste parametre u odnosu na plitke modele. Ovo znači da dublji modeli mogu modelirati funkcije koristeći eksponencijalno manji broj parametara u odnosu na plitke modele uz istu količinu podataka [14]. Smanjenjem broja slojeva, a isto vrijedi i za broj neurona, smanjuje se vrijeme trajanja iteracije učenja, ali uz cijenu rasta pristranosti.

Način povezivanja neurona – poprima diskretne vrijednosti. Ovaj izbor je vođen strukturom podataka koju model treba obraditi, npr. za obradu slika pogodno je koristiti 2D konvoluciju, za vremenske serije izbor je obično neka od povratnih neuronskih mreža ili 1D konvolucijska arhitektura, a za strukturirane podatke korisni su potpuno povezani slojevi. Što se tiče utjecaja na pristranost i varijancu, arhitekture s dijeljenim težinama smanjuju varijancu.

Ostali hiperparametri modela – izbor aktivacijske funkcije, vrste inicijalizacije težina i specijalizirani slojevi su hiperparametri koji poprimaju diskretne vrijednosti. Njihov izbor nije primarno vođen utjecajem na pristranost i varijancu, već utjecajem na efikasnost procesa učenja. Aktivacijske funkcije poput ReLU, tanh i sigmoide objašnjene su u odjeljku vezanom za arhitekture sa svojim prednostima i nedostacima. Glavna prednost ReLU funkcije je njen utjecaj na problem nestajućeg gradijenta i na bržu konvergenciju, a nedostatak da je njen negativni dio jednak nuli. Ovdje je moguće spomenuti jednu od verzija ReLU funkcije kojom se nastoji ublažiti problem negativnog dijela. ELU (engl. *exponential linear unit*) funkcija u svom negativnom dijelu nije jednaka nuli, već zavisi od dodatnog parametra u kombinaciji s eksponencijalnom funkcijom. Sporija je za izračun i kod učenja i u fazi primjene naučenog modela zbog primjene eksponencijalne funkcije, međutim prema određenim istraživanjima [110] omogućava konvergenciju u manjem broju iteracija od ReLU funkcije. Inicijalizacija težina povezana je s vrstom aktivacijske funkcije. Glorot inicijalizacija [111] koristi se u kombinaciji sa sigmoidnom i tanh funkcijom, dok je za ReLU funkciju i njene varijante dobro koristiti He inicijalizaciju [44]. Obje inicijalizacije moguće je napraviti slučajnim uzorkovanjem iz uniformne ili normalne distribucije vjerojatnosti uz odgovarajuću postavu statističkih parametara distribucije. Svrha ovih inicijalizacija je ublažavanje problema nestanka ili eksplozije gradijenta. Dodatna rješenja za probleme s gradijentom kod dubokih modela mogu biti ranije spomenuti sloj za normalizaciju grupe uzoraka, koji također može umanjiti i varijancu modela. Alternativa za smanjenje varijance može biti dropout tehnika [112], u praksi implementirana kao specijalni sloj. Ova tehnika primjenjuje se na skrivene značajke kasnijih slojeva modela na način da se u fazi učenja nasumično „ugasi“ dio skrivenih značajki prema definiranoj vjerojatnosti izbacivanja. Efekt izbacivanja je da model mora naučiti kako se nositi

sa smanjenim kapacitetom i naučiti čim deskriptivnije reprezentacije podataka. Ova tehnika se može protumačiti i kao uprosječivanje većeg broja različitih modela.

Hiperparametri optimizacijske metode

Vrsta optimizacijske metode – poprima diskretne vrijednosti. Alternative su opisane u odjeljku o optimizacijskim metodama. Dobar inicijalni izbor su metode s momentom ili ADAM metoda. Ima smisla promijeniti optimizacijsku metodu u slučaju da proces učenja stagnira tj. gubitak se ne smanjuje kroz iteracije.

Stopa učenja – poprima vrijednosti iz skupa realnih brojeva obično u rasponu od $1 \cdot 10^{-5}$ do 1 , što nije čvrsto pravilo. Radi se o jednom od najvažnijih hiperparametara. Utjecaj stope učenja na pristranost i varijancu je složeniji od svih ostalih hiperparametara. Optimalni kapacitet modela nije u općem slučaju vezan za posebno malu ili veliku stopu učenja, već ona mora biti podešena s obzirom na optimizacijski problem. Kada je stopa učenja prevelika, gradijentna metoda divergira od minimuma te gubitak učenja raste. Kada je stopa učenja premala, učenje je sporo te može trajno zaglaviti u području visokog gubitka.

Raspored stope učenja (engl. *learning rate scheduling*) – poprima diskretne vrijednosti. Ovo je alternativa fiksnoj stopi učenja, koja je posebno korisna kada se koriste stohastičke gradijentne metode. Ovi pristupi obično rade na način da se inicijalna stopa učenja smanjuje kroz iteracije ili na temelju fiksne funkcije (npr. linearno ili eksponencijalno) ili na temelju definiranog kriterija, npr. ako gubitak ne pada nekoliko uzastopnih iteracija. S druge strane, postoje i rasporedi koji ciklički povećavaju i smanjuju stopu učenja na temelju definirane funkcije [113].

Dodavanje ranog zaustavljanja (engl. *early stopping*) – poprima binarnu vrijednost. Proces učenja zaustavlja se na temelju definiranog kriterija, npr. gubitak u nekoliko uzastopnih iteracija stagnira ili raste. Ovaj pristup smanjuje varijancu, ali povećava pristranost.

Broj iteracija i epoha učenja – poprima cjelobrojne vrijednosti. Epoha predstavlja jedan prolaz svih primjera kroz model, dok se iteracija odnosi na prolaz jedne grupe primjera kroz model. Veći broj iteracija smanjuje pristranost, ali povećava varijancu. U slučaju da se koristi rano zaustavljanje odabir broja epoha nije toliko važan te može biti postavljen na proizvoljno veliku vrijednost.

Veličina grupe primjera – poprima cjelobrojne vrijednosti. Ovaj hiperparametar utječe na učinkovitost modela i vrijeme učenja, glavna prednost veće grupe je da ih se na današnjim računalnima može efikasno obraditi. Veće grupe daju bolju procjenu gradijenta.

Hiperparametri funkcije gubitka

Vrsta funkcije gubitka – poprima diskretne vrijednosti. Izbor funkcije gubitka, slično kao i izbor arhitekture, povezan je s strukturom podataka, konkretnije s tipom oznaka. Osim toga, u obzir treba uzeti i tip izlaznih vrijednosti iz zadnjeg sloja modela. Funkciju gubitka je moguće odrediti, kako je opisano u odjeljku o funkcijama gubitka, na temelju negativne log-izglednosti, pri čemu je moguće funkciji priključiti dodatne članove kojima se ograničava magnituda težina modela sa svrhom smanjenja varijance.

Regularizacija

Regularizacija podrazumijeva bilo kakve dodatke ili promjene na modelu, funkciji gubitka ili optimizacijskoj metodi napravljene sa svrhom smanjenja gubitka generalizacije, uz moguće povećanje gubitka učenja. Drugim riječima, svrha regularizacije je smanjenje varijance uz cijenu povećanja pristranosti. Funkcija gubitka se regularizira tako da joj se dodaju članovi koji penaliziraju normu vektora težina, a obično su to $L1$ ili $L2$ norma, prema izrazu (2.56) gdje je $\lambda \in [0, \infty)$ faktor kojim je određena relativna važnost oba člana jednadžbe.

$$J_{reg}(\boldsymbol{w}; \mathcal{P}) = J(\boldsymbol{w}; \mathcal{P}) + \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2 \tag{2.56}$$

Veća vrijednost λ rezultirati će određenim težinama čije su vrijednosti blizu nule za $L2$ normu, ili u slučaju $L1$ norme težinama koje su jednake nuli. Kod optimizacijske metode primjena ranog zaustavljanja može se protumačiti kao regularizacijska metoda, jer uslijed prekida procesa učenja dio težina ostaje ograničen na manjim vrijednostima. Dijeljeni parametri modela također se mogu smatrati vrstom regularizacije jer ograničavaju kapacitet modela. Od ostalih načina regularizacije modela moguće je spomenuti primjenu slojeva za normalizaciju grupe ili dropout slojeva. Zanimljivo je da se regularizacija može primijeniti i na same podatke za učenje uvođenjem slučajnog šuma ili generiranjem novih podataka na temelju distribucije postojećih. Dobar primjer su podatci u obliku slika kod kojih se različitim transformacijama (translacije, rotacije, promjene veličine) može stvoriti sasvim novi skup podataka za učenje i povećati robusnost modela na ovu vrstu varijacija. Iz ove posljednje tvrdnje slijedi zaključak da je dodavanje dodatne količine podataka za učenje pouzdan način za rješavanje problema varijance.

Postavljaju se dva pitanja: što je potrebno za traženje optimalnih hiperparametara i na koji način je sustavno moguće odrediti vrijednost hiperparametara? Dva najučestalija načina za provjeru kvalitete hiperparametara su primjena dodatnog skupa podataka i unakrsna validacija (engl.

cross-validation). Prvi način zahtjeva da se dio podataka iz skupa za učenje izdvoji za provjeru hiperparametara. Taj skup se tada naziva validacijski skup $\mathcal{P}_{val}$ koji služi za procjenu generalizacijskog gubitka uz odabrane postavke hiperparametara. Bitno je napomenuti da se uslijed ponovljenog korištenja validacijskog skupa smanjuje vjerodostojnost procjene generalizacijskog gubitka, jer model postaje prenaučen na te podatke. Kada je količina podataka za učenje premala za kreiranje validacijskog skupa, alternativa može biti stohastička podjela skupa podataka za učenje u K nepreklapajućih podskupova podataka jednake veličine, nakon čega se učenje radi na $K - 1$ podskupova, a validacija na preostalom podskupu. Ovaj postupak se ponavlja za svaki podskup, te se za procjenu generalizacijskog gubitka uzima prosjek svih K ponavljanja. Podešavanje svih hiperparametara zahtjeva da se prati gubitak učenja i gubitak na validacijskom skupu kako bi se utvrdila podnaučenost ili prenaučenost modela. Jedina iznimka je stopa učenja za koju se prati samo ponašanje gubitka učenja. Prema [14] postoje tri glavna načina traženja hiperparametara:

- Ručno traženje – koje zahtjeva da analitičar ima dobro poznavanje utjecaja pojedinog hiperparametra na kapacitet modela ili iskustvo na sličnim problemima.
- Automatsko traženje – primjenom pretraživanja po rešetci (engl. *grid search*) ili nasumičnog pretraživanja (engl. *random search*).
- Traženje putem optimizacijske metode – npr. Bayesova optimizacija.

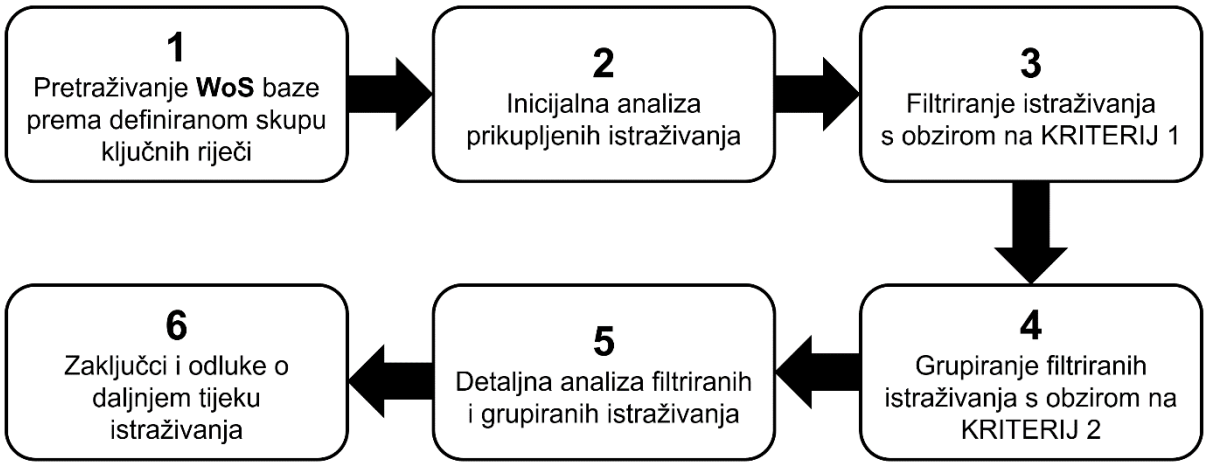
Ručno traženje hiperparametara ima smisla kada već postoje slični modeli u domeni problema, ili kada postoje ograničenja u vidu vremenskog ili tehničkog resursa. Dodatna prednost ručnog traženja je u finoj kontroli, u smislu da automatizirani algoritmi za traženje hiperparametara zahtijevaju završetak cijelog eksperimenta kako bi izračunali kvalitetu pojedinog postava. Primjenom ručnog pretraživanja lakše je detektirati raspone pojedinih hiperparametara koji ne pomažu u smanjenju generalizacijskog gubitka i odbaciti ih u ranoj fazi te na taj način uštedjeti na vremenu eksperimentiranja. No, s druge strane, krajnji cilj je razvoj algoritma koji ne treba ljudsku intervenciju, a automatizirani pristupi su korak prema tom cilju. Kada je potrebno podesiti manji broj hiperparametara razumno je koristiti pretraživanje po rešetci. Za ovaj pristup fokus treba biti na odabiru kvalitetnih raspona i točaka u tim rasponima, jer će u procesu pretraživanja sve točke biti ispitane. Pretraživanje po rešetci najbolje radi kad se ponavlja nekoliko puta, uz fino podešavanje raspona hiperparametara. U praksi je ipak bolje raditi nasumično pretraživanje. Kod nasumičnog pretraživanja potrebno je definirati odgovarajuće distribucije vjerojatnosti za pojedini hiperparametar, te pomoću njih generirati vrijednosti hiperparametara. Nasumičnim pretraživanjem se brže pronalaze dobre postavke

hiperparametara zbog toga jer nema nužno svaki hiperparametar istu važnost. Kada hiperparametri nemaju istu važnost moguće je da se pretraživanjem po rešetci za različite vrijednosti hiperparametara dobije slični gubitak, što rezultira nepotrebnim utroškom vremenskog resursa na takve eksperimente. Uz slučajno uzorkovanje vrijednosti hiperparametara vjerojatnost prethodno opisane situacije je puno manja. Primjena optimizacijskih metoda za traženje optimalnih hiperparametara je problematična iz razloga što većina hiperparametara ne poprima kontinuirane vrijednosti, stoga za njih nije moguće izračunati gradijent koji bi optimizacijska metoda mogla pratiti. Metode bazirane na Bayesovoj regresiji ovom problemu pristupaju na način da izgrade regresijski model za procjenu očekivanog gubitka na validacijskom skupu za svaki hiperparametar. Ovo je i dalje računalno intenzivno zbog potrebe za velikim brojem eksperimenata, što neke od ovih metoda pokušavaju ublažiti na način da čuvaju rezultate prijašnjih eksperimenata kako bi dovoljno rano prekinule one eksperimente koji vode prema lošim rezultatima [114].

2.4 Problem istovremenog prepoznavanja i vremenske segmentacije aktivnosti

Problem istovremenog prepoznavanja i vremenske segmentacije aktivnosti danas je relevantan u različitim domenama ljudske aktivnosti. Iako aktivno istraživačko područje, količina istraživanja, a posljedično i napredak u smjeru rješenja problema, nije na istoj razini poput nekih drugih zadataka računalnog vida. Razlozi za ovo se mogu tražiti u kombinaciji faktora, poput računarskog troška potrebnog za razvoj adekvatnih modela do toga da su rani skupovi podataka bili manji obujmom, što je negativno djelovalo na generalizacijske sposobnosti modela. Stoga ne čudi činjenica da pristupi temeljeni na dubokom strojnom učenju nisu ispunili očekivanja istraživača u smislu da nisu postigli značajno bolje rezultate od klasičnih modela strojnog učenja, kao što je slučaju u raznim drugim zadatcima računalnog vida. Međutim, unazad nekoliko godina, broj istraživanja usmjerenih na primjenu dubokog učenja za rješenje spomenutog problema počeo je rasti. Ovaj trend je logičan s obzirom da ljudske aktivnosti imaju inherentno hijerarhijsku strukturu, a pretpostavka takve strukture je temelj dubokog učenja. Nažalost, dok je u raznim industrijama prepoznat poslovni potencijal koji proizlazi iz razvijenih modela sposobnih za rješavanje ovog zadatka, to nije slučaj u proizvodnoj industriji. Ova izjava se temelji na postojanju iznimno male količine istraživanja koja se bavi rješavanjem problema prepoznavanja i vremenske segmentacije aktivnosti u kontekstu proizvodnje. U ovom odjeljku napravljena je analiza postojećih istraživanja u raznim domenama ljudske djelatnosti, pa tako i proizvodnji, kako bi se utvrdilo koji to pristupi bazirani na dubokom učenju imaju potencijal za primjenu u uvjetima proizvodne industrije.

Pregled postojećih istraživanja proveden je na temelju procedure koje je prikazana na slici 2.30.



Slika 2.30 Procedura za provedbu pregleda postojećih istraživanja

U prvoj fazi definirano je da će pregled biti proveden putem citatne baze *Web of Science* (WoS) na temelju definiranih ključnih riječi koje se nalaze u tablici 2.1. Ključne riječi su kategorizirane kao glavne i pomoćne, što znači da je inicijalni unos u tražilicu bio napravljen na temelju glavne riječi dok je rafiniranje pretrage rađeno putem pomoćne riječi. Napravljeno je nekoliko iteracija pretraživanja spomenute baze, pri čemu je zaključni prolaz kroz bazu napravljen 26.9.2019..

Tablica 2.1 Popis ključnih riječi kod pretraživanja

Glavne riječi	Action detection; Activity detection; Action recognition; Action segmentation; Activity segmentation; Action sequence; Hierarchical event detection; Motion sequence; Structured activity analysis; Task detection; Task sequence; Temporal segmentation; Workflow analysis; Workflow monitoring
Pomoćne riječi	Continuous video; Convolution; Deep learning; Human factor; Human performance; LSTM; Machine Learning; Manufacturing; Skill assessment; Time study; Video data

Prva faza rezultirala je prikupljanjem 315 istraživanja. U drugoj fazi napravljen je površinski prolaz po istraživanjima koji je uključivao pregled sažetaka i zaključaka istraživanja, nakon čega je preostalo 162 potencijalno korisna izvora u kontekstu disertacije. U trećoj fazi definiran je kriterij na temelju kojeg je napravljeno fino filtriranje preostalih istraživanja. Kriterij su zadovoljila ona istraživanja koja pokušavaju riješiti problem „*action detection*"-a ili „*action segmentation*"-a primjenom dubokog strojnog učenja na podatcima u obliku RGB video zapisa te su objavljena nakon 2011. godine. Iznimka od ovog kriterija su istraživanja koja su

napravljena u proizvodnji. Na temelju prethodno spomenutog kriterija, isključena su istraživanja koja problemu pristupaju bez korištenja dubokog učenja, barem u jednom dijelu vlastitog algoritma. Također, isključena su istraživanja koja koriste samo podatke prikupljene senzorima dubine ili nosivim senzorima. Nakon treće faze preostalo je 31 istraživanje koje se bavi definiranim problemom u različitim domenama ljudskih aktivnosti te 7 radova koji su usmjereni na problem proizvodnih aktivnosti. Filtrirana istraživanja grupirana su s obzirom na pristup rješavanju problema u grupu *„action detection"* ili *„action segmentation"* te su unutar grupa kronološki organizirana, dok radovi iz proizvodne domene čine posebnu grupu, jer se neki od njih bave isključivo prepoznavanjem aktivnosti. Rezultati analize u petoj fazi strukturirani su na način da sadrže informacije o korištenom algoritmu za pripremu ulaznih podataka te o vrstama modela za finalnu klasifikaciju i/ili regresiju. Osim toga izneseni su globalni zaključci o prednostima i nedostatcima pojedinih pristupa.

2.4.1 O terminologiji i oznakama u pregledu istraživanja

U pregledu istraživanja korištene su različite kratice, od kojih su neke objašnjene već ranije, a uvedene su i neke nove s ciljem da pregled algoritama koji se javljaju u literaturi bude što pregledniji i koncizniji. U tablici 2.2 su objašnjenja korištenih kratica u pregledu.

Tablica 2.2 Značenje oznaka u pregledu istraživanja

Oznaka	**Značenje**
1D/2D/3D CNN	Konvolucijska n. mreža, za obradu definirane dimenzionalnosti primjera
biLSTM	Dvosmjerna povratna n. mreža s LSTM slojevima
CRF	Model uvjetnih slučajnih polja (engl. *conditional random fields*)
FA	Fiksni algoritam
FNN	Unaprijedna n. mreža
GRU	Povratna n. mreža s GRU slojevima
LSTM	Povratna n. mreža s LSTM slojevima
OF	Optički tok (*engl. optical flow*)

Jedan od novih termina koji se javlja u gornjoj tablici je model uvjetnih slučajnih polja (CRF). Radi se o modelu iz statistike i strojnog učenja u području grafičkih modela, koji primjenu nalazi kod problema sa strukturom rešetke, slično kao i CNN modeli. Sljedeći novi termin vezan je uz pojam fiksnog algoritma (FA). Svaki algoritam koji nema mogućnost učenja parametara na temelju ulaznih podataka primjenom neke optimizacijske metode, u pregledu je klasificiran kao fiksni algoritam. Završno, novi termin je i optički tok (OF). Optički tok je algoritam iz

područja računalnog vida koji primjenu nalazi u obradi video zapisa [115]. Primjenom ovog algoritma moguće je procijeniti kretanje različitih vizualnih objekata između sličica u video zapisu. Procjena OF-a se obično provodi rješavanjem varijacijskog računa [116], dok ga je danas moguće i učiti primjenom CNN modela [115]. Ovo znači da bi se OF isto mogao klasificirati kao fiksni algoritam, ali zbog njegove učestalosti u literaturnim izvorima dodijeljena mu je zasebna oznaka. U kontekstu strojnog učenja, optički tok je jedna od najpopularnijih i najkorisnijih ručno dizajniranih značajki u analizi video zapisa. Glavni nedostatak procjene OF-a je visok računarski trošak koji može biti reduciran, ali uz cijenu smanjene točnosti procjene.

2.4.2 Pregled istraživanja iz grupe „*action segmentation*“

Istraživanja iz ove grupe bave se problemom prepoznavanja i segmentacije aktivnosti na način da zadatak strojnog učenja formuliraju kao problem klasifikacije svake sličice u video zapisu, pri čemu su granice i ukupno trajanje aktivnosti određeni implicitno. U ovu grupu klasificirano je 15 istraživanja. U tablici 2.3 nalazi se pregled korištenih pristupa za pripremu ulaznih podataka te modela za finalnu predikciju iz literature.

Tablica 2.3 Istraživanja iz grupe „action segmentation“

Izvor	Priprema ulaznih podataka	Model za klasifikaciju
[16]	2D CNN + 1D CNN + FA	biLSTM
[117]	2D CNN	LSTM
[118]	3D CNN	LSTM + FA
[24]	2D CNN + FA	1D CNN
[119]	2D CNN + OF	1D CNN + biLSTM
[120]	2D CNN + OF	FA
[121]	3D CNN + FA	3D CNN + FA
[122]	2D CNN	LSTM + FA
[123]	2D CNN + OF	FNN + CRF
[124]	2D CNN + FA	LSTM + 3D CNN
[125]	2D CNN + OF	1D CNN
[126]	2D CNN	LSTM
[70]	3D CNN + OF	1D CNN
[127]	2D CNN + OF	1D CNN
[23]	3D CNN + OF	1D CNN

Kako bi smanjili dimenzionalnost ulaznih podataka i definirali deskriptivne značajke, pristupi iz ove grupe najčešće su koristili 2D konvolucijske modele za obradu slika u fazi pripreme podataka. Korišteni su modeli poput VGG16 [16,24,119,120,123,125], VGG19 [117], AlexNet [126] i ResNet50 [122,127] uz primjenu koncepta prijenosa znanja i finog podešavanja, što znači da su spomenuti modeli bili prednaučeni na ImageNet skupu podataka i fino podešeni na njihovim podatcima. S obzirom da 2D CNN modeli mogu opisati samo prostorne značajke na temelju pojedinačnih slika, u istraživanjima su često korištene značajke dobivene izračunom optičkog toka [119,120,123,125,127] ili sličnih, ali računarski manje zahtjevnih, algoritama [16,24,124] s ciljem ugrađivanja ograničenog vremenskog konteksta u značajke. Alternativno, u određenim radovima su korištene 3D konvolucijske mreže poput C3D [118,121] koje istodobno uzimaju u obzir i prostornu i kratkoročnu vremensku komponentu prisutnu u podatcima, dok neki koriste kombinaciju 3D CNN-a i optičkog toka [23,70]. Kritika pristupa temeljenih na korištenju 3D CNN-a je da se radi o računarski skupom i neefikasnom načinu pripreme podataka [127]. Ovo se može pokazati kroz usporedbu, npr. 2D CNN modela poput ResNet50 koji ima 50 slojeva i 25 milijuna parametara, u odnosu na 3D CNN model poput C3D koji ima 11 slojeva i 80 milijuna parametara. Zbog prethodno navedenih razloga, autori koji su koristili 3D CNN modele za izvlačenje značajki, nisu radili fino podešavanje, što u krajnjoj liniji može utjecati na manju učinkovitost modela za klasifikaciju.

Najčešće korišteni modeli za klasifikaciju u ovoj grupi istraživanja su bili LSTM i 1D CNN modeli. Tako je, primjerice, u radu [16] korišten dvosmjerni LSTM kako bi se iskoristile informacije s oba kraja vremenskog niza. U ovom članku spomenuta je još jedna zanimljiva ideja vezana za pripremu ulaznih podataka. Autori su kao ulaz biLSTM-a koristili značajke koje su kombinacija 4 zasebna 2D CNN modela, gdje su dvije mreže kao ulaz dobivale podatke koji pokrivaju cijelu dimenziju slike, a preostale dvije naučene su na podatcima koji su fokusirani na osobu. Na ovaj način su pokušali prikupiti informacije o aktivnostima koje ovise o lokaciji i onima koje ne ovise o lokaciji na kojoj se aktivnost izvodi. U članku [24] predložena je arhitektura koja je bazirana na hijerarhiji 1D konvolucijskih slojeva, s dva glavna dijela modela, pisačem (engl. *coder*) i čitačem (engl. *encoder*), pri čemu jedan dio koristi slojeve sažimanja, a drugi dio koristi naduzorkovanje (engl. *upsampling*). Cilj primjene ovakve arhitekture je postizanje sposobnosti hvatanja dugoročnih vremenskih pravilnosti u podatcima. Prednost ovog modela, u odnosu na model iz [16], je u puno bržem učenju (30 puta) na istom skupu podataka, pri čemu je učinkovitost bolja ili usporediva ovisno o korištenoj metrici. Nedostatak ovog modela je potreba za ekstremnim poduzorkovanjem videozapisa, konkretno model koristi jednu

do dvije sličice po sekundi, a tako smanjena vremenska rezolucija može utjecati na točnost modela. Istraživanje [119] povezuje ideje iz prethodna dva članka, na način da koristi arhitekturu iz članka [24], ali u drugom dijelu modela umjesto 1D CNN-a koristi biLSTM. Ove izmjene rezultirale su neznatno većom točnosti, ali puno sporijim učenjem. Rad [122] predlaže istovremeno učenje modela za izvlačenje značajki i modela za klasifikaciju, pri čemu je prvi model ResNet50, a drugi je LSTM, no uslijed toga moraju raditi na isječcima iz video zapisa trajanja 2 sekunde. Rezultati su neznatno bolji, nego kada su prvi i drugi model učeni zasebno, ali je vrijeme učenja integriranog modela dulje. Pokušali su koristiti i dublje modele za izvlačenje značajki (npr. ResNet101), ali tada uslijed računalnih ograničenja nisu mogli učiti integrirani model. Istraživači u radu [123] koriste dvije paralelne unaprijedne neuronske mreže za obradu značajki dobivenih na temelju pojedinačnih slika i optičkog toka. Klasifikacija je rezultat povezivanja ove dvije grane u zajednički vektor, pri čemu su ispitali dva pristupa povezivanja, težinskim prosjekom ili maksimalnim rezultatom jedne od grana. Finalno, kako bi dobili čim preciznije granice aktivnosti koriste model uvjetnih slučajnih polja. U članku [125] predlažu unapređenja u odnosu na slična istraživanja poput [24,119] na način da model ima dvije paralelne vremenske grane. Grana koja je odgovorna za analiziranje videa u punoj vremenskoj rezoluciji (svaka sličica), te služi za finu segmentaciju akcija, nazvana je rezidualnom granom. Druga grana koristi sažimanje i naduzorkovanje, te obrađuje video zapis na različitim vremenskim rezolucijama sa svrhom poboljšanja točnosti klasifikacije svake sličice. Ove dvije grane spojene su primjenom deformabilnog vremenskog rezidualnog modula koji koristi deformabilne konvolucijske filtere. Slabost ovog modela je u tome da jako loše segmentira kraće akcije koje se nalaze između dva segmenta koji imaju duže trajanje. Model iz [126] baziran je na LSTM-u, ali uz korištenje Monte Carlo dropout-a i aktivnog učenja kako bi dobili duboku Bayesovu mrežu, što omogućava korištenje manje količine podataka za učenje u odnosu na konkurentske metode. Istraživanje [70] primjenjuje ideju iz [24] o više slojeva dilatiranih vremenskih konvolucija uz udvostručavanje faktora dilatacije u svakom sloju. Implementiran je pristup koji koristi informacije iz prošlih i budućih vremenskih koraka, te rezidualne veze kako bi bilo olakšano učenje. Za razliku od modela iz [24], ovaj model može raditi na punoj vremenskoj rezoluciji video zapisa. Ideja dilatiranih konvolucija primijećena je i u modelu iz [127]. U članku [23] je predloženo daljnje unapređenje modela iz [70] na način da se razdvajaju faze generiranja i rafiniranja predikcija iz modela. Modul za generiranje predikcija služi za generiranje inicijalnih predikcija na temelju značajki svake sličice, a modul za rafiniranje preuzima ulogu podešavanja predikcija kroz niz etapa. Pristup klasifikaciji koji je predložen u [120] razlikuje se od svih ostalih na način da se klasifikacija provodi izravno iz

značajki izračunatih modelom za pripremu podataka primjenom njihovog fiksnog algoritma. U spomenutom radu, aktivnost je modelirana kao vremenski prozor proizvoljne duljine uz pretpostavku da postoji jedna sličica koja predstavlja početak i kraj, te više njih koje predstavljaju sredinu akcije. Prilikom predikcije, akcija je lokalizirana kroz traženje strukturirane maksimalne sume (engl. *structured maximal sum*) niza koji sadrži početak, sredinu i kraj, te ima najbolju točnost po sličici. Opisani algoritam temeljen je na dinamičkom programiranju.

Funkcija gubitka koja je najčešće korištena kod pristupa iz ove grupe je unakrsna entropija. Određeni istraživači smatraju da modeli koji klasificiraju svaku sličicu rezultiraju predikcijama koje imaju prekomjernu segmentiranost[9], pa sukladno tome predlažu aditivni dodatak gubitku unakrsne entropije koji penalizira prekomjernu segmentaciju s ciljem izglađivanja predikcija [23,70]. Dodavanje aditivnih članova funkciji gubitka istraženo je i u članku [117], gdje se tvrdi da za algoritam koji vidi sve veći dio nekog vremenskog segmenta ulaznog niza povjerenje u klasifikaciju ne smije padati, odnosno u najmanju ruku mora se monotono održavati na istoj razini. S time na umu, predložen je dodatak koji penalizira model ako se naruši monotonost povjerenja u klasifikaciju. Povezano s formulacijom aktivnosti kao strukturirane radnje, rad [120] predlaže adekvatnu funkciju gubitka koja je u korespondenciji s time da akcija ima početak, sredinu i kraj.

Najčešće primjenjivane optimizacijske metode kod ovih istraživanja su bile ADAM metoda i stohastička gradijentna metoda s momentom i definiranim faktorom smanjenja stope učenja kroz iteracije.

Ostali algoritamski dodatci koji su predloženi u radovima iz grupe *„action segmentation“* odnose se na obradu izlaza iz modela s ciljem povećanja točnosti predikcije. U članku [122] predložen je algoritam PKI (engl. *prior knowledge information*), koji koristi domensko znanje o problemu kako bi poboljšao prediktivnu točnost modela. U konkretnom slučaju, radi se o prepoznavanju aktivnosti kod medicinskog operativnog zahvata gdje se koristi struktura i redoslijed koraka u zahvatu s ciljem povećanja točnosti modela. Slične ideje korištene su i u radu [118] s ciljem izglađivanja (engl. *smoothing*) izlaza modela. Zanimljivo je primijetiti da se kod pristupa iz oba spomenuta istraživanja u fazi pripreme podataka nije koristila informacija o vremenskom kontekstu primjenom optičkog toka ili 3D CNN-a.

[9] Prepoznaju veći broj aktivnosti u odnosu na stvarni broj prisutan u oznakama.

2.4.3 Pregled istraživanja iz grupe „*action detection*“

Kod radova iz ove grupe količina aktivnosti od interesa u video zapisu je relativno mala u odnosu na pozadinske aktivnosti, te se zadatak strojnog učenja formulira kao problem regresije i klasifikacije. Regresijom se nastoje odrediti početak i kraj aktivnosti, a vrsta aktivnosti je određena klasifikacijom. U ovom dijelu analizirano je 16 istraživanja, prema tablici 2.4. Kod *„action detection“* grupe istraživanja, finalna predikcija zahtjeva dva koraka. U prvom koraku definirani model generira prijedloge vremenskih segmenata (engl. *proposals*). Prijedlozi predstavljaju one dijelove video zapisa koji sadrže aktivnosti od interesa neovisno, od njihove točne klase. Drugim riječima, svrha prijedloga je razlučiti aktivnosti od pozadinske klase. Za određivanje točne oznake klase odgovorna je druga faza, odnosno dodatni model koji osim predikcije klase radi i fino podešavanje granica aktivnosti.

Tablica 2.4 Istraživanja iz grupe „action detection“

Izvor	**Priprema ulaznih podataka**	**Model**	
		Generiranje prijedloga	**Klasifikacija i lokalizacija**
[128]	3D CNN + FA	3D CNN	3D CNN
[129]	3D CNN	LSTM	FNN
[130]	2D CNN + OF	1D CNN	FNN
[71]	3D CNN + FA	GRU + FA	FNN
[131]	3D CNN + FA	GRU + FA	FNN
[132]	3D CNN	FA	FNN
[133]	3D CNN	FA	FNN
[134]	3D CNN	1D CNN	FNN
[135]	3D CNN	3D CNN	FNN
[136]	2D CNN + OF	FA	FNN
[137]	3D CNN + OF	FA	3D CNN+LSTM
[138]	3D CNN + OF	1D CNN	FNN
[139]	2D CNN + OF	1D CNN	FNN
[140]	3D CNN	3D CNN	3D CNN
[141]	2D CNN + OF	1D CNN +FA	FNN
[72]	3D CNN + OF	3D CNN	FNN

Izvlačenje značajki se kod ove grupe istraživanja uglavnom izvodilo primjenom 3D CNN modela poput C3D [71,128,129,131–134,140] ili Res3D [137], a u manjoj mjeri primjenom 2D CNN [130] ili 3D CNN [138] uz optički tok. Razlozi za ovu odluku mogu se tražiti u činjenici da se kod ove grupe pristupa, za razliku od prethodne, neće provoditi klasifikacija pojedinačnih sličica, već će se razmatrati granice segmenata. Stoga je u redu da se vremenska rezolucija smanji, uz pretpostavku da će ključne informacije biti zadržane u kreiranim značajkama dobivenima 3D konvolucijom.

Kao što je ranije napisano, finalna predikcija se kod pristupa iz ove grupe obavlja na temelju dvije faze koje se uglavnom provode odvojeno, faze generiranja prijedloga i faze klasifikacije i završne lokalizacije granica segmenata. U istraživanju [128] predložen je pristup koji koristi tri 3D CNN model, gdje prvi generira prijedloge, drugi ih klasificira, a treći podešava granice segmenata. Radi se o računarski neefikasnom pristupu koji je zasnovan na vremenskim prozorima. Vremenski prozori su definirani u fazi pripreme podataka, a prije ulaska u prvi model, primjenom fiksnog algoritma koji rezultira velikim preklapanjem prozora različitih rezolucija, što znači da je više puta potrebno obraditi iste elemente video zapisa. Pristupi predloženi u radovima [129] i [130] nastoje riješiti nedostatak prethodnog istraživanja na način da je samo jednom potrebno proći kroz cijeli video zapis. U [129] se prijedlozi različitih vremenskih rezolucija generiraju u jednom prolazu primjenom LSTM modela, dok se u [130] za istu stvar koristi 1D CNN model. Daljnje unapređenje pristupa iz [129] dano je u članku [71], koji također ima sposobnost analize video zapisa bez potrebe da obrađuje preklapajuće vremenske prozore. Predloženi algoritam ima tri faze. U prvoj fazi, nakon što su izvučene značajke iz C3D modela, primjenjuje se PCA[10] kako bi se dodatno smanjile dimenzije ulaznih značajki, nakon čega slijedi obrada niza primjenom GRU modela koji daje prijedloge za svaki vremenski korak. Za učenje GRU povratnog modela osmislili su poseban postupak kako bi osigurali njegovu generalizaciju na duge vremenske nizove. Finalno, izlazni model daje povjerenje u svaki generirani prijedlog. Nedostatak ovog pristupa je prepoznat kod obrade aktivnosti kratkog trajanja koje slijede jedna iza druge, što bi mogao biti problem kod primjene ove metode u studiju vremena. Daljnja nadogradnja modela iz [71] opisana je u istraživanju [131] u kojemu se želi integrirati korak generiranja prijedloga i njihove klasifikacije po uzoru na slične pristupe iz domene detekcije objekata na slikama. Kako bi to ostvarili, predložili su dvije paralelne grane GRU slojeva, pri čemu je jedna grana odgovorna za generiranje prijedloga, a druga za njihovu klasifikaciju. Ova podjela zadataka osigurana je primjenom

[10] PCA (engl. *principal component analysis*) – analiza glavnih sastavnica

odgovarajuće funkcije gubitka koja usmjerava specijalizaciju svake grane. Izlazi iz obje grane su kasnije spojeni u jedan vektor, nakon čega slijedi finalna predikcija modela. U radovima [132,133] istraživači su usmjereni na komponentu modela vezanu za generiranje prijedloga koji imaju visok odziv[11] (engl. *recall*), a istodobno su računarski efikasni. Predlažu kaskadni regresijski pristup u kojem se granice aktivnosti podešavaju i u fazi generiranja prijedloga i u fazi konačne predikcije. U članku [135] predlažu pristup koji koristi samo 3D CNN modele, te je učinkovit jer se u fazi generiranja prijedloga i finalne klasifikacije koriste jedne te iste značajke. Kako bi model mogao raditi predikciju prijedloga različitih vremenskih dimenzija uvode koncept sidra (engl. *anchor*) po analogiji na slične modele iz domene detekcije objekata. Sidra su unaprijed definirani prozori različitih vremenskih rezolucija uniformno raspoređeni po vremenskim lokacijama. Za učenje nužno je dati oznaku svakom sidru. Predikcija granica segmenata je napravljena kao pomak u odnosu na centar i duljinu za svako sidro u svakoj vremenskoj poziciji. Prema njihovoj argumentaciji, ovo povećava točnost prijedloga jer je to bolji pristup od izravne regresije vremenskih granica. Rad [137] glavnu ideju temelji na pretpostavci da ljudi kada rade detekciju koriste pristup koji sadrži dvije faze: detekciju s finom granuliranosti, a zatim detekciju sa grubom granuliranosti. Predlažu model koji će koristiti prethodno navedeni pristup na način da će to koristiti u fazi generiranja vremenskih prijedloga, te u fazi klasifikacije.

Funkcija gubitka se kod pristupa iz ove grupe uglavnom sastoji od dvije komponente, unakrsne entropije kao gubitka za klasifikaciju, te regresijskog gubitka na temelju kojeg se prate odstupanja granica segmenata u odnosu na stvarne granice aktivnosti. Inovacije po pitanju funkcija gubitka su u analiziranim istraživanjima vezane uz regresijsku komponentu. U radu [128] je predložena funkcija gubitka temeljena na metrici preklapanja segmenta i stvarne oznake, na način da se prati omjer između presjeka i unije predloženog i stvarnog segmenta. Istraživanje [133] umjesto uobičajenog regresijskog gubitka temeljenog na kvadratnom odstupanju, koristi L1 normu. Adaptivno mijenjanje važnosti pojedine komponente gubitka u procesu učenja istraženo je u radu [131]. U navedenom istraživanju, na početku učenja komponente funkcije gubitka vezane za prijedloge i njihovu klasifikaciju imaju veću važnost, dok se u kasnijim fazama naglasak prebacuje na finalnu klasifikaciju i lokalizaciju granica aktivnosti. Dominantno korištene optimizacijske metode kod ove grupe istraživanja u potpunosti su iste kao kod prethodne grupe. Za ovu grupu pristupa karakteristično je da se nakon

[11] Odziv je metrika učinkovitosti, koju je u ovom kontekstu moguće tumačiti kao broj aktivnosti od interesa koje je model prepoznao u odnosu na ukupan broj aktivnosti od interesa

finalne predikcije uvijek koristi algoritam potiskivanja prijedloga (engl. *non-maximum suppression–NMS*). Ovo je rađeno iz razloga što modeli generiraju veliki broj segmenata koji se preklapaju sa stvarnim aktivnostima, pa je potrebno za svaki pojedini stvarni segment odbaciti sve one koji se preklapaju s njim, osim najboljeg. Drugim riječima, samo najbolji segmenti predstavljaju predikciju modela. Osim ovog algoritma, od dodatnih algoritama koji se često koriste u ovoj grupi pristupa moguće je izdvojiti algoritam vremenskog selektivnog diskriminativnog pretraživanja [137]. Ovaj algoritam služi za generiranje kvalitetnih prijedloga te je alternativa generiranju prijedloga primjenom modela poput 1D CNN ili LSTM-a.

2.4.4 Pregled istraživanja iz domene proizvodnje

U odjeljku 1.2 uvodno su opisana istraživanja iz ove grupe, dok će u ovom djelu detaljnije biti opisani konkretni problemi koji su rješavani u okviru istraživanja te pristupi pripremi ulaznih podataka i razvoju modela. Istraživanja prezentirana ovdje većinom se oslanjaju na klasične modele strojnog učenja, u kombinaciji s ručno definiranim značajkama. Fokus dijela istraživanja je na prepoznavanju pojedinačnih aktivnosti, ali ne i na određivanju njihova vremena trajanja. Kao i kod prethodne dvije grupe, radovi su organizirani kronološki.

U istraživanju [142] tema je nadzor procesa zavarivanja koji uključuje sedam radnih aktivnosti. Cilj koji su autori definirali je, na video zapisu trajanja od dvije do pet minuta, prepoznati sve aktivnosti koje se događaju, pri čemu je ideja klasificirati svaku sličicu. Podatci su pripremljeni algoritmom razvijenim u okviru istraživanja. Algoritam su nazvali „mreža pokreta“ (engl. *motion grid*), pri čemu su razvili koncept „lokalnih monitora pokreta“ (engl. *local motion monitor*) koji predstavljaju elemente unutar mreže pokreta. Konkretno, računaju razliku između susjednih piksela u blizini definiranog piksela u trenutnoj i prošloj sličici, sumiraju sve razlike, te nakon toga tu sumiranu razliku uspoređuju s vrijednosti praga, ako je vrijednost veća od praga dobiva oznaku 1, inače dobiva oznaku -1. Takav postupak ponavlja se za sve odabrane piksele i oni čine mrežu pokreta, pri čemu se susjedstva mogu preklapati. Nakon toga mreža pokreta se zapisuje kao stupac vektor za sličicu. Osim toga, prilikom kreiranja ulaznih značajki koristili su i domensko znanje u vidu toga što su znali da se aktivnosti mogu odvijati samo na određenim lokacijama jer su na njima objekti rada, pa su se mogli fokusirati samo na te piksele kod svake slike. Značajke izračunate ovim pristupom pokazale su se značajno robusnijima za primjenu u proizvodnim uvjetima, u odnosu na značajke definirane na temelju putanja (engl. o*bject trajectories*) i praćenja (engl. *tracking*) koje se susreću s poteškoćama u slučaju složene pozadine slike. Kao model korišten je ESN (engl. *echo state network*) [143] koji je jedna od inačica povratne neuronske mreže. Karakteristika ovog modela je da se u fazi učenja ažuriraju

samo težine vezane za izlaze, dok su težine između skrivenih stanja i ulaza te skrivenih stanja trenutnog i prethodnog vremenskog koraka slučajno inicijalizirane te ostaju takve. Ovaj algoritam je biran umjesto skrivenog Markovljevog modela, jer HMM zahtjeva dobro definirana stanja što nije uvijek moguće u industriji, npr. zbog mogućih permutacije u redoslijedu aktivnosti. Glavna prednost primjene ESN-a je brzina učenja, s obzirom da je jedino potrebno učiti parametre izlaza, a u slučaju da je izlaz linearna kombinacija moguće je provesti učenje primjenom normalne jednadžbe. Nedostatak je slabija reprezentativna moć u odnosu na dublje neuronske mreže.

Rad [144] bavi se istim problemom na istom skupu podataka kao i istraživanje [142] te su kod njega korištene iste ulazne značajke. U radu je testirano nekoliko vrsta modela temeljenih na kombinaciji genetskog algoritma i HMM-a, uz ulaze koji su bili s dvije različite kamere ili koji su bili fuzija podataka s dvije kamere. Pristup su podijelili u dva koraka. U prvom koraku rade segmentaciju kontinuiranog video zapisa. Koriste HMM za prepoznavanje završetka pojedine aktivnosti, pri čemu je ulaz u niz određen klizajućim prozorom unaprijed definirane veličine. Osim toga, koriste domensko znanje kako bi modelirali trajanje pojedinih aktivnosti primjenom modela Gaussove mješavine (engl. *Gauss mixture model*). Kombinacijom ovih dvaju pristupa određuju vjerojatnost završetka aktivnosti. U drugom koraku rade prepoznavanje aktivnosti unutar definiranih segmenata. Navode dvije bitne vjerojatnosti na temelju kojih je moguće odrediti klasu aktivnosti u segmentu: vjerojatnost i-te aktivnosti u ciklusu te uvjetna vjerojatnost i-te aktivnosti u slučaju da joj je prethodila j-ta aktivnost. Uvjetnu vjerojatnost računaju unaprijed na temelju domenskog znanja o slijedu aktivnosti u procesu. Razvili su funkciju cilja koja je suma težina ovih dviju vjerojatnosti za svaku aktivnost koja se dogodila u nizu. Za ovaj korak iskoristili su genetski algoritam kako bi pronašli približno rješenje mogućih nizova aktivnosti. Nedostatak ovog pristupa je izuzetno visok udio domenskog znanja potrebnog za dizajn ovakvog algoritma, što ograničava njegov transfer na različite procese.

Istraživači se u radu [145] bave problemom zavarivanja iz istog proizvodnog pogona kao i [142,144], međutim ovaj puta odabiru proces koji se sastoji od 12 aktivnosti. Kod razvoja značajki koriste algoritam PCH [146] (engl. *pixel change history*) uz korištenje domenskog znanja o lokacijama na slici gdje se mogu događati aktivnosti. Ovako kreirane značajke omogućile su određivanje početka i završetka aktivnosti. To je napravljeno na sljedeći način: pojava nekoliko uzastopnih sličica na kojima je vektor značajki bio jednak nul vektoru za promatrano područje interesa, definirana je kao završetak aktivnosti. Pojava vektora različitog od nul vektora nakon niza nul vektora, predstavlja početak aktivnosti. Kao model korišten je

pristup zasnivan na HMM-u, i to 12 zasebnih HMM-ova za svaku od aktivnosti. Ponovno su koristili proces modeliranja u više faza. Nakon prve faze, u kojoj su definirani početci i završetci aktivnosti, provedena je klasifikacija.

Istraživanje [26] je u ideji slično radu [145], ali je provedeno na istim aktivnostima kao istraživanje [142,144] koristeći samo podatke snimljene kamerom postavljenom iznad glave izvoditelja aktivnosti. Korišten je model temeljen na HMM-u.

Fokus rada [7] je na procesu prepoznavanja četiri tipa pokreta korištenih kod ručne montaže pri čemu su podatci prikupljeni u laboratorijskim uvjetima. U ovom članku pokazano je i kako izračunati vrijeme trajanja pojedine aktivnosti. Korištene su manualno razvijene značajke procesom koji se sastojao od tri faze. U prvoj fazi provodi se izvlačenje ključnih sličica pristupom koji su autori nazvali *„dynamic key frame extraction“*, koji funkcionira na taj način da prva sličica niza postaje svojevrsni obrazac koji se uspoređuje sa svakom sljedećom sličicom na temelju L2 norme. Kada je razlika veća od praga, nova sličica postaje ključna sličica te se uspoređuje sa svakom sljedećom. U drugom koraku radili su prepoznavanje područja od interesa s fokusom na ruke izvoditelja aktivnosti. U posljednjoj fazi koristili su prethodno spominjani SIFT algoritam kako bi generirali značajke koje će biti ulaz klasifikacijskog algoritma. Za klasifikaciju je korišten algoritam stroja potpornih vektora.

U izvoru [18] iskorišten je isti skup podataka kao u istraživanjima [142,144]. Za pripremu podataka korišten je algoritam MHI [147] (engl. *motion history image*) kojim se smanjuje vremenska rezolucija video zapisa, iz razloga što su koristili nepreklapajuće segmente video zapisa kod izračuna značajki. Kako bi smanjili dimenziju ulaznih značajki koristili su diskretnu transformaciju funkcijom kosinusa (engl. *discrete cosine transform*). Spomenuti korak napravljen je na način da je svaka „slika“ dobivena iz MHI algoritma podijeljena u prostorne blokove dimenzija 32×32 piksela, a zatim se na blokove primjenjivala transformacija kosinusa pri čemu je svaki blok predstavljen s 16 dominantnih vrijednosti. Kao model je korištena 2D konvolucijska neuronska mreža. Ovim pristupom dobiveni su najbolji rezultati po pitanju preciznosti (86%) i odziva (89%) u odnosu na pristupe iz prethodnih radova. Međutim, navedeni rezultati su ostvareni na način da je iz ulaznih podataka u potpunosti izbačena pozadinska klasa aktivnosti, dok su prethodni pristupi u obzir uzimali i pozadinske aktivnosti. Razlog za ovu odluku je ograničenje MHI algoritma u vidu generiranja korisnih značajki za pozadinsku klasu aktivnosti.

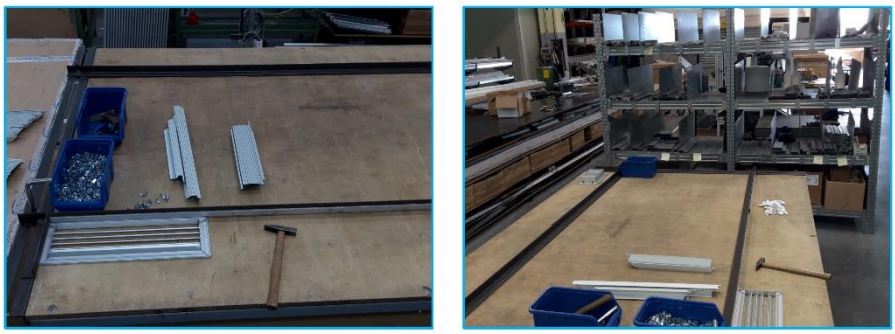
Pristup prikupljanju podataka i razvoju modela koji je opisan u radu [6] bitno se razlikuje od dosad navedenih radova iz ove grupe. Ovo istraživanje usmjereno je na proces lakiranja u automobilskoj industriji koji se sastoji od 9 aktivnosti, pri čemu su podatci prikupljeni senzorom dubine. Shodno tome, zadatak istraživanja je definiran kao klasifikacija svake „slike“ dubine. Prikupljeni podatci obuhvaćaju 7 radnih sati, pri čemu pola tih sati izvodi jedan radnik, a pola drugi, te je ukupno zabilježeno 350 radnih ciklusa. Ulazne značajke su bile pozicije zglobova koje su dobivene izravno iz senzora dubine, a osim toga korištene su i izvedene značajke poput kutova između zglobova i udaljenosti između ruku. Istraženi su pristupi nadziranog i nenadziranog učenja, što znači da u procesu učenja nisu korištene oznake povezane s ulaznim primjerima. Korišteno je nekoliko različitih vrsta modela kako bi se usporedile njihove učinkovitosti, poput nadziranog i nenadziranog HMM-a i algoritma k-sredina (engl. *k-means*). Najbolji rezultati su ostvareni nadziranim HMM pristupom (točnost 70%), dok je od nenadziranih pristupa najbolji rezultat ostvario nenadzirani HMM (točnost 65%). Prednost predloženog pristupa je da se izbjegava ručno obilježavanje podataka, međutim ostvareni rezultati nisu konkurentni nadziranim pristupima.

2.4.5 Zaključak

Ove spoznaje dovode do zaključka da su potrebna nova istraživanja u domeni analize ljudskih aktivnosti iz video zapisa usmjerena na primjenu algoritama dubokog strojnog učenja bez primjene manualno kreiranih ulaznih značajki, posebice u segmentu proizvodnje koji je slabo zastupljen u literaturi. Nužno je ispitati primjenjivost dubokih modela na realnim podatcima prikupljenima u proizvodnji iz čega slijedi da je potrebno prikupiti takav tip podataka u dostatnoj količini. Fokus istraživanja treba biti problem istovremenog prepoznavanja i vremenske segmentacije koji je još uvijek nedovoljno istražen u različitim vrstama ljudske djelatnosti uključujući i proizvodnju.

3. PRIKUPLJANJE PODATAKA IZ REALNOG PROIZVODNOG PROCESA

Prikupljeni skup podataka odnosi se na proces ručne montaže metalnih rešetki HVAC[12] sustava. Podatci su prikupljani tijekom sedam radnih dana u periodu od 20.11.2019. do 17.3.2020.. Snimanje je provedeno u realnim proizvodnim uvjetima što ih razlikuje od postojećih skupova podataka iz literature u domeni istovremenog prepoznavanja i vremenske segmentacije aktivnosti iz video zapisa, koji su obično prikupljani ili u kontroliranim uvjetima ili se odnose na zapise ljudskih aktivnosti iz svakodnevnog života [17,148].



Slika 3.1 Lokacija prikupljanja podataka

Opažanja iz ovog skupa podataka sadrže interakcije između izvoditelja aktivnosti, radnog objekta i alata. Aktivnosti su izvodila četiri različita operatera kako bi se obuhvatila varijabilnost uzrokovana ljudskim faktorom. Oznake različitih operatera u uzorku su O1, O2, O3 i O4.

Slika 3.2 Četiri operatera u uzorku

[12] Sustav za grijanje, ventilaciju i klimatizaciju (engl. ***H**eating **V**entilation **A**ir **C**onditioning*)

Radne aktivnosti montaže provedene su na tri različita tipa metalnih rešetki koje se razlikuju s obzirom na dimenzije proizvoda. Unutar uzorka tipovi proizvoda su obilježeni oznakama T1, T2 i T3, prema slici 3.3.

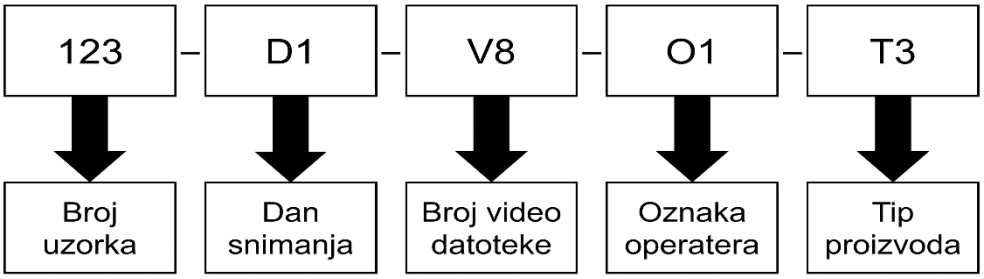
Slika 3.3 Tri tipa proizvoda u uzorku

Video zapisi su prikupljeni za tri različita kadra snimanja: kadar iznad glave (HE), kadar fokusiran na ruke operatera (Fokus) i kadar s lijeve bočne strane operatera (Fine), što je prikazano na slici 3.4. Snimanje procesa iz više kadrova je motivirano ciljem utvrđivanja utjecaja kadra snimanja na učinkovitost modela. Izbor kadra HE i Fokus inspiriran je literaturnim izvorom [149] u kojem je navedeno da je proces učenja moguće ubrzati korištenjem slika niže rezolucije, međutim to za posljedicu nosi gubitak važnih vizualnih detalja. Kompromisno rješenje je korištenje različitih kadrova niže rezolucije uz fokus na objekt rada. U okviru disertacije nije korišten kadar Fine iz razloga što je jedan od operatera ljevoruk te je uslijed toga zaklanjao vidno polje kamere u procesu rada.

Slika 3.4 Tri kadra snimanja uzorka

Samo snimanje izvedeno je primjenom dvije akcijske kamere GoPro Hero7, na rezoluciji od 1920×1080, uz frekvenciju od 30 sličica po sekundi u MP4 formatu. Po završetku snimanja napravljena je inicijalna obrada video zapisa. Ova obrada uključivala je smanjenje rezolucije video zapisa prikupljenih iz oba kadra. Kadar HE smanjen je na rezoluciju 320×192, a kadar Fokus smanjen je na rezoluciju 320×160. Nadalje, napravljeno je poduzorkovanje video zapisa s frekvencijom od 5 sličica po sekundi. Ove aktivnosti formatiranja podataka poduzete su iz nekoliko praktičnih razloga: a) kako bi se ubrzalo učenja modela, b) kako bi modeli mogli biti naučeni primjenom postojeće infrastrukture i 3) poduzorkovanje je opravdano jer postoji

značajna redundantnost između susjednih sličica u video zapisu. Sljedeća faza pripreme podataka bila je izrezivanje video zapisa u kraće isječke. Kriteriji kod rezanja video zapisa bili su da svaki isječak mora obuhvaćati sve aktivnosti procesa montaže, ali mora obuhvaćati i pozadinske aktivnosti te ne smije biti trajanja duljeg od dvije minute. Dva su razloga za ovu odluku. Prvi razlog vezan je uz domenu istraživanog problema, u smislu da problem istovremenog prepoznavanja i segmentacije aktivnosti podrazumijeva video zapise koji uključuju veći broj aktivnosti, pri čemu su neke pozadinske. Drugi razlog je praktične prirode te je ponovno povezan s ograničenjem tehničkih resursa i brzinom učenja, koje uvjetuju količinu eksperimenata koju je moguće provesti. Rezanjem prikupljenih video zapisa formirano je 620 isječaka ukupnog vremena trajanja od 10 sati, što ujedno i predstavlja veličinu uzorka korištenog u ovom istraživanju. Svaki video obilježen je prema strukturi prikazanoj na slici 3.5.



Slika 3.5 Struktura oznake video zapisa

Sljedeća faza pripreme podataka bila je označavanje vrsta aktivnosti u uzorku i njihovog trajanja kako bi se mogao koristiti nadzirani pristup strojnom učenju. U procesu montaže prepoznato je devet radnih aktivnosti potrebnih za sastavljanje proizvoda i njegovo odlaganje. Kako bi označavanje aktivnosti bilo provedeno što konzistentnije razvijena su pravila označavanja koja precizno definiraju trenutak početka i završetka aktivnosti, kao što je pokazano u tablici 3.1.

Tablica 3.1 Kriteriji kod označavanja aktivnosti u uzorku

A 1: Formiranje kuta		
		
Prva stranica u ruci		Barem jedna ruka više nije u kontaktu sa stranicama

A 2: Umetanje i učvršćivanje kopče – lijeve gornje

Kopča je ruci

Čekić je ispušten iz ruke

A 3: Umetanje lamela

Prva lamela je u ruci

Posljednja lamela je umetnuta

A 4: Postavljanje poprečne stranice

Stranica je u ruci

Barem jedna ruka više nije u kontaktu sa stranicama

A 5: Postavljanje uzdužne stranice

Stranica je u ruci

Barem jedna ruka više nije u kontaktu sa stranicama

A 6: Odlaganje gotovog proizvoda

Proizvod je u ruci

Proizvod je izvan vidnog polja kamere

Označavanje je napravljeno na dva načina. Prvi način označavanja (tablica 3.2) odgovara formulaciji problema kakva se susreće kod istraživanja koja spadaju u grupu „*action detection*“ pristupa. Ovaj način označavanja eksplicitno definira vrijeme početka i kraja aktivnosti, dok sve ostalo predstavlja pozadinsku klasu aktivnosti koja nema nikakvu specijalnu oznaku.

Tablica 3.2 Primjer oznaka za „action detection“ definiciju problema kod 1_D1_V1_O1_T1

Activity_ID	Start_time	End_time	Start_frame	End_frame
1	1.0	5.0	6	26
2	5.2	11.6	27	59
3	12.8	22.2	65	112
4	23.6	28.8	119	145
7	29.0	35.0	146	176
5	35.8	38.4	180	193
8	38.6	42.6	194	214
9	42.8	52.0	215	261
6	52.2	53.6	262	269

Drugi način označavanja (tablica 3.3) slijedi definiciju problema koja odgovara istraživanjima iz grupe *„action segmentation“* pristupa. Kod ovog načina označavanja svakoj sličici je pridružena odgovarajuća oznaka aktivnosti, uključujući i oznaku pozadinske aktivnosti. U ovom uzorku oznaka pozadinske aktivnosti je 0. Vrijeme trajanja aktivnosti je ovakvim načinom označavanja implicitno određeno te ga je moguće izračunati na temelju korištene vremenske rezolucije. Vremenska rezolucija povezana je s brojem sličica u sekundi kod video zapisa. U konkretnom slučaju, vremenska rezolucija je 0,2 sekunde, što znači da je razlika između dvije susjedne sličice dvije desetinke sekunde.

Tablica 3.3 Primjer oznaka prema „action segmentation“ definiciji problema

Video_ID	**Oznake**
1_D1_V1_O1_T1	0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2 ,2,2,2,2,2,2,2,2,2,2,2,2,2,0,0,0,0,0,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3, 3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,0,0,0,0,0,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4 ,4,4,4,4,4,4,4,4,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,0,0,0,5,5,5, 5,5,5,5,5,5,5,5,5,5,5,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,9,9,9,9,9,9,9,9,9,9,9,9,9,9 ,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,6,6,6,6,6,6,6,6,0,0,0

U tablici 3.3 moguće je primijetiti da većina sličica sadrži aktivnosti od interesa, dok uzorci u skupovima podataka poput [17,148] uglavnom sadrže pozadinske aktivnosti (70 % pozadine) te je u videu obično prisutna jedna aktivnost koja se ponavlja u različitim trenutcima u vremenu. Dodatno, odlika uzorka prikupljenog u procesu montaže je da su objekti rada prisutni gotovo cijelo vrijeme u kadru te se izgled pozadine mijenja kroz video zapise što ovaj skup čini izazovnim kod modeliranja. Također, razlike između pojedinih klasa aktivnosti su često minimalne, stoga je određene aktivnosti moguće raspoznati jedino na temelju vremenskog konteksta što zahtjeva od modela da imaju sposobnost uhvatiti dugoročne vremenske zavisnosti između aktivnosti.

Završni korak pripreme podataka uključivao je podjelu uzorka u skup za učenje, validaciju i testiranje na način da je 480 opažanja u skupu za učenje, dok skupovi za validaciju i testiranje sadrže po 70 opažanja. Podjela u ova tri skupa napravljena je slučajnim uzorkovanjem, ali uz stratifikaciju kako je pokazano u tablici 3.4. Ovo znači da su podatci uzorkovani na način da udio slojeva populacije iskazan u cjelovitom uzorku bude reflektiran i u skupovima za učenje,

validaciju i testiranje[13]. Stratifikacija je napravljena prema operateru i tipu proizvoda. Ovaj pristup za cilj ima osigurati bolju generalizaciju modela.

Tablica 3.4 Stratificirana podjela uzorka u tri skupa podataka

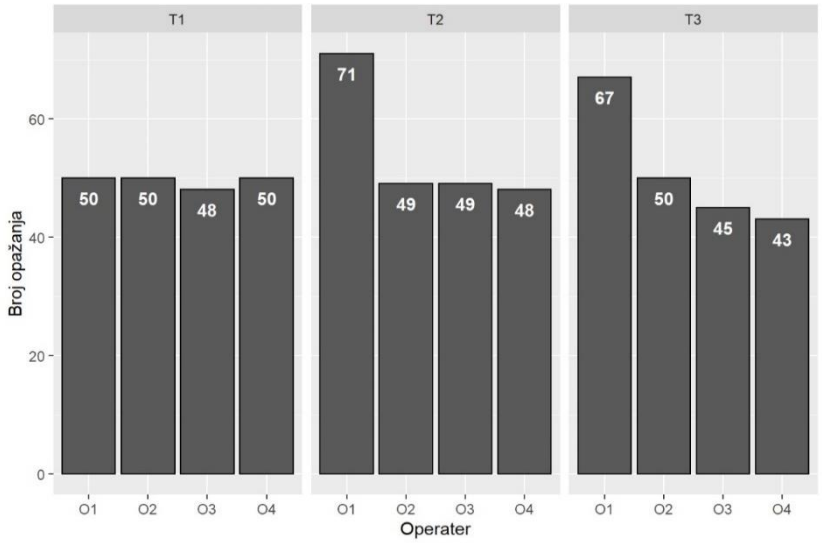
Sloj populacije	**Uzorak [%]**	**Učenje [%]**	**Validacija [%]**	**Testiranje [%]**
O1_T1	8.06	8.12	7.14	8.57
O1_T2	11.45	11.46	11.43	11.43
O1_T3	10.81	10.83	10.00	11.43
O2_T1	8.06	7.92	8.57	8.57
O2_T2	7.90	7.92	8.57	7.14
O2_T3	8.06	7.92	8.57	8.57
O3_T1	7.74	7.92	7.14	7.14
O3_T2	7.90	7.92	8.57	7.14
O3_T3	7.26	7.29	7.14	7.14
O4_T1	8.06	7.92	8.57	8.57
O4_T2	7.74	7.92	7.14	7.14
O4_T3	6.94	6.88	7.14	7.14

Pregledom tablice vidljivo je da su udjeli pojedinih slojeva otprilike jednaki u sva tri skupa podataka, pri čemu prate udjele prisutne u cijelom uzorku. Detaljnija statistička analiza svojstava prikupljenog skupa podataka tema je sljedećeg poglavlja.

[13] Npr. udio opažanja u kojima operater O1 radi na proizvodu T1 treba biti sličan u sva tri podskupa cijelog uzorka

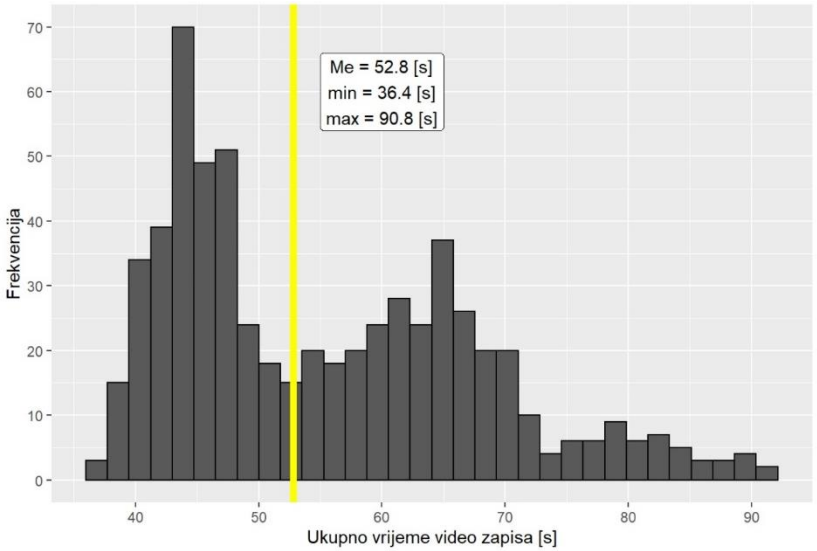
4. STATISTIČKA ANALIZA UZORKA

Cilj analize je bio utvrditi utjecaj operatera i tipa proizvoda na trajanje ukupnog procesa montaže i na pojedine radne aktivnosti. Ove informacije će kasnije pomoći u tumačenju učinkovitosti modela, posebno s aspekta prepoznavanja mogućih uzroka pogrešnih predikcija. Prilikom oblikovanja uzorka cilj je bio imati ujednačene količine opažanja po operaterima i tipovima proizvoda, što je vidljivo iz slike 4.1.



Slika 4.1 Broj opažanja po operateru za svaki tip proizvoda

Tipično vrijeme trajanja video zapisa (vidi sliku 4.2) je izraženo medijanom te iznosi 52,8 sekundi. Vrijeme trajanja video zapisa varira u ovisnosti od operatera i vrste proizvoda te je stoga minimalno trajanje video zapisa 36,4 sekunde, a maksimalno 90,8 sekundi.



Slika 4.2 Raspodjela ukupnog vremena trajanja video zapisa

U prvom dijelu analize fokus će biti na ukupnom vremenu montaže, a u nastavku na vremenima pojedinih aktivnosti.

4.1 Analiza ukupnog vremena izvođenja procesa montaže

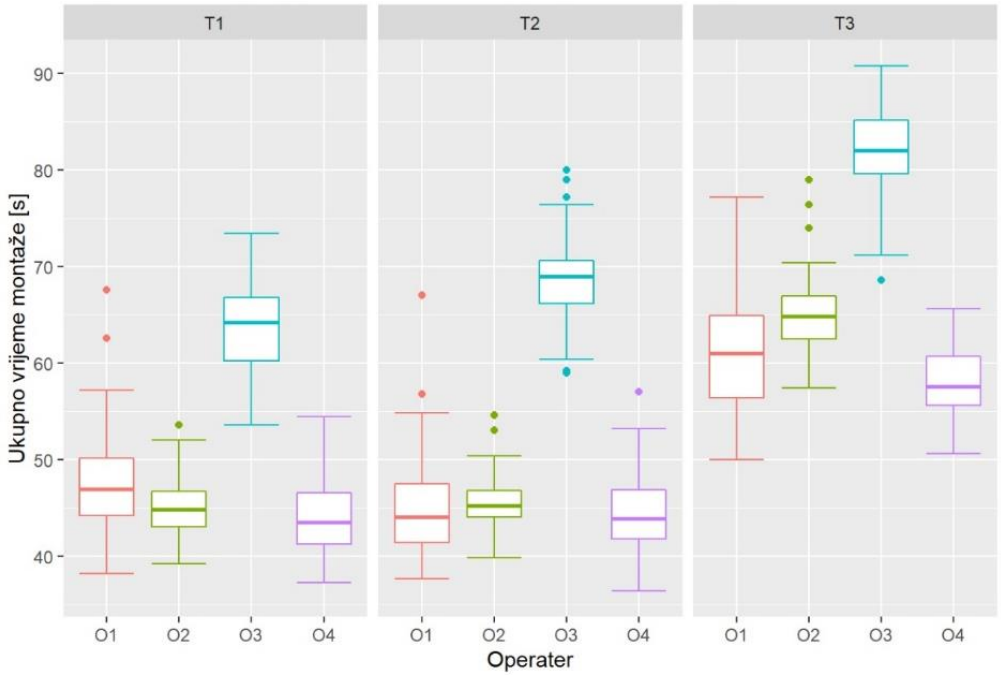
Ukupno vrijeme montaže mjereno je od početka prve aktivnosti do završetka posljednje aktivnosti. Iz razloga što video zapis može započeti i završiti s pozadinskom aktivnosti postoje razlike u vremenu trajanja montaže i video zapisa. Nadalje, i između pojedinih radnih aktivnosti može doći do pojave pozadinskih aktivnosti. Pojava pozadinskih aktivnosti u procesu montaže posljedica je različitih rasipanja i smetnji koje se mogu dogoditi u radu operatera te bi trebale biti predmet zahvata unapređenja. U tablici 4.1 prikazana su prosječna vremena radnih aktivnosti i ukupnog vremena montaže s obzirom na tip proizvoda i operatera. Ako se ova vremena protumače rječnikom metodologije poput lean menadžmenta [150], onda bi se za ukupno vrijeme montaže moglo reći da uključuje aktivnosti koje donose vrijednost, aktivnosti koje ne donose vrijednost ali su trenutno neophodne za izvođenje procesa te čisti gubitak, dok je iz vremena radnih aktivnosti isključen čisti gubitak.

Tablica 4.1 Pregled prosječnih vremena radnih aktivnosti i montaže

Tip proizvoda	**Operater**	**Prosječno vrijeme radnih aktivnosti [s]**	**Prosječno ukupno vrijeme montaže [s]**	**Udio radnih aktivnosti u montaži [%]**
T1	O1	42.6	47.7	89.3
T1	O2	40.2	45.0	89.4
T1	O3	57.9	63.5	91.2
T1	O4	40.6	44.3	91.6
T2	O1	40.3	45.0	89.7
T2	O2	41.1	45.5	90.4
T2	O3	62.8	68.5	91.7
T2	O4	41.0	44.4	92.3
T3	O1	56.2	61.3	91.7
T3	O2	60.5	65.1	92.9
T3	O3	75.1	81.9	91.7
T3	O4	54.5	58.0	94.0

U kontekstu disertacije, bitno je tumačenje koje proizlazi iz računanja razlike ovih dvaju prosječnih vremena. Spomenuta razlika daje dobru procjenu prosječnog udjela pozadinskih aktivnosti u video zapisima. Kako su video zapisi većinom ispunjeni aktivnostima od interesa, donesena je odluka da će se kod izrade modela zadatak istovremenog prepoznavanja aktivnosti i vremenske segmentacije formulirati kao problem klasifikacije svake sličice što odgovara *„action segmentation“* pristupu.

Napravljena je usporedba ukupnog vremena montaže po operateru i tipu proizvoda kako je pokazano boxplot grafom sa slike 4.3, na kojem je kao mjera položaja korišten medijan, a kao mjera rasprostiranja interkvartilni raspon. Razlog korištenja medijana i interkvartilnog raspona kao statističkih pokazatelja leži u činjenici da je Shapiro-Wilk test normalnosti distribucije [151] ukupnih vremena montaže za 12 podskupova podataka[14] ukazao na to da raspodjela nije normalno distribuirana za četiri podskupa, na razini povjerenja od 95%.



Slika 4.3 Usporedba ukupnog vremena montaže po operateru i tipu proizvoda

U nastavku je napravljeno testiranje razlike ukupnog vremena izvođenja montaže između operatera po svakom tipu proizvoda primjenom Kruskal-Wallis[15] testa [152]. Rezultati sva tri provedena Kruskal-Wallis testa potvrdili su da postoje značajne razlike u vremenu izvođenja

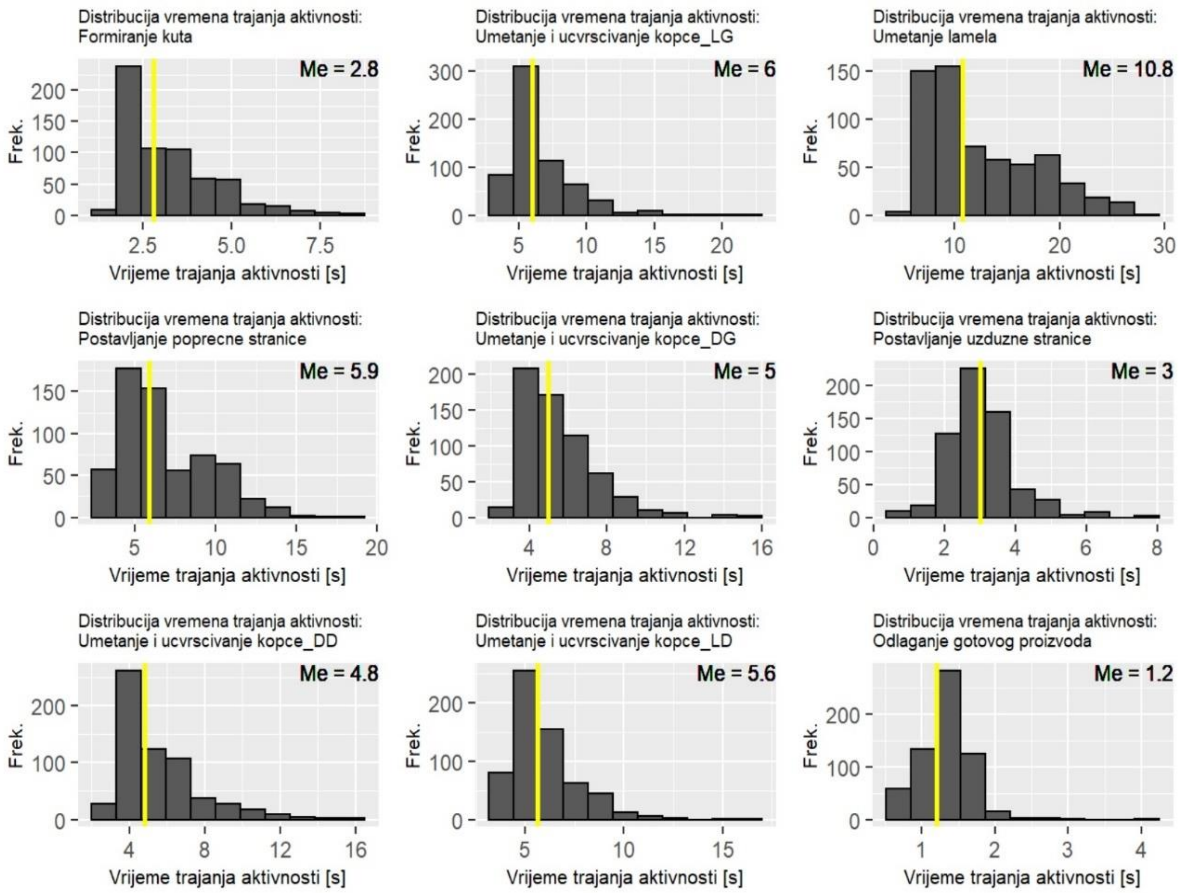
[14] Četiri operatera i tri tipa proizvoda

[15] Ovo je alternativa jednosmjernoj analizi varijance (ANOVA) u slučaju da se podatci ne pokoravaju normalnoj distribuciji

montaže između barem jednog para operatera po svakom tipu proizvoda na razini povjerenja od 95%. Pretpostavka je da najveći utjecaj na ovakav rezultat ima operater O3. Stoga je napravljen još jedan krug testiranja pri čemu je iz testa isključen operater O3. Novi testovi pokazali su da samo u slučaju proizvoda T2 ne postoje statistički značajne razlike u vremenu izvođenja montaže. Drugim riječima, u slučaju proizvoda T2 operateri O1, O2 i O4 u prosjeku rade istom brzinom, dok za proizvode T1 i T3 to nije slučaj.

4.2 Analiza vremena izvođenja pojedinih aktivnosti

Vrijeme izvođenja pojedine aktivnosti varira, na što između ostalog utječu različiti operateri i tipovi proizvoda. Distribucija vremena izvođenja pojedinih aktivnosti, neovisno od tipa operatera i tipa proizvoda, prikazana je na slici 4.4, pri čemu je posebno naznačena vrijednost medijana vremena aktivnosti.



Slika 4.4 Raspodjela vremena trajanja pojedine aktivnosti

U ovom dijelu disertacije zasebno je analizirana svaka od devet radnih aktivnosti montaže. Dva su glavna produkta analize pojedinih aktivnosti.

1. Vizualna usporedba vremena izvođenja aktivnosti

U ovom dijelu generirani su boxplot grafovi vremena izvođenja radne aktivnosti sa svrhom usporedbe operatera po pojedinom tipu proizvoda.

2. *Testiranje razlika u vremenu izvođenja aktivnosti*

Na temelju vizualne usporedbe vremena izvođenja aktivnosti proizlaze dva pitanja:

- Postoje li značajne razlike između različitih operatera u vremenu izvođenja iste aktivnosti na istom tipu proizvoda?
- Postoje li značajne razlike u vremenu izvođenja iste aktivnosti na različitim tipovima proizvoda od strane jednog operatera?

Kako bi se ispitale ove dvije hipoteze korištena je sljedeća metodologija za svaku aktivnost:

a) Testiranje normalnosti podataka vremena izvođenja aktivnosti koji su grupirani po tipu proizvoda, aktivnosti i operateru, za što je korišten Shapiro-Wilk test. Svrha ovog koraka je pomoć u odabiru odgovarajućih testova za usporedbu više uzoraka i parova uzoraka. Konkretnije, bilo je potrebno odlučiti hoće li biti korišteni parametarski ili neparametarski testovi. Shapiro-Wilk testovi su pokazali da je moguće odbaciti hipotezu da podatci slijede normalnu distribuciju.

b) Usporedba razlika između više uzoraka[16] primjenom Kruskal-Wallis testa. Ovaj test je odabran na temelju rezultata Shapiro-Wilk testa. Hipoteza H0 za ovaj test glasi da ne postoje statistički značajne razlike između grupa, a H1 da postoji statistički značajna razlika barem između jednog para grupa.

c) Usporedba parova uzoraka jednostranim Mann-Whitney testom [153]. Ovaj test je također izabran na temelju rezultata Shapiro-Wilk testa. Hipoteza H0 za ovaj test glasi da ne postoji statistički značajna razlika između dvije grupe, a H1 da postoji statistički značajna razlika između dvije grupe.

Svi testovi su provedeni na razini povjerenja od 95%, uz Bonferroni korekciju [154] kako bi se ublažili problemi uslijed višestrukih usporedbi na istim uzorcima. U nastavku rada dan je sažetak rezultata[17] za svaku aktivnost, a koji uključuje prethodno spomenute produkte analize.

[16] Uzorak u ovom kontekstu predstavlja dio opažanja koja pripadaju u specifičnu grupu s obzirom na faktore operatera i tipa proizvoda. Npr., jedan uzorak su sva opažanja za radnu aktivnost 8 operatera O2 na proizvodu T3.

[17] Detalji dostupni na: https://github.com/Miha87/phd_mg

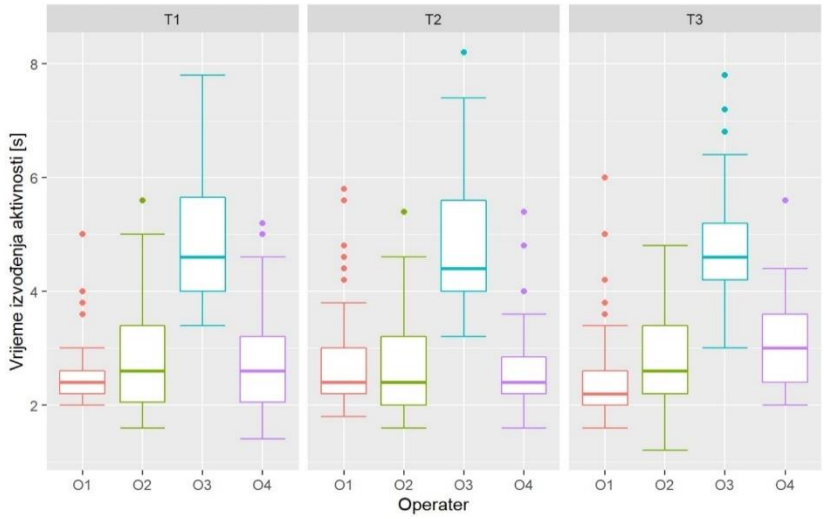
4.2.1 1. Aktivnost: Formiranje kuta

Rezultati testiranja razlika u vremenu izvođenja 1. aktivnosti između različitih operatera na istom tipu proizvoda dani su u tablici 4.2.

Tablica 4.2 Kruskal-Wallis test razlika u vremenu izvođenja 1. aktivnosti na istom proizvodu

Proizvod	Rezultat testa (α=0,05)
T1	H1
T2	H1
T3	H1

Zaključak 1: Uspoređivanjem svih operatera po pojedinom tipu proizvoda potvrđeno je da postoji značajna razlika u vremenu izvođenja 1. aktivnosti kod svih tipova proizvoda.



Slika 4.5 Usporedba vremena izvođenja 1. aktivnosti po operateru i tipu proizvoda

Graf sa slike 4.5 sugerira da je razlog spomenutih razlika najvjerojatnije operater O3, stoga je napravljeno dodatno testiranje u kojem taj operater nije bio uključen.

Tablica 4.3 Kruskal-Wallis test razlika u vremenu izvođenja 1. aktivnosti bez operatera O3

Proizvod	Rezultat testa (α=0,05)
T1	H0
T2	H0
T3	H1

Zaključak 2: Kada je iz testiranja isključen operater O3 značajna razlika u brzini izvođenja 1. aktivnosti prisutna je samo u slučaju proizvoda T3. Mann-Whitney testom je utvrđeno da je ova razlika posljedica toga što operater O1 radi brže u odnosu na preostala dva operatera.

Test razlika u vremenu izvođenja 1. aktivnosti od strane istog operatera na različitim tipovima proizvoda prikazan je u tablici 4.4.

Tablica 4.4 Kruskal-Wallis test razlika u vremenu izvođenja 1. aktivnosti od strane istog operatera na različitim tipovima proizvoda

Operater	Rezultat testa (α=0,05)
O1	H0
O2	H0
O3	H0
O4	H1

Zaključak 3: 1. aktivnost gotovo svi operateri izvode isto neovisno o vrsti proizvoda, jedino je kod O4 uočena razlika u brzini izvođenja ove aktivnosti. Mann-Whitney testom je utvrđeno da je ova razlika posljedica razlika u izvođenju aktivnosti na proizvodima T2 i T3.

Kako je na boxplot grafu sa slike 4.5 u slučaju operatera O1 uočena najveća količina vrijednosti izvan raspona ostatka podataka (engl. *outliers*), napravljena je dodatna analiza video zapisa s operaterom O1.

Zaključak 4: Pokazalo se da su odstupanja posljedica toga što su poprečne stranice potrebne za formiranje kuta bile previše udaljene od operatera. Ovo ukazuje na moguća poboljšanja u vidu organizacije i rasporeda materijala na radnom mjestu.

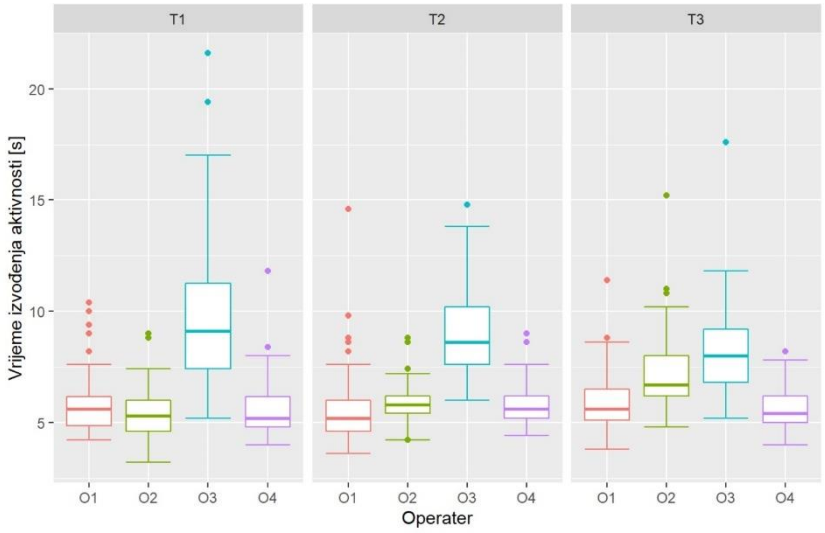
4.2.2 2. Aktivnost: Umetanje i učvršćivanje kopče – lijeve gornje

Rezultati testiranja razlika u vremenu izvođenja 2. aktivnosti između različitih operatera na istom tipu proizvoda dani su u tablici 4.5.

Tablica 4.5 Kruskal-Wallis test razlika u vremenu izvođenja 2. aktivnosti na istom proizvodu

Proizvod	Rezultat testa (α=0,05)
T1	H1
T2	H1
T3	H1

Zaključak 1: Uspoređivanjem svih operatera po pojedinom tipu proizvoda potvrđeno je da postoji značajna razlika u brzini izvođenja 2. aktivnosti kod svih tipova proizvoda.



Slika 4.6 Usporedba vremena izvođenja 2. aktivnosti po operateru i tipu proizvoda

Graf sa slike 4.6 sugerira da je razlog spomenutih razlika najvjerojatnije operater O3, stoga je napravljeno dodatno testiranje u kojem taj operater nije bio uključen.

Tablica 4.6 Kruskal-Wallis test razlika u vremenu izvođenja 2. aktivnosti bez operatera O3

Proizvod	Rezultat testa (α=0,05)
T1	H0
T2	H1
T3	H1

Zaključak 2: Kada je iz testiranja isključen operater O3 značajna razlika u brzini izvođenja 2. aktivnosti prisutna je u slučaju proizvoda T2 i T3. Mann-Whitney testom je potvrđeno da su ove razlika posljedica toga što operater O1 radi značajno brže od operatera O2 za proizvod T2, a u slučaju proizvoda T3 i O1 i O4 rade brže od operatera O2.

Test razlika u vremenu izvođenja 2. aktivnosti od strane istog operatera na različitim tipovima proizvoda prikazan je u tablici 4.7

Tablica 4.7 Kruskal-Wallis test razlika u vremenu izvođenja 2. aktivnosti od strane istog operatera na različitim tipovima proizvoda

Operater	Rezultat testa (α=0,05)
O1	H0
O2	H1
O3	H1
O4	H0

Zaključak 3: 2. aktivnost operateri O1 i O4 izvode isto, neovisno o vrsti proizvoda. Kod operatera O2 i O3 uočena je razlika u brzini izvođenja ove aktivnosti, ovisno o vrsti proizvoda.

Kako su kod operatera O3 uočene vrijednosti izvan raspona ostatka podataka kod proizvoda T1 (vidi sliku 4.6), napravljena je dodatna analiza.

Zaključak 4: U slučaju operatera O3 odstupanja su posljedica činjenice da je aktivnost odužena zbog toga jer postoje fizički problemi kod umetanja kopče.

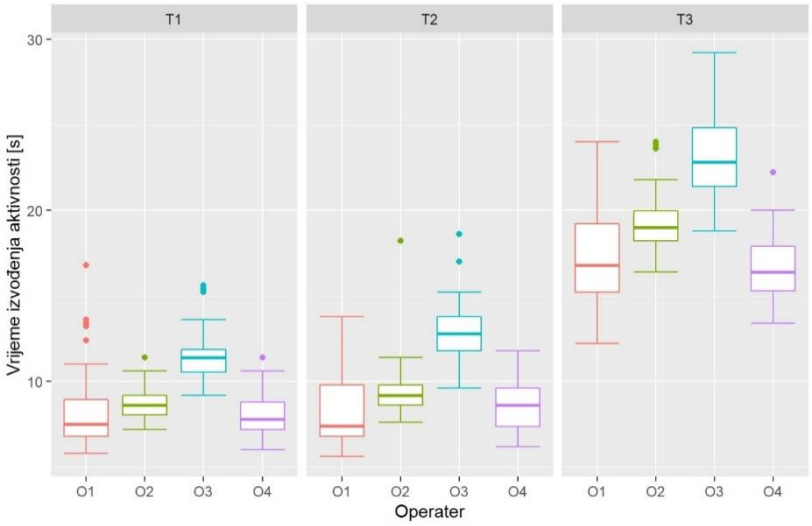
4.2.3 3. Aktivnost: Umetanje lamela

Rezultati testiranja razlika u vremenu izvođenja 3. aktivnosti između različitih operatera na istom tipu proizvoda dani su u tablici 4.8.

Tablica 4.8 Kruskal-Wallis test razlika u vremenu izvođenja 3. aktivnosti na istom proizvodu

Proizvod	Rezultat testa (α=0,05)
T1	H1
T2	H1
T3	H1

Zaključak 1: Uspoređivanjem svih operatera po pojedinom tipu proizvoda potvrđeno je da postoji značajna razlika u brzini izvođenja 3. aktivnosti kod svih tipova proizvoda.



Slika 4.7 Usporedba vremena izvođenja 3. aktivnosti po operateru i tipu proizvoda

Graf sa slike 4.7 sugerira da je razlog spomenutih razlika najvjerojatnije operater O3, stoga je napravljeno dodatno testiranje u kojem taj operater nije bio uključen.

Tablica 4.9 Kruskal-Wallis test razlika u vremenu izvođenja 3. aktivnosti bez operatera O3

Proizvod	Rezultat testa (α=0,05)
T1	H1
T2	H1
T3	H1

Zaključak 2: Kada je iz testiranja isključen operater O3 značajna razlika u brzini izvođenja 3. aktivnosti prisutna je i dalje u slučaju svih proizvoda. U slučaju sva tri tipa proizvoda operateri O1 i O4 rade statistički značajno brže od operatera O2 prema Mann-Whitney testu.

Test razlika u vremenu izvođenja 3. aktivnosti od strane istog operatera na različitim tipovima proizvoda prikazan je u tablici 4.10.

Tablica 4.10 Kruskal-Wallis test razlika u vremenu izvođenja 3. aktivnosti od strane istog operatera na različitim tipovima proizvoda

Operater	Rezultat testa (α=0,05)
O1	H1
O2	H1
O3	H1
O4	H1

Zaključak 3: Kod 3. aktivnosti brzina svih operatera značajno je različita jer je proizvod T3 većih dimenzija te su stoga svi operateri sporiji na tome proizvodu.

4.2.4 4. Aktivnost: Postavljanje poprečne stranice

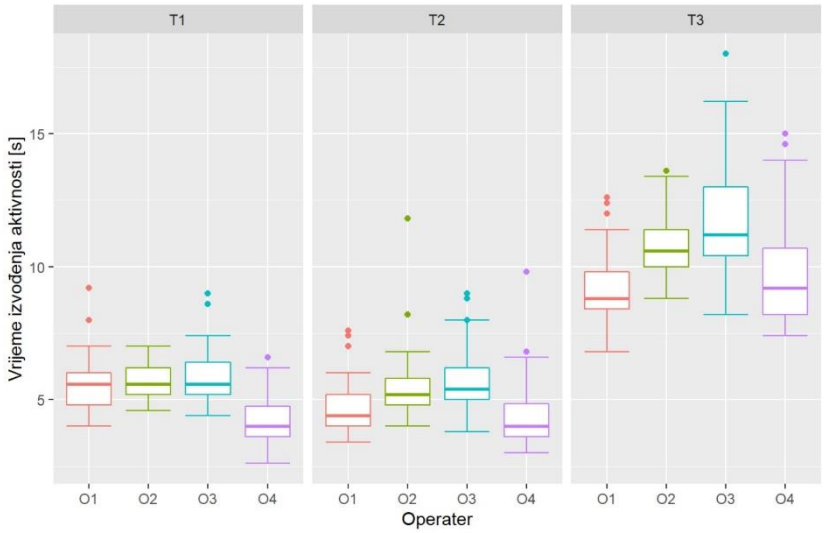
Rezultati testiranja razlika u vremenu izvođenja 4. aktivnosti između različitih operatera na istom tipu proizvoda dani su u tablici 4.11.

Tablica 4.11 Kruskal-Wallis test razlika u vremenu izvođenja 4. aktivnosti na istom proizvodu

Proizvod	Rezultat testa (α=0,05)
T1	H1
T2	H1
T3	H1

Zaključak 1: Uspoređivanjem svih operatera po pojedinom tipu proizvoda potvrđeno je da postoji značajna razlika u brzini izvođenja 4. aktivnosti kod svih tipova proizvoda. Kod proizvoda T1 operater O4 je brži od svih ostalih operatera prema Mann-Whitney testu. Kod T2

operater O4 je brži od svih operatera, a O1 od svih ostalih. Za T3 svi operateri su brži od O3. Ova situacija je dodatno prikazana na grafu sa slike 4.8.



Slika 4.8 Usporedba vremena izvođenja 4. aktivnosti po operateru i tipu proizvoda

Test razlika u vremenu izvođenja 4. aktivnosti od strane istog operatera na različitim tipovima proizvoda prikazan je u tablici 4.12.

Tablica 4.12 Kruskal-Wallis test razlika u vremenu izvođenja 4. aktivnosti od strane istog operatera na različitim tipovima proizvoda

Operater	Rezultat testa (α=0,05)
O1	H1
O2	H1
O3	H1
O4	H1

Zaključak 2: Kod 4. aktivnosti brzina svih operatera značajno je različita jer je proizvod T3 većih dimenzija te su stoga svi operateri sporiji na tome proizvodu. Razlog tome može biti, to što je poprečna stranica dulja te ima više pozicija za umetanje lamela.

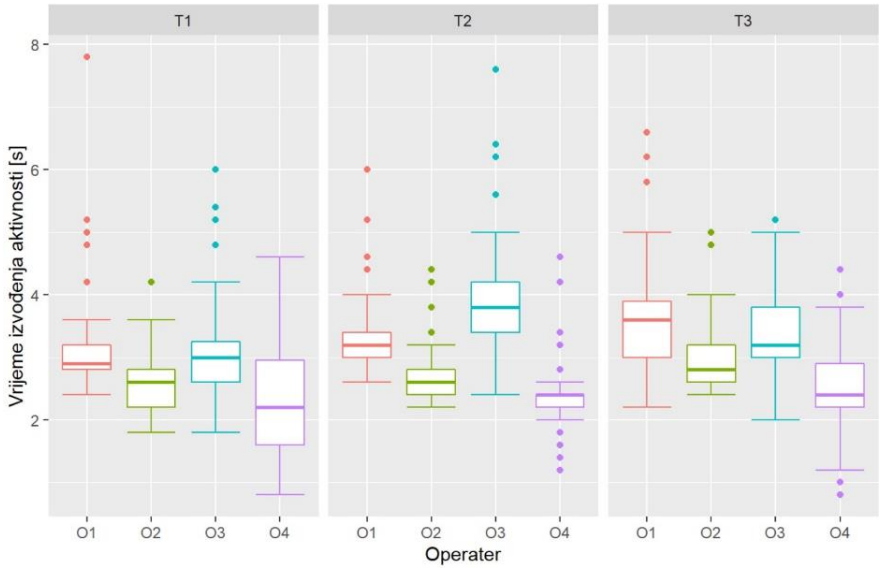
4.2.5 5. Aktivnost: Postavljanje uzdužne stranice

Rezultati testiranja razlika u vremenu izvođenja 5. aktivnosti između različitih operatera na istom tipu proizvoda dani su u tablici 4.13.

Tablica 4.13 Kruskal-Wallis test razlika u vremenu izvođenja 5. aktivnosti na istom proizvodu

Proizvod	Rezultat testa (α=0,05)
T1	H1
T2	H1
T3	H1

Zaključak 1: Uspoređivanjem svih operatera po pojedinom tipu proizvoda potvrđeno je da postoji značajna razlika u brzini izvođenja 5. aktivnosti kod svih tipova proizvoda. Za proizvod T1 operater O4 brži je od svih operatera, a O2 od ostalih, prema Mann-Whitney testu. Za T2 operater O4 brži je od svih, O2 od ostalih, a O1 od O3. Za T3 operater O4 brži je od svih operatera, a O2 od ostalih.



Slika 4.9 Usporedba vremena izvođenja 5. aktivnosti po operateru i tipu proizvoda

Test razlika u vremenu izvođenja 5. aktivnosti od strane istog operatera na različitim tipovima proizvoda prikazan je u tablici 4.14.

Tablica 4.14 Kruskal-Wallis test razlika u vremenu izvođenja 5. aktivnosti od strane istog operatera na različitim tipovima proizvoda

Operater	Rezultat testa (α=0,05)
O1	H1
O2	H1
O3	H1
O4	H0

Zaključak 2: Kod 5. aktivnosti brzina svih operatera, osim O4, značajno varira ovisno o tipu proizvoda.

Kako je kod operatera O4 na boxplot grafu sa slike 4.9 uočena veća količina vrijednosti izvan raspona ostatka podataka za proizvod T2, napravljena je dodatna analiza video zapisa.

Zaključak 3: U slučaju operatera O4 odstupanja su posljedica činjenice da je ovu aktivnost radio drugačije od svih ostalih operatera, na način da odmah pušta stranicu te je kasnije podešava kada počne s umetanjem kopče u 8. aktivnosti, stoga je ta aktivnost kod njega veoma kratka.

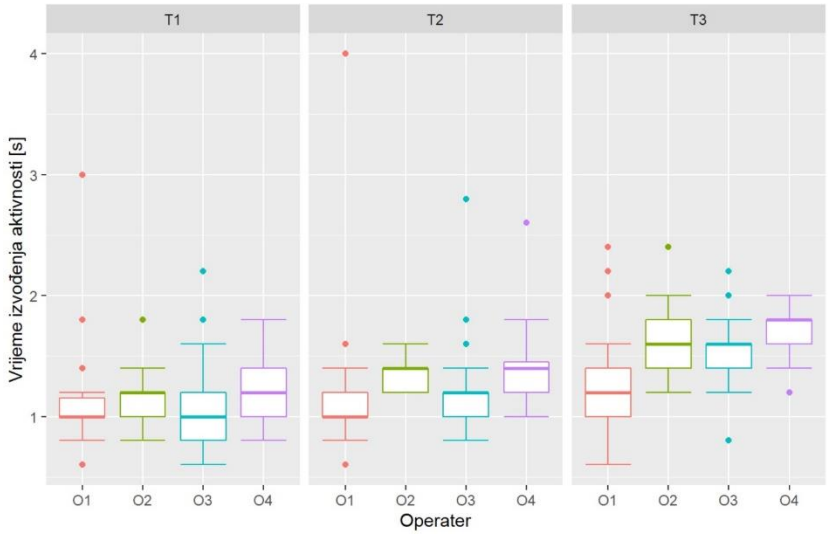
4.2.6 6. Aktivnost: Odlaganje gotovog proizvoda

Rezultati testiranja razlika u vremenu izvođenja 6. aktivnosti između različitih operatera na istom tipu proizvoda dani su u tablici 4.15.

Tablica 4.15 Kruskal-Wallis test razlika u vremenu izvođenja 6. aktivnosti na istom proizvodu

Proizvod	Rezultat testa (α=0,05)
T1	H1
T2	H1
T3	H1

Zaključak 1: Uspoređivanjem svih operatera po pojedinom tipu proizvoda potvrđeno je da postoji značajna razlika u brzini izvođenja 6. aktivnosti kod svih tipova proizvoda. Za proizvod T1 operater O1 i O3 brži su od operatera O2, prema Mann-Whitney testu. Za T2 operater O1 brži je od svih, a O3 od ostalih. Za T3 operater O1 brži je od svih operatera, a O3 od ostalih, a O2 od O4.



Slika 4.10 Usporedba vremena izvođenja 6. aktivnosti po operateru i tipu proizvoda

Test razlika u vremenu izvođenja 6. aktivnosti od strane istog operatera na različitim tipovima proizvoda prikazan je u tablici 4.16.

Tablica 4.16 Kruskal-Wallis test razlika u vremenu izvođenja 6. aktivnosti od strane istog operatera na različitim tipovima proizvoda

Operater	Rezultat testa (α=0,05)
O1	H1
O2	H1
O3	H1
O4	H1

Zaključak 2: Kod 6. aktivnosti brzina svih operatera značajno varira ovisno o tipu proizvoda. Pretpostavka je da je potrebno više vremena kod odlaganja proizvoda većih dimenzija.

4.2.7 7., 8. i 9. Aktivnost: Umetanje i učvršćivanje kopče (desna gornja, desna donja, lijeva donja)

Moguće je dati generalni zaključak za 7., 8. i 9. aktivnost.

Zaključak 1: Aktivnosti 7, 8 i 9 slično kao i 2. aktivnost odnose se na umetanje kopče, međutim na različitoj poziciji konstrukcije. Razlog zašto je ova aktivnost razbijena na četiri različite leži u tome da su one vizualno različite pa je pretpostavka da će modelima lakše biti naučiti svaku od njih kao posebnu aktivnost. Većina razlika u brzini izvođenja između operatera po pojedinom proizvodu proizlazi iz činjenice da je operater O3 sporiji od ostalih. Ovo je generalni

trend kod ova četiri tipa aktivnosti, koje bi se mogle kategorizirati kao fine aktivnosti u kojima se O3 slabije snalazi.

S druge strane, zanimljivo je primijetiti da se vrijeme izvođenja 7. i 9. aktivnosti od strane istog operatera na različitim proizvodima razlikuje, usprkos pretpostavci da na ove aktivnosti ne utječu dimenzije proizvoda. U slučaju 8. aktivnosti potvrđena je prethodna pretpostavka za sve operatere, što je pokazano u tablici 4.17.

Tablica 4.17 Kruskal-Wallis test razlika u vremenu izvođenja 7., 8. i 9. aktivnosti od strane istog operatera na različitim tipovima proizvoda

Operater	Rezultat testa (α=0,05)		
	Aktivnost **7**	Aktivnost **8**	Aktivnost **9**
O1	H1	H0	H1
O2	H1	H0	H1
O3	H1	H0	H1
O4	H1	H0	H0

U sljedećem poglavlju bit će objašnjeni proces izrade i evaluacije model na temelju prikupljenog uzorka.

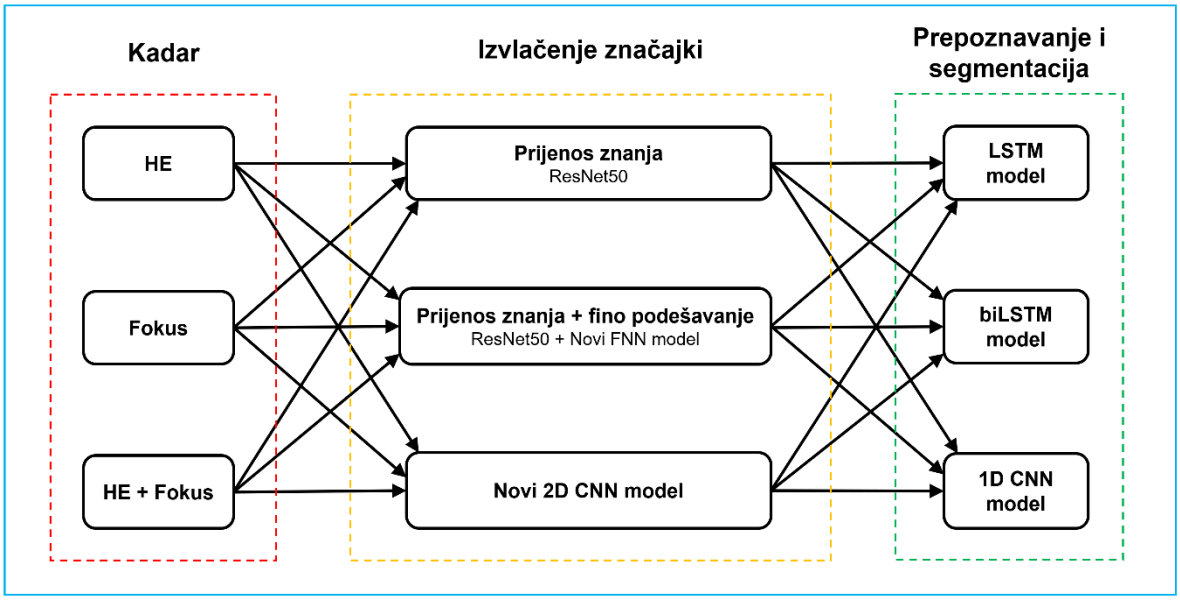
5. MODEL ZA ISTOVREMENO PREPOZNAVANJE I VREMENSKU SEGMENTACIJU AKTIVNOSTI

Postojeći pristupi istovremenog prepoznavanja i vremenske segmentacije aktivnosti u proizvodnim procesima opisani u odjeljku 2.4.4, oslanjaju se na manualno definirane značajke i algoritme strojnog učenja s plitkom strukturom. Argumentirana rasprava o nedostatcima takvih pristupa dana je u dijelovima rada 1.2 i 2.3, a sažetak je taj da se u slučaju složenih problema, u kakve spadaju oni iz domene računalnog vida, rijetko može znati koje će to značajke olakšati posao klasifikacijskog algoritma i osigurati visoku učinkovitost pristupa. Ovo je bio poticaj da se istraže pristupi temeljeni na dubokom strojnom učenju, stoga je i napravljen pregled literature vezan za primjenu takvih metoda u području prepoznavanja i vremenske segmentacije aktivnosti koji su predstavljeni u odjeljcima 2.4.2 i 2.4.3. Na temelju pregleda prepoznati su učestali modeli i pristupi iz domene problema te njihove prednosti i nedostatci. U poglavlju 3 obrazložen je pristup prikupljanja podataka potrebnih za razvoj novih modela dubokog strojnog učenja. Odlika kreiranog uzorka je ta da je sačinjen od podataka prikupljenih u realnim proizvodnim uvjetima, na različitim tipovima proizvoda i od strane različitih izvoditelja čime su stvorene dobre pretpostavke za ispitivanje objektivne primjenjivosti modela dubokog strojnog učenja. Analiza uzorka iz poglavlja 4 dovela je do bitne odluke koja je proizašla na temelju strukture pojedinačnih opažanja. Iz razloga što većina opažanja sadrži mali udio aktivnosti, odlučeno je da će zadatak istovremenog prepoznavanja i vremenske segmentacije aktivnosti biti postavljen kao zadatak klasifikacije svake sličice video zapisa, zbog čega je ovaj rad moguće smjestiti u „*action segmentation*“ grupu istraživanja. Pristupi iz ove grupe koriste dva koraka za obradu ulaznih podataka, zbog čega je ovaj način bio korišten za razvoj modela dubokog strojnog učenja u okviru disertacije.

5.1 Pristup izradi modela

Procedura koja je korištena u razvoju modela iz ovog rada prikazana je slikom 5.1, kojom se želi ukazati na činjenicu da su provedeni eksperimenti s tri promjenjiva faktora. Ti faktori su: kadar snimanja aktivnosti, pristupi izvlačenja značajki i arhitekture modela za prepoznavanje i vremensku segmentaciju. U poglavlju 3 navedeno je da će biti korišteni podatci iz dva kadra snimanja, podatci prikupljeni kamerom koja je bila postavljena iznad glave izvoditelja aktivnosti (HE kadar) i podatci iz kadra s fokusom na ruke izvoditelja (Fokus kadar). Treće stanje faktora kadra snimanja odnosi se na fuziju podataka iz oba kadra, kojemu je pridodana oznaka *Concat.* Dva su razloga za korištenje ovih stanja faktora kadra snimanja. Prvi razlog je

ispitivanje utjecaja izvora podataka na učinkovitost modela. Drugi razlog motiviran je time da je prikupljenim video zapisima smanjena rezolucija s ciljem brže i jednostavnije daljnje obrade zbog čega dolazi do gubitka dijela informacija što se nastoji kompenzirati kroz korištenje različitih kadrova.



Slika 5.1 Procedura izrade modela

Faktor pristupa izvlačenja značajki i faktor arhitekture završnog modela izravna su posljedica toga što je korišten ranije spomenuti pristup obrade video zapisa u dva koraka koji je uobičajen za područje „*action segmentation*“-a. Iako su u odjeljku 2.3 navedene prednosti integriranog učenja značajki i klasifikacije kod modela dubokog učenja, zbog složenosti podataka u obliku video zapisa u praksi se pribjegava obradi kroz odvojene korake izvlačenja značajki i finalne klasifikacije i segmentacije. Ovu odluku moguće je opravdati dvama argumentima. Prvi argument je taj da nisu postojale tehničke pretpostavke za integrirano učenje modela u okviru istraživanja, dok se drugi odnosi na saznanja iz istraživanja u kojima se koristilo integrirano učenje modela [122]. Rezultati integriranog učenja u spomenutom istraživanju bili su neznatno bolji nego kada se koristio pristup u dva koraka, ali je vrijeme učenja modela bilo znatno dulje.

Tri pristupa izvlačenja značajki koja su korištena u eksperimentima su:

- Prijenos znanja iz prednaučenog modela bez finog podešavanja – FE pristup
- Prijenos znanja iz prednaučenog modela s finim podešavanjem primjenom pojedinačnih sličica video zapisa iz vlastitog uzorka – TL pristup
- Učenje modela na pojedinačnim sličicama video zapisa iz vlastitog uzorka i naknadno izvlačenje značajki – TB pristup

Koncept prijenosa znanja odnosi se na tehnike korištenja slojeva modela naučenog na nekim podatcima $\mathcal{P}_{old}$ s ciljem izvlačenja korisnih značajki iz novih podataka $\mathcal{P}_{new}$, pri čemu distribucije spomenutih skupova podataka nisu nužno iste. Drugi vid korištenja slojeva naučenog modela je da služe kao temelj novih modela na način da se na stare slojeve dodaju novi slojevi i provede proces učenja. Ovaj koncept bit će detaljnije objašnjen u sljedećem odjeljku. Praksa korištenja prijenosa znanja za izvlačenje značajki veoma je učestala u domeni računalnog vida pa tako i kod problema „*action segmentation*"-a, gdje svi radovi iz pregleda literature u 2.4.2 koriste neki oblik prijenosa znanja. Međutim, u analiziranim radovima nije pronađeno ni jedno istraživanje koje koristi model za obradu slika koji je u potpunosti naučen na novim podatcima te je iskorišten za izvlačenje značajki iz pojedinačnih sličica video zapisa. Kako bi se istražila svojstva navedenih pristupa i njihov utjecaj na finalni model u eksperimentima su korištena sva tri načina izvlačenja značajki. Svaki od ova tri pristupa ima svoje prednosti i mane. Inicijalna pretpostavka je bila da prijenos znanja bez finog podešavanja (FE pristup) omogućava brzo izvlačenje značajki slabije kvalitete, u smislu da će finalni modeli naučeni na njima biti manje učinkovitosti u odnosu na modele koji će koristiti značajke dobivene nekim od preostala dva pristupa. S druge strane očekivano je da će model za izvlačenje značajki razvijen na vlastitim podatcima (TB pristup) dati najbolje značajke, ali će proces učenja i izvlačenja značajki biti najduži u odnosu na preostala dva pristupa. ResNet50 [43] prednaučen na ImageNet [80] podatcima odabran je kao model koji će se koristiti za prva dva pristupa izvlačenja značajki, dok će novo razvijeni model za izvlačenje značajki u arhitekturi koristiti tehnike rezidualnog učenja i preskočnih veza kako bi bio usporediv s prva dva pristupa. ResNet50 model i tehnika rezidualnog učenja bit će objašnjeni u sljedećem odjeljku. Razlozi odluke da se koristi 2D CNN model za izvlačenje značajki, u konkretnom slučaju ResNet50, povezani su uz njegova svojstva kod učenja i njegove funkcionalne karakteristike koje se smatraju boljima u odnosu na 3D CNN modele ili kombinaciju 2D CNN modela i algoritama poput optičkog toka. Konkretnije, preskočne veze, koje su korištene kod ResNet50 modela, omogućuju lakše učenje dubokih modela uslijed olakšanog protoka gradijenta. Prednosti dubokih modela u odnosu na modele s jednakim brojem parametara, ali manjom dubinom, teorijski i empirijski su dokazane i svode se na to da dubina omogućuje učinkovitiji način izvlačenja statističkih pravilnosti iz podataka. Ovu tvrdnju je moguće potkrijepiti konkretnom usporedbom ResNet50 modela s 3D CNN modelom poput C3D [155], gdje ResNet50 ima 50 slojeva i 25 milijuna parametara, a C3D 11 slojeva i 80 milijuna parametara. Dio istraživača smatra da nije dovoljno koristiti samo 2D CNN modele za izvlačenje značajki jer se time gubi dio informacija vezanih za vremenski kontekst. Jedan od

ciljeva disertacije bio je razviti pristup prepoznavanja i vremenske segmentacije aktivnosti koji ne koristi komponente koje nisu sposobne učiti na temelju podataka, čime su eliminirani optički tok i njemu slični algoritmi, a ocjena učinkovitosti fiksnih algoritama bit će temeljena na rezultatima finalnih modela koji će koristiti značajke izvučene FE pristupom. Konačno, pretpostavka istraživanja je da su 2D CNN modeli adekvatni za izvlačenje značajki iz video zapisa, dok odgovornost za vremenski kontekst treba biti u potpunosti na modelima za finalnu segmentaciju i klasifikaciju, stoga 3D CNN modeli, koji omogućuju hvatanje ograničenog vremenskog konteksta, nisu korišteni za izvlačenje značajki.

Tri arhitekture modela za istovremeno prepoznavanje aktivnosti i vremensku segmentaciju korištene u eksperimentima temeljene su na LSTM slojevima, dvosmjernom LSTM-u[18] i 1D konvoluciji. Barem jedna od ove tri vrste arhitekture korištena je kod svih analiziranih radova iz literature o „*action segmentation*"-u za finalnu segmentaciju i klasifikaciju, uz iznimku istraživanja [123] koje se oslanja na kombinaciju unaprijedne mreže i algoritma uvjetnih slučajnih polja (CRF). Dio istraživanja koristi i razne fiksne algoritme, kreirane na temelju poznavanja domene, za dodatno izglađivanje predikcija modela kako bi se povećala točnost. U disertaciji nisu korištene ove vrste algoritama zbog ranije navedenog cilja da se iz razvijenog pristupa isključe svi elementi koji nemaju sposobnost učenja. Povratne neuronske mreže s LSTM slojevima opisane su u 2.3.4, dok biLSTM predstavlja njihovo proširenje u vidu korištenja zasebnih slojeva za obradu niza iz dva različita smjera. Pretpostavka, temeljena na rezultatima iz literature, bila je da će modeli temeljeni na LSTM i biLSTM slojevima imati veću točnost, ali će biti teži i sporiji za učenje od 1D konvolucijskih modela, s obzirom da se radi o tipu arhitekture koji je specifično razvijen za rad s nizovima. Shodno tome, očekivalo se da će 1D konvolucijski modeli biti brži u učenju, ali i potencijalno neprecizniji. Zanimljivo je da u analiziranoj literaturi nije pronađena kombinacija 2D CNN modela za izvlačenje značajki s 1D CNN modelom za segmentaciju i klasifikaciju, a istražena je u disertaciji. Mogući razlog zašto ova kombinacija nije prisutna u literaturi je ranije spomenuta argumentacija dijela znanstvene zajednice da izgubljeni vremenski kontekst primjenom 2D CNN modela za pripremu značajki nije moguće kasnije povratiti. Primjena dilatirane 1D konvolucije uočena je u većem broju radova [23,24,70,125], a dio pristupa predlaže i korištenje dvije vremenske grane [23,125] kako bi se istovremeno uhvatio kratkoročni i dugoročni vremenski kontekst. Ove ideje su korištene u razvoju 1D CNN modela iz ovog istraživanja. Ova odluka je temeljena na analizi prikupljenog uzorka iz proizvodnog procesa. Analiza je ukazala na to da je radne aktivnosti jedino moguće

[18] U nastavku teksta će se koristiti oznaka biLSTM, inspirirana engleskim nazivom „*bidirectional LSTM*"

raspoznati na temelju toga koje su im aktivnosti prethodile, za što je potrebno poznavanje dugoročnog vremenskog konteksta, i trenutne informacije, za što je potreban kratkoročni vremenski kontekst.

Kako je svaki od tri opisana faktora imao tri moguća stanja, provedeni su eksperimenti za 27 grupa modela, s ciljem pronalaska najboljeg modela iz svake grupe. Da bi proces eksperimentiranja bio što jednostavniji i brži, razvijena je programska biblioteka naziva „*phd_lib*“[19] za programski jezik Python, verzija 3.8, koja se oslanja na biblioteku za numeričku matematiku i strojno učenje Tensorflow [156], verzija 2.3. Biblioteka je usmjerena na duboko strojno učenje iz podataka u obliku video zapisa te predstavlja kompilaciju dobrih praksi za efikasnu pripremu i slanje podataka modelu, upravljanje procesom učenja i evaluaciju modela iz domene istovremenog prepoznavanja i vremenske segmentacije aktivnosti. Ključni moduli biblioteke opisani su u tablici 5.1.

Tablica 5.1 Moduli phd_lib biblioteke

Naziv modula	**Opis**
callbacks	Nadzor i upravljanje procesom učenja. Podmodul *lr_helpers* sadrži dodatne module za traženje optimalne stope učenja i upravljanje rasporedom stope učenja. Podmodul *training_monitor* služi za praćenje procesa učenja te spremanje metrike učinkovitosti i funkcije gubitka. Podmodul *epoch_checkpoint* služi za pohranu inačica modela za vrijeme učenja.
data_pipeline	Priprema ulaznih podataka i generiranje optimalnog ulaznog toka podataka, s svrhom smanjenja latencije između pripreme podataka i učenja modela. Podmodul *tf_record_helpers* sadrži niz funkcija za zapisivanje i parsiranje podataka u preporučenom formatu za primjenu s Tensorflow bibliotekom. Podmodul *pipe_builders* služi za generiranje optimiziranog ulaza podataka u model.
models	Sadrži podmodule za kreiranje modela za vremensku segmentaciju i modela za izvlačenje značajki koji su korišteni u eksperimentima u okviru disertacije.
config	Organizacija putanja za pohranu različitih izlaznih podataka.
metrics	Segmentacijske i detekcijske metrike te funkcije za pripremu oznaka kako bi bile pogodne za izračun metrike.
video_editing_utils	Manipulacija video zapisa, npr. rezanje, poduzorkovanje, pretvorba u sličice i slično. Zahtjeva besplatni software *ffmpeg*[20] u pozadini.

[19] https://github.com/Miha87/phd_mg

[20] https://ffmpeg.org

5.2 Modeli za izvlačenje značajki

Svrha korištenja modela za izvlačenje značajki je smanjenje dimenzionalnosti ulaznih podataka i kreiranje deskriptivnih značajki koje će olakšati zadatak istovremenog prepoznavanja aktivnosti i vremenske segmentacije. Korištena su tri različita pristupa za izvlačenje značajki. Za ovaj korak korišteni su 2D CNN modeli za obradu slika temeljeni na ResNet50 modelu, nakon što su video zapisi rastavljeni na pojedinačne sličice. S obzirom na zahtjeve ovog modela, ulazne slike bile su podvrgnute dodatnoj obradi kako bi svaka i-ta slika bila opisana tenzorom $\mathbf{X}^{(i)} \in \mathbb{R}^{224\times224\times3}$. Dodatne operacije pripreme ulaznih slika bit će objašnjene za svaki pristup zasebno. Rezultat izvlačenja značajki iz slike $\mathbf{X}^{(i)}$ je vektor značajki $\boldsymbol{x}^{(i)} \in \mathbb{R}^{2048}$, razlog za ovu dimenziju vektora značajki bit će objašnjen u daljnjem tekstu.

Prva dva pristupa izvlačenja značajki oslanjaju se na koncept prijenosa znanja (engl. *transfer learning*). Prijenos znanja koristi glavno svojstvo dubokog strojnog učenja koje se odnosi na sposobnost učenja značajki iz sirovih podataka kroz slojeve modela, pri čemu raniji slojevi uče jednostavnije koncepte, a dublji slojevi složenije koncepte koji su kombinacija koncepata naučenih u ranijim slojevima. Glavna ideja prijenosa znanja je da se reprezentacije ulaznih podataka naučene na jednom skupu mogu iskoristiti za rješavanje problema na drugom skupu podataka. Spomenuti skupovi podataka ne moraju nužno proizlaziti iz iste distribucije, ali je pretpostavka da je velik broj faktora koji objašnjava varijacije u jednom skupu koristan za objašnjavanje varijacija iz drugog skupa podataka. Na primjeru obrade slika korištenjem konvolucijskih mreža najlakše je shvatiti zašto je prijenos znanja učinkovito rješenje. CNN modeli u prvim slojevima koriste jezgre malog receptivnog polja zbog čega uče kako raspoznati jednostavne vizualne objekte poput linija, bridova i tekstura te njihove kombinacije. Intuicija je da spomenuti primitivni vizualni objekti čine korisne značajke za opisivanje različitih složenih vizualnih objekata, odnosno da imaju sposobnost generalizacije na probleme raspoznavanja klasa objekata prisutnih u različitim skupovima podataka [157,158]. Na primjer, horizontalni bridovi naučeni na skupu podataka o brodovima mogu biti iskorišteni za prepoznavanje tipova automobila. S druge strane, reprezentacije naučene u kasnijim slojevima koji su bliži završnom (klasifikacijskom ili regresijskom) sloju, specifične su za konkretan skup podataka na kojem je model naučen, zbog čega je manja vjerojatnost da će biti korisne na različitim skupovima podataka. Prijenos znanja nalazi čestu primjenu kada je potrebno naučiti novi model na malom skupu podataka. Iz razloga što modeli dubokog strojnog učenja zahtijevaju veliku količinu podataka, što posebno vrijedi za domenu računalnog vida, u pravilu je uvijek korisno primijeniti prijenos znanja. ImageNet [80] je skup podataka koji sadrži više

od 14 milijuna slika s preko 20.000 različitih klasa objekata. Upravo zbog ove raznolikosti, smatra se da je ImageNet dovoljno reprezentativan za opisivanje većine ključnih vizualnih koncepata iz domene slika, stoga postoji znatan broj modela za obradu slika koji su prednaučeni na ovom skupu podataka. Ovo je bio jedan od faktora za izbor ResNet50 modela prednaučenog na ImageNet podatcima za izvlačenje značajki. U ovom radu korištene su dvije osnovne vrste prijenosa znanja iz prednaučenog modela, bez finog podešavanja i s finim podešavanjem na novim podatcima. Način provedbe pojedinog tipa prijenosa znanja te njihove razlike tema su sljedeća dva odjeljka, dok je završni odjeljak rezerviran za opis izvlačenja značajki iz modela u potpunosti naučenog na vlastitim podatcima.

5.2.1 Prijenos znanja iz prednaučenog modela bez finog podešavanja – FE pristup

Procedura prijenosa znanja bez finog podešavanja slijedi dva koraka:

a) Iz izabranog modela uklanja se sloj odgovoran za finalnu klasifikaciju, jer je on specifičan za problem na kojem je model naučen. U slučaju da se zadatak i skup podataka na kojem je model za prijenos znanja naučen znatno razlikuju od zadatka i skupa podataka kojeg je potrebno riješiti, uklanjaju se i završni slojevi koji su blizu sloja za klasifikaciju. Razlog uklanjanja spomenutih slojeva je da su reprezentacije naučene u njima također specifične za inicijalni skup podataka.
b) Nakon uklanjanja odgovarajućeg broja slojeva, ostatak modela se tretira kao fiksni algoritam koji obavlja transformaciju sirovih ulaznih podataka i vraća izračunati vektor značajki.

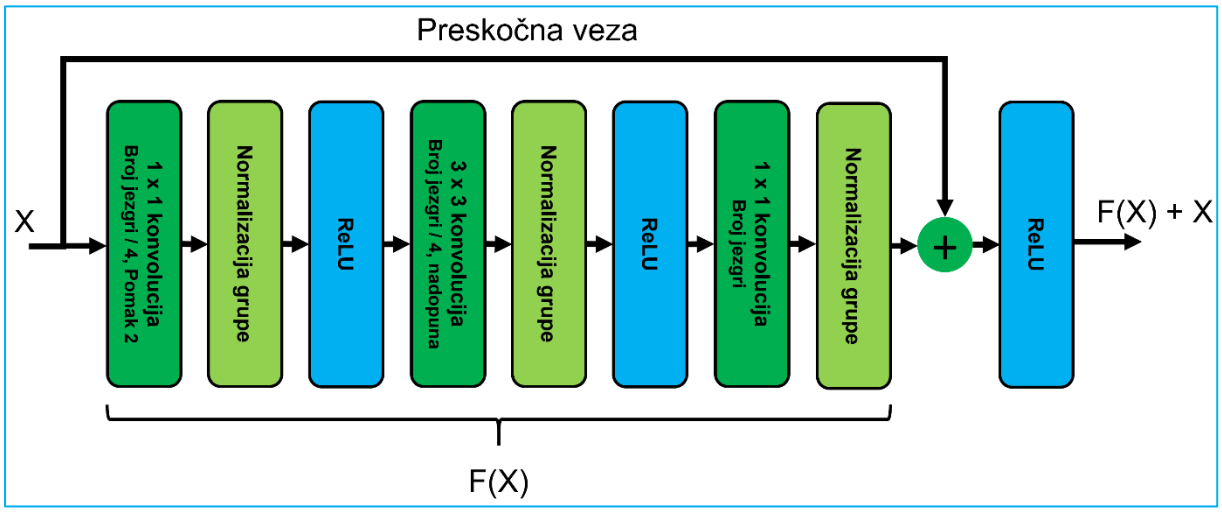
Ovaj pristup je odabran kako bi bilo moguće dati ocjenu učinkovitosti fiksnih algoritama na zadatku istovremenog prepoznavanja i vremenske segmentacije aktivnosti, kroz usporedbu s pristupima gdje su sve komponente podložne učenju. Eksperiment je proveden primjenom izvornog ResNet50 modela koji je opisan u radu [43] koji je naučen na ImageNet podatcima.

Glavna odlika ResNet50 modela je korištenje koncepta *rezidualnog učenja,* koji znatno olakšava učenje jako dubokih modela[21]. U suštini, ideja je olakšati učenje modela na način da se izvorna funkcija $H(\boldsymbol{x})$ koju model nastoji aproksimirati, zamjeni s funkcijom $F(\boldsymbol{x}) = H(\boldsymbol{x}) - I(\boldsymbol{x})$, gdje je $I(\boldsymbol{x}) = \boldsymbol{x}$ funkcija identiteta. Drugim riječima, pretpostavka je da je izvornu funkciju moguće rastaviti na zbroj rezidualne funkcije $F(\boldsymbol{x})$ i ulaza $\boldsymbol{x}$, što bi u najgorem

[21] Ovisno od skupa podataka, broj slojeva može biti i veći od 1000, ali u izvornim implementacijama kretao se od 50 do 100 slojeva.

slučaju trebalo omogućiti optimizacijskom postupku da barem nauči funkciju identiteta. Empirijski je dokazano da duboki modeli imaju većih problema od plitkih modela u učenju funkcije identiteta [43], što je kontra intuitivno s obzirom da modeli veće složenosti imaju veću reprezentativnu moć, stoga se smatra da su problemi obično povezani s optimizacijskom metodom, a rezidualno učenje ublažava ovaj problem. U praktičnom smislu, rezidualno učenje je implementirano na način da su u arhitekturu modela dodane *preskočne veze* (vidi sliku 5.2), koje predstavljaju funkciju identiteta s obzirom da ne sadrže nikakve transformacijske slojeve, odnosno parametre. Preskočne veze ujedno olakšavaju prosljeđivanje gradijenta i razlog su zašto je moguće učiti modele s velikim brojem slojeva te je po toj karakteristici svrha preskočnih veza slična ideji memorijskih ćelija u LSTM slojevima. Osim toga dodatna korist preskočne veze je ta da omogućuje primjenu visoke stope učenja ($1 \cdot 10^{-1}$), što bi kod većine drugih modela rezultiralo divergencijom.

U ResNet50 modelu rezidualno učenje je implementirano kroz rezidualne blokove kako je prikazano na slici 5.2.



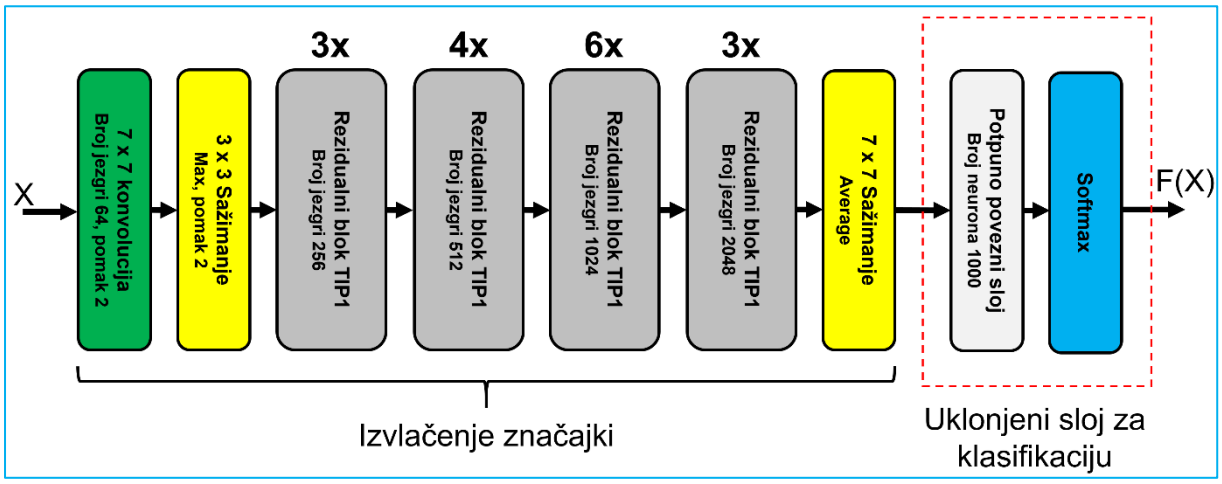
Slika 5.2 Rezidualni blok TIP1 prema [43]

Transformacijska (rezidualna) grana rezidualnog bloka sadrži tri konvolucijska sloja pri čemu iza prva dva slijede slojevi za normalizaciju grupe i ReLU aktivacijska funkcija, a izlaz iz posljednjeg konvolucijskog sloja nakon normalizacije grupe zbraja se s ulazom rezidualnog bloka i prosljeđuje u ReLU aktivacijsku funkciju. Kako bi zbrajanje preskočne i rezidualne grane bilo moguće, dimenzije ulaznog tenzora $\mathbf{X}$ i tenzora koji je produkt rezidualne grane $\mathbf{F}(\mathbf{X})$ moraju biti iste, a u slučaju da nisu, u preskočnoj grani se radi linearna transformacija s 1×1

konvolucijskim slojem[22] s odgovarajućim brojem jezgri. Prvi i treći konvolucijski sloj rezidualnog bloka koriste 1 × 1 konvoluciju, dok drugi koristi 3 × 3 konvoluciju. Prvi i drugi sloj imaju 4 puta manji broj jezgri od posljednjeg konvolucijskog sloja, uz to da je u prvom konvolucijskom sloju pomak postavljen na 2, a u drugom se koristi nadopuna nulama kako bi izlaz iz njega imao iste prostorne dimenzije kao izlaz iz prvog konvolucijskog sloja. Svrha ovako izabranih vrijednosti hiperparametara konvolucijskih slojeva je sljedeća:

- Zadatak prvog konvolucijskog sloja je da napravi redukciju prostornih dimenzija ulaznog tenzora primjenom pomaka S=2, što ima za posljedicu brže vrijeme obrade modela.
- Prva dva konvolucijska sloja smanjuju dimenziju dubine ulaznog tenzora jer koriste 4 puta manji broj jezgri u odnosu na posljednji sloj, čime je dodatno smanjen broj parametara što također pomaže ubrzanju obrade.
- Zadatak zadnjeg sloja je da poveća reprezentativni kapacitet modela, bez da se promijene veličine prostornih dimenzija, stoga ima 4 puta veći broj jezgri i koristi 1 × 1 konvoluciju.

Arhitektura cijelog ResNet50 modela prikazana je na slici 5.3 na kojoj je zanimljivo primijetiti da se u cjelokupnom modelu sloj sažimanja koristi samo dva puta. Jedan od zadataka sloja sažimanja je smanjenje prostornih dimenzija tenzora, a u ResNet50 modelu ovaj zadatak je prebačen na dio konvolucijskih slojeva koji koriste pomak veličine 2.



Slika 5.3 ResNet50 [43] mreža prednaučena na ImageNet podatcima za izvlačenje značajki

[22] Termin operacije s „konvolucijskim slojem dimenzije 1 x 1“, odnosi se na to da je provedena operacija konvolucije s jezgrom čije su prostorne dimenzije 1 x 1. Ova terminologija se često koristi u nastavku rada.

Ulazni tenzor $\mathbf{X}^{(i)} \in \mathbb{R}^{224\times224\times3}$ se u modelu prvo propušta kroz 7×7 konvoluciju s pomakom 2 i brojem jezgri 64, kako bi se smanjile prostorne dimenzije i povećala dubina, nakon čega slijedi dodatno smanjenje prostornih dimenzija 3×3 sažimanjem maksimalnom vrijednosti uz pomak veličine 2. U nastavku modela izmjenjuju se četiri varijante rezidualnih blokova sa slike 5.2, koji se razlikuju po broju jezgri koje će imati izlazni tenzor iz bloka, na način da se s dubinom modela udvostručuje broj jezgri. Ovo je standardna praksa, u smislu da smanjenje prostornih dimenzija tenzora prati povećanje dubine, koja se oslanja na pretpostavku da su značajke kasnijih slojeva bitnije za finalnu klasifikaciju, stoga se povećava njihov broj koji je kontroliran brojem jezgri. Izlazni tenzor iz zadnjeg rezidualnog bloka je dimenzije $7 \times 7 \times 2048$, na kojeg se primjenjuje 7×7 sažimanje prosječnom vrijednosti, što daje $1 \times 1 \times 2048$ dimenzionalni tenzor. Zbog svojstva izomorfizma[23] između vektorskih prostora $\mathbb{R}^{1\times1\times2048}$ i $\mathbb{R}^{2048}$, izlazni tenzor je moguće predstaviti vektorom $\boldsymbol{x}^{(i)} \in \mathbb{R}^{2048}$ te je kao takav proslijeđen u potpuno povezani sloj koji rezultira vektorom dimenzije 1000, što odgovara broju različitih klasa na kojima je ResNet50 model naučen. Cjelokupni model imao je 25.583.592 parametara.

U okviru istraživanja uklonjen je završni potpuno povezani sloj (vidi sliku 5.3) te je iskorišten ostatak modela za izvlačenje značajki iz slika unutar prikupljenog uzorka. Nakon uklanjanja ovog sloja model je imao 23.534.592 parametara. Ulazne slike iz oba kadra snimanja su dodatno formatirane kako bi imale prostorne dimenzije 224×224 te su vrijednosti piksela u pojedinim kanalima centrirane oduzimanjem aritmetičkih sredina izračunatih na ImageNet skupu podataka, a to su $(123{,}68;\ 116{,}779;\ 103{,}939)$ kako bi odgovarale zahtjevima ResNet50 mreže. Po obradi svake slike iz oba kadra dobiveni su vektori značajki $\boldsymbol{x}^{(i)} \in \mathbb{R}^{2048}$. Vektor značajki trećeg „kadra“ Concat, koji je fuzija kadrova HE i Fokus, dobiven je naslaganjem izračunatih vektora značajki oba kadra za istu sliku u zajednički vektor dimenzije 4096. Ovako izračunati vektori korišteni su u razvoju modela za istovremeno prepoznavanje i vremensku segmentaciju. Vrijeme potrebno za izračun značajki svih slika iz uzorka bilo je oko 34 minute po kadru snimanja.

[23] Vektorski prostori A i B su izomorfni, ako i samo ako, je dimenzija prostora A jednaka dimenziji prostora B.

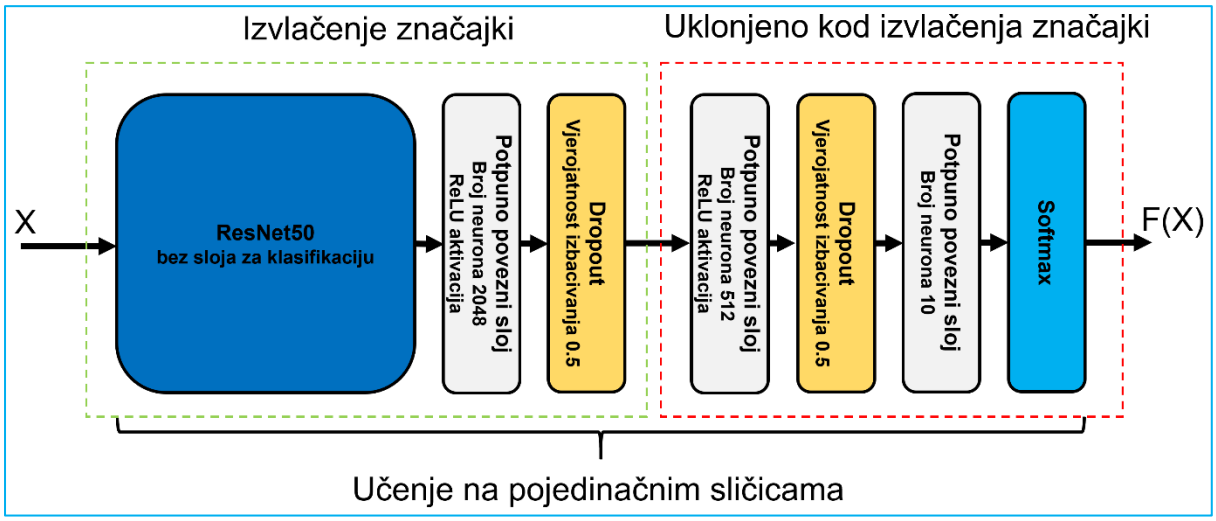
5.2.2 Prijenos znanja iz prednaučenog modela s finim podešavanjem primjenom sličica iz vlastitog uzorka – TL pristup

Procedura prijenosa znanja s finim podešavanjem slijedi ove korake:

a) Iz izabranog modela uklanja se sloj odgovoran za finalnu klasifikaciju, jer je on specifičan za problem na kojem je model naučen, a ako su zadatak i skup podataka na kojem je model za prijenos znanja naučen znatno drugačiji od zadatka i skupa podataka kojeg je potrebno riješiti, uklanjaju se i završni slojevi koji su blizu sloja za klasifikaciju.
b) Definiraju se novi slojevi sa stohastički inicijaliziranim vrijednostima parametara te se povezuju s modelom dobivenim iz koraka a). Broj novih slojeva i njihovi hiperparametri određuju se eksperimentalno kako bi odgovarali novom zadatku i skupu podataka.
c) Proces učenja modela iz koraka b) izvodi se na takav način da se prvo uče samo novo dodani slojevi s niskom stopom učenja (npr. $1 \cdot 10^{-4}$ ili $1 \cdot 10^{-5}$), na način da se gradijent ne prosljeđuje baznim prednaučenim slojevima. Razlozi za ovakvu praksu su ti da je potrebno određeno vrijeme da novi slojevi počnu učiti, stoga bi na početku učenja prosljeđivanje velikog gradijenta *starim* slojevima dovelo do poništavanja već naučenih korisnih reprezentacija.
d) Ovisno o veličini novog skupa podataka i sličnosti između starog i novog zadatka, provodi se učenje novih slojeva zajedno s određenim brojem završnih slojeva baznog modela. Ako je veličina novog skupa podataka mala, učenje završnih slojeva baznog modela brzo dovodi do prenaučenosti. U slučaju da su stari i novi zadatak veoma različiti, učenje starih slojeva na novim podatcima može povećati učinkovitost cijelog modela. Empirijski je pokazano da nije korisno podešavati ranije slojeve baznog modela jer oni sadrže reprezentacije primitivnih vizualnih objekata koje su slične kod većine podataka iz domene slika.
e) Uklanjanje odabranog broja slojeva iz novog modela te izvlačenje značajki primjenom preostalog fino podešenog modela.

Pristup izvlačenja značajki s finim podešavanjem je u eksperimentima proveden na način da je kao bazni model korišten ResNet50 kod kojeg je uklonjen klasifikacijski sloj. Nakon toga dodana su tri nova potpuno povezana sloja s 2048, 512 i 10 neurona, pri čemu je na izlazne vektore prvog i drugog sloja primijenjena regularizacija dropout tehnikom s vjerojatnosti izbacivanja neurona od 50% (vidi sliku 5.4). Odluka o izgledu arhitekture novo dodanog potpuno povezanog modela na bazni ResNet50 model vođena je ciljem da prvi potpuno povezani sloj kao rezultat daje vektore značajki jednake dimenzije kao i model iz prethodnog

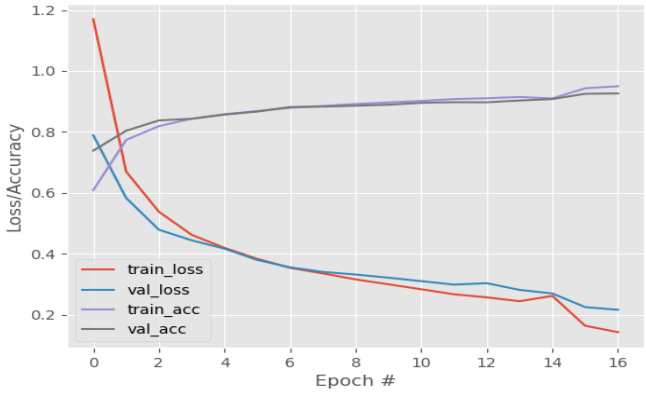
odjeljka, pošto će on biti korišten kao posljednji sloj u procesu izvlačenja značajki, te će na taj način biti moguća njihova usporedba. Srednji potpuno povezani sloj ubačen je s namjerom da se dimenzija podataka kaskadno smanjuje, jer određene empirijske spoznaje ukazuju na to da naglo smanjenje dimenzije može dovesti do gubitka bitnih informacija. Razmatrane su dvije opcije za izlaznu dimenziju srednjeg sloja, 1024 i 512. Eksperimenti su ukazali da se sa slojem od 1024 neurona brzo dolazi do prenaučenosti bez obzira na primjenu dropout-a, što je i donekle razumljivo jer model tada ima otprilike milijun parametara više. Dropout je dodan kako bi se ublažio problem prenaučenosti, što je bitno jer je kasnije u proces učenja bio uključen i zadnji rezidualni blok iz ResNet50 mreže. Izlazna dimenzija posljednjeg potpuno povezanog sloja uvjetovana je brojem različitih klasa aktivnosti u prikupljenom uzorku.



Slika 5.4 Kombinacija ResNet50 mreže i novog unaprijednog modela za izvlačenje značajki

Naučena su dva zasebna modela s arhitekturom koja je prikazana na slici 5.4, za slike iz kadra HE i kadra Fokus. Broj parametara kod ovih modela je bio 28.785.162, od čega 5.250.570 otpada na novo dodane slojeve. Modeli su naučeni na temelju podataka iz skupa za učenje $\mathcal{P}_{train}$. Ulazni podatci su pripremljeni na isti način kao što je opisano u prethodnom odjeljku. U procesu učenja za oba modela je kao funkcija gubitka korištena unakrsna entropija, a ADAM metoda je izabrana kao optimizacijski postupak. Prema proceduri koja je gore opisana, korištena je stopa učenja od $1 \cdot 10^{-4}$ uz veličinu grupe opažanja od 64 slike. Oba modela učena su do konvergencije, koja je bila određena uvjetom ranog zaustavljanja u slučaju rasta omjera gubitka na validacijskom skupu i gubitka na skupu za učenje iznad definiranog praga ili stagnacije pada gubitka na validacijskom skupu kroz tri uzastopne epohe. Za model kojem su ulaz bile slike iz kadra HE, u prvih 13 epoha učenje je bilo rađeno samo za dodane potpuno povezane slojeve, a od 14. epohe gradijent je propuštan i u zadnji rezidualni blok baznog modela

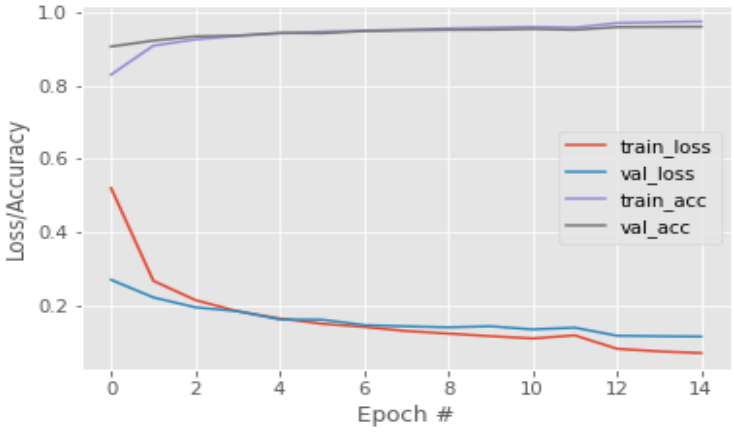
koji sadrži 1.050.624 parametara. U 15. epohi nakon što je uočena stagnacija u procesu učenja manualno je spuštena stopa učenja na $1 \cdot 10^{-5}$, jer je ova praksa prepoznata u radu [43] kod učenja ResNet50 modela. Proces učenja je zaustavljen nakon 15. epohe. Prikaz gubitka i točnosti kao funkcije broja epoha kod modela za slike iz kadra HE nalazi se na slici 5.5



Slika 5.5 Gubitak i točnost kao funkcije broja epoha kod modela za slike iz kadra HE

Po završetku učenja, točnost[24] klasifikacije modela na validacijskom skupu bila je 90,79%, dok je točnost u slučaju da se ne podešavaju parametri zadnjeg rezidualnog bloka baznog modela bila 89,73%.

Sličan postupak proveden je za model koji je učio na temelju slika iz kadra Fokus. Kod ovog modela prvih 11 epoha su učeni samo novi slojevi dok je od 12. epohe u učenje uključen i zadnji rezidualni blok ResNet50 modela. Proces učenja je zaustavljen u 13. epohi, nakon koje je točnost modela na validacijskom skupu bila 95,34%, a bez podešavanja rezidualnog bloka baznog modela točnost je bila 95,11% (vidi sliku 5.6).



Slika 5.6 Gubitak i točnost kao funkcije broja epoha kod modela za slike iz kadra Fokus

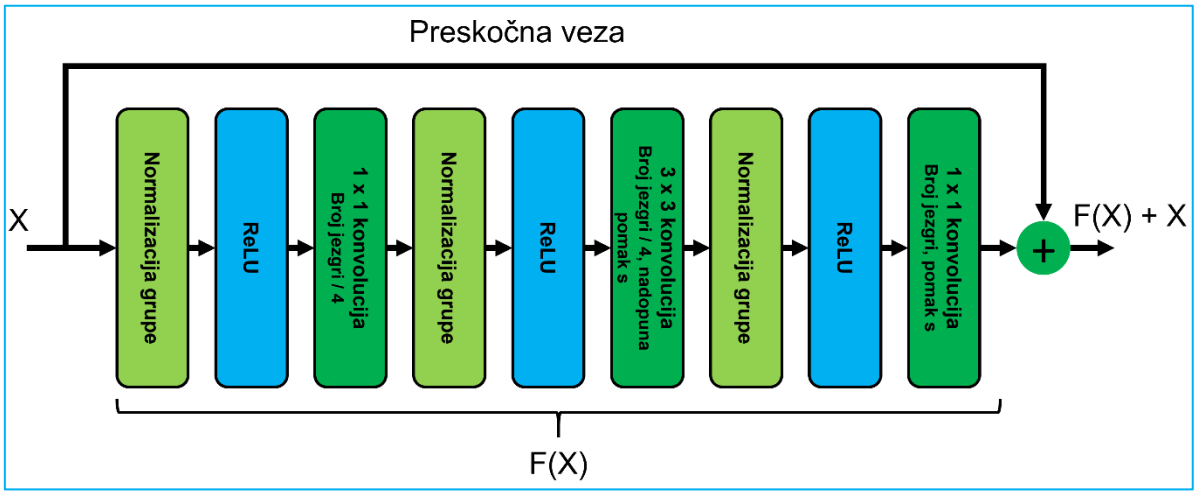
[24] Metrika točnosti bit će detaljno objašnjena u odjeljku vezanom za evaluaciju i izbor optimalnih modela.

Pitanje koje se nameće iz gore prikazanih rezultata je da li su ovako razvijeni modeli primjenjivi za rješavanje problema istovremenog prepoznavanja i vremenske segmentacije aktivnosti? S obzirom da je ovaj problem u kontekstu „*action segmentation*“ grupe pristupa formuliran kao problem klasifikacije svake sličice, napravljen je eksperiment u kojem su predikcije modela za pojedinačne slike iz istog video zapisa agregirane kako bi se utvrdila učinkovitost razvijenih modela na ovom problemu. Činjenica koju je potrebno uzeti u obzir kod ocjene učinkovitosti nekog modela za problem segmentacije je ta da metrika točnosti ništa ne govori o tome koliko je različitih vremenskih segmenata model prepoznao u video zapisu. Konkretnije, u kontekstu studija vremena iznimno je bitno znati točan broj aktivnosti u nekom procesu i njihov redoslijed, stoga model koji ne brine o ove dvije stvari nije odgovarajuće rješenje. Metrika koja kažnjava model u slučaju prevelikog ili premalog broja prepoznatih segmenata naziva se segmentacijska F1 metrika s definiranim pragom preklapanja segmenata te će njena svojstva i izračun biti tema narednih odjeljaka. U ovoj argumentaciji jedino što je bitno su rezultati razvijenih modela na segmentacijskoj F1 metrici uz prag preklapanja od 0,5. Ona je za model razvijen na slikama iz HE kadra na temelju validacijskog skupa iznosila 48,19%, a za model slika iz kadra Fokus 70,82%. Ovi rezultati nisu iznenađujući iz razloga što 2D CNN modeli nemaju sposobnost prepoznavanja vremenske zavisnosti između susjednih sličica u video zapisu pa imaju problema s određivanjem stvarnog broja aktivnosti i njihovog redoslijeda. Zbog toga se modeli razvijeni u ovom odjeljku koriste za izvlačenje značajki, a ne vremensku segmentaciju. Međutim, rezultati točnosti ovih modela služili su kao dobar indikator donje granice koju su morali nadmašiti finalni modeli za segmentaciju i klasifikaciju kako bi njihov razvoj i primjena bili opravdani.

Proces izvlačenja značajki za ova dva modela bio je sličan pristupu iz prethodnog odjeljka uz iznimku da su vektori značajki izvučeni iz prvog potpuno povezanog sloja novo dodanog modela, što znači da su značajke dobivene na temelju 27.730.944 parametara. Po izvlačenju značajki svih slika iz oba kadra kreirani su vektori značajki za Concat kadar kao i kod prethodno opisanog pristupa. Vrijeme učenja za pojedini kadar iznosilo je 23,2 minute po epohi, a za naknadno izvlačenje značajki bilo je potrebno 34 minute po kadru.

5.2.3 Učenje modela na pojedinačnim sličicama iz vlastitog uzorka i naknadno izvlačenje značajki – TB pristup

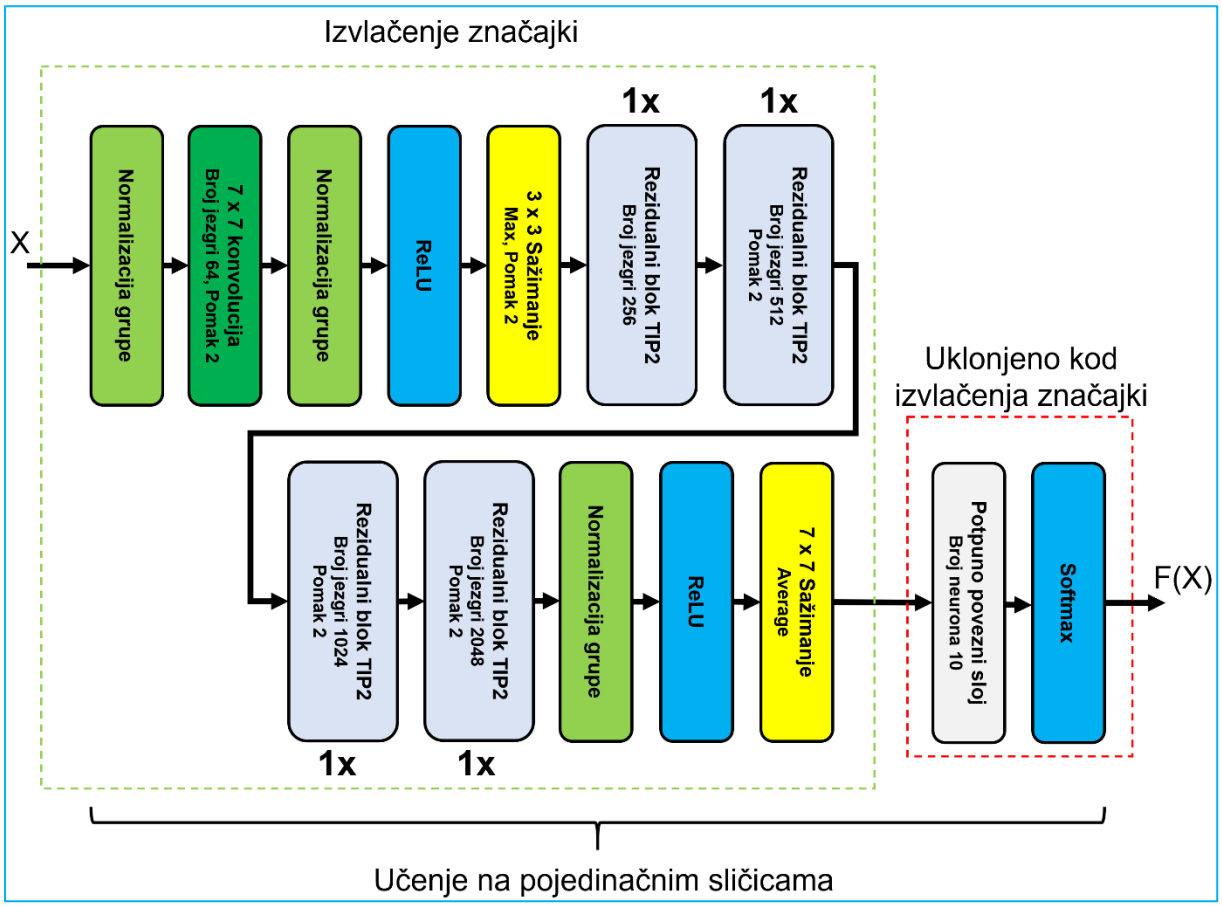
Posljednji pristup izvlačenja značajki koji je korišten u eksperimentima temeljen je na modelu koji je naučen samo na slikama iz prikupljenog uzorka. Razvijeni model u arhitekturi koristi vrstu rezidualnog bloka koja je predstavljena u istraživanju [102]. Ovaj blok je posljedica eksperimenata koji su uključivali sve moguće permutacije redoslijeda konvolucijskog sloja, sloja za normalizaciju grupe i aktivacijske funkcije te se raspored prikazan na slici 5.7 pokazao kao najbolji po pitanju točnosti modela.



Slika 5.7 Rezidualni blok TIP2 prema [102]

Rezidualni blok sa slike 5.7, kao i rezidualni blok (slika 5.2) korišten u ResNet50 modelu, sastoji se od tri konvolucijska sloja s gotovo istim postavom hiperparametara, pri čemu je razlika u tome da drugi i treći konvolucijski blok koriste pomak veličine S, koji je u eksperimentima postavljen na 2. Svrha svakog konvolucijskog sloja je ista kao i kod prethodno opisane verzije rezidualnog bloka iz odjeljka 5.2.1. U razvoju modela prikazanog na slici 5.8 korištene su dobre prakse uočene kod ResNet50 arhitekture. Prvi sloj modela je sloj za normalizaciju grupe, a razlog za to je želja da se izbjegne potreba za pripremom ulaznih slika centriranjem primjenom aritmetičkih sredina svih piksela pojedinog kanala. Izračun aritmetičke sredine svih piksela u pojedinom kanalu na temelju svih slika iz uzorka je zahtjevna operacija, a sličan efekt je moguće dobiti primjenom normalizacije grupe. Sljedeća tri sloja u modelu imaju isti raspored kao i slojevi unutar rezidualnog bloka opisanog u ovom odjeljku, pri čemu su hiperparametri konvolucijskog sloja isti kao i kod ResNet50 modela za prvi konvolucijski sloj. Prije prvog rezidualnog bloka radi se 3×3 sažimanje maksimalnom vrijednosti uz pomak veličine 2, nakon kojeg slijede četiri rezidualna bloka kod kojih se broj jezgri udvostručuje s

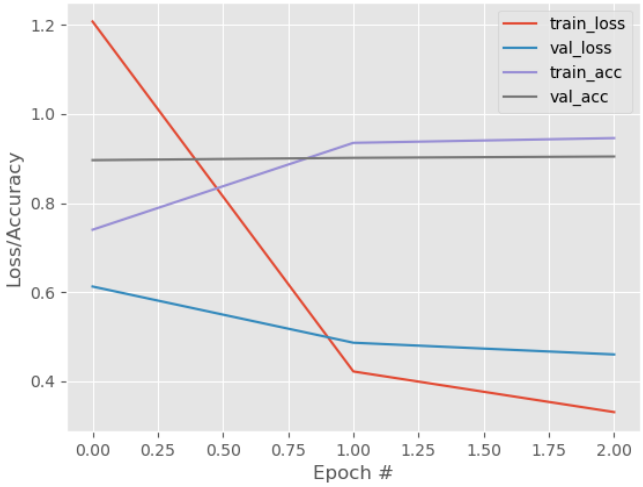
dubinom, po uzoru na ResNet50. Dimenzija transformiranih podataka nakon zadnjeg rezidualnog bloka je $7 \times 7 \times 2048$. Kako je cilj bio da izvučene značajke iz modela imaju iste dimenzije kao i one u prethodna dva pristupa, potrebno je bilo napraviti 7×7 sažimanje prosječnom vrijednosti, nakon čega je vektor značajki proslijeđen u potpuno povezani sloj odgovoran za klasifikaciju slika.



Slika 5.8 Novi model temeljen na rezidualnim blokovima za izvlačenje značajki

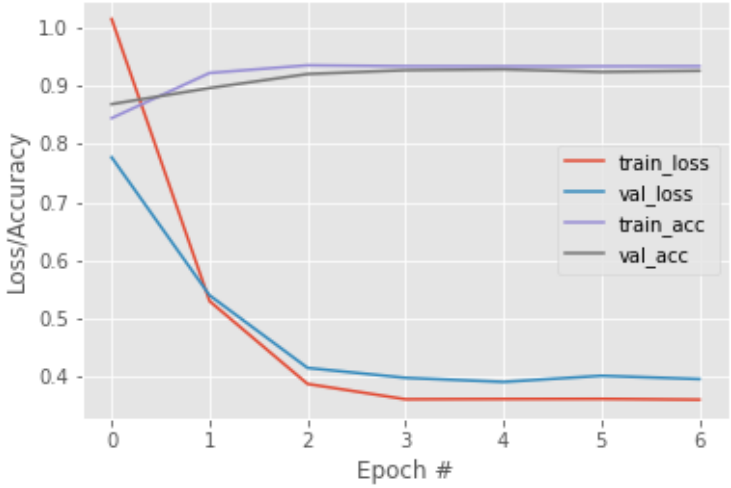
Model sa slike 5.8 sadrži 8.040.662 parametara, odnosno oko 3 puta manje nego ResNet50, što ima smisla s obzirom da je ovaj model učen na stotinjak tisuća slika u odnosu na ResNet50 koji je učen na ImageNet skupu podataka koji je znatno većeg obujma. Ponovno su naučeni posebni modeli za slike iz svakog kadra snimanja, kao i kod modela za prijenos znanja s finim podešavanjem. Pretpostavka je bila da bi ovi modeli trebali imati u najmanju ruku točnost koja je na razini najboljeg modela iz prethodnog pristupa, s obzirom da su u potpunosti temeljeni na vlastitom uzorku. U prvom eksperimentu oba modela su učena primjenom unakrsne entropije kao funkcije gubitka i stohastičkim gradijentnim spustom s faktorom momenta od 0,9 uz grupu opažanja od 64 slike. Zbog korištenja rezidualnih blokova s preskočnim vezama izabrana je stopa učenja od $1 \cdot 10^{-1}$ kako bi se ubrzala konvergencija modela, koja je i za ovaj pristup

definirana kriterijem ranog zaustavljanja u slučaju prevelikog razilaženja gubitka na validacijskom skupu i skupu za učenje ili stagnacije pada gubitka na validacijskom skupu kroz tri uzastopne epohe. Model učen na slikama iz kadra HE već je nakon tri epohe postigao uvjet zaustavljanja (slika 5.9), nakon čega je točnost na validacijskom skupu bila 90,45%.



Slika 5.9 Gubitak i točnost kao funkcije broja epoha kod modela za slike iz kadra HE, eksperiment 1

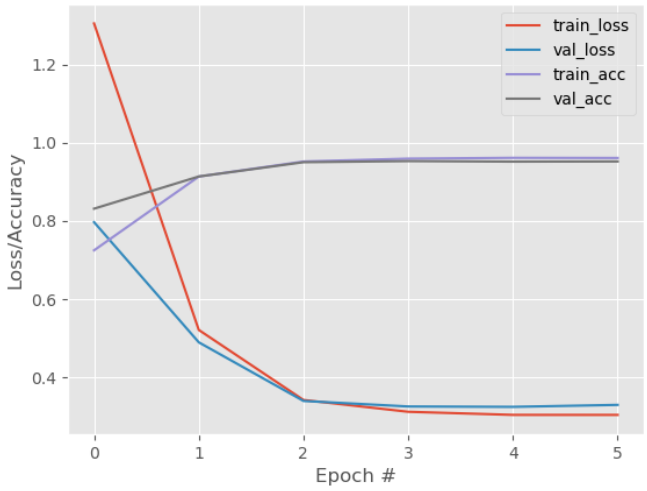
Slična situacija dogodila se i u prvom eksperimentu za model učen na slikama iz kadra Fokus, koji je postigao uvjet zaustavljanja nakon pet epoha (slika 5.10), pri čemu je točnost na validacijskom skupu bila 92,45%.



Slika 5.10 Gubitak i točnost kao funkcije broja epoha kod modela za slike iz kadra Fokus, eksperiment 1

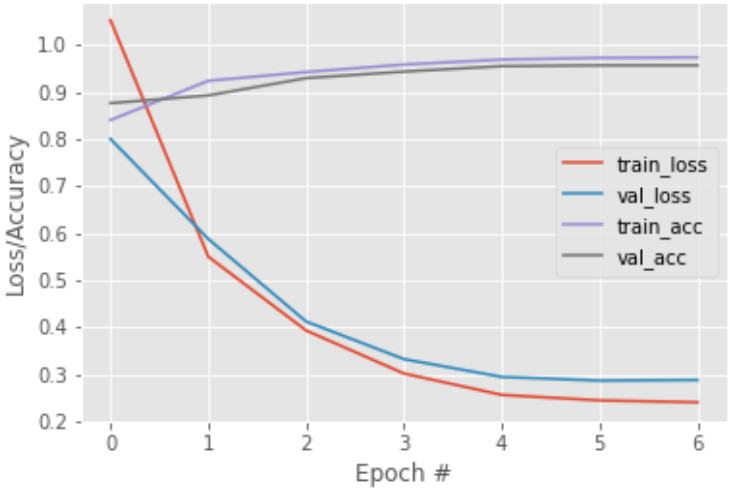
S obzirom da nije postignuta ciljana točnost niti kod jednog modela, odlučeno je da će biti napravljeni dodatni eksperimenti. Kako je kod modela za slike iz kadra HE došlo do

zaustavljanja učenja zbog brzog ulaska u područje prenaučenosti, u novom eksperimentu je za oba modela u funkciju gubitka dodana L2 regularizacija s faktorom od $\lambda = 5 \cdot 10^{-4}$. Proces učenja je kod modela za slike iz Fokus kadra zaustavljen zbog stagnacije u padu gubitka stoga je kod njega u postupak učenja dodan linearni raspored smanjenja stope učenja. Drugi eksperiment za model sa slikama iz HE kadra postigao je uvjet zaustavljanja u četvrtoj epohi, uz to da je nakon druge epohe stopa učenja manualno spuštena na $1 \cdot 10^{-3}$ (slika 5.11). Nakon ovog eksperimenta točnost na validacijskom skupu bila je 95,28 % što je bilo zadovoljavajuće.



Slika 5.11 Gubitak i točnost kao funkcije broja epoha kod modela za slike iz kadra HE, eksperiment 2

Uvjet zaustavljanja kod modela za obradu slika iz kadra Fokus ostvaren je nakon pet epoha u drugom eksperimentu (slika 5.12). Točnost modela bila je 95,48%.



Slika 5.12 Gubitak i točnost kao funkcije broja epoha kod modela za slike iz kadra Fokus, eksperiment 2

Za ove modele provjerena je vrijednost segmentacijske F1 metrike na validacijskom skupu. Za model razvijen na slikama iz kadra HE segmentacijska F1 metrika bila je 73,72%, a za model razvijen na slikama iz kadra Fokus 77,78%. Ovo i dalje nisu zadovoljavajuće vrijednosti za primjenu modela u području studija vremena, što stvara potrebu za razvojem modela koji su specijalizirani za istovremeno prepoznavanje aktivnosti i vremensku segmentaciju.

Izvlačenje značajki iz razvijenih modela slijedilo je sličan obrazac kao i kod prethodna dva pristupa, jedina razlika bila je u pripremi ulaznih slika, za koje je napravljeno skaliranje piksela u interval $[0, 1]$. Nakon odstranjivanja posljednjeg potpuno povezanog sloja modela, izračunati su vektori značajki dimenzije 2048 za svaku sliku. Vektori značajki za kadar Concat dobiveni su na ekvivalentan način kao i kod prethodna dva pristupa. Vrijeme učenja za pojedini kadar iznosilo je 38 minuta po epohi, a za naknadno izvlačenje značajki bilo je potrebno 15 minuta po kadru.

5.2.4 Zaključak o pristupima izvlačenja značajki

Usporedbom funkcionalnih karakteristika pristupa za izvlačenje značajki jasno je da se FE pristupom najprije dolazi do izračunatih značajki u ukupnom vremenu od 68 minuta. U vrijeme izvlačenja značajki za pristup TL i TB potrebno je ukalkulirati i vrijeme učenja. Iako je vrijeme učenja po epohi za TL pristup oko 39% kraće nego kod TB pristupa, ukupno vrijeme učenja je kraće za TB pristup jer puno prije dolazi do konvergencije. Ovo je vidljivo na primjeru učenja modela na slikama iz kadra HE, gdje je TL pristupom potrebno 15 epoha, a TB pristupom 5 epoha, uslijed čega je ukupno vrijeme učenja modela iz TB pristupa za 45% kraće u odnosu na TL pristup. Ovi odnosi su razumljivi jer na vrijeme trajanja epoha između ostalog utječe i broj parametara modela koji je za TB pristup nešto veći od 8 milijuna, dok se kod TL pristupa inicijalno radi učenje samo novo dodanih slojeva koji imaju oko 5 milijuna parametara. Konačno, samo izvlačenje značajki je kraće kod TB pristupa za 11 minuta u odnosu na TL pristup, na što također utječe broj parametara u modelima za izvlačenje značajki, gdje modeli iz TB pristupa imaju oko 3,4 puta manje parametara. Kvaliteta izvučenih značajki, u smislu njihovog utjecaja na učinkovitost modela za istovremeno prepoznavanje i vremensku segmentaciju, bit će analizirana kod evaluacije finalnih modela.

Ustanovljeno je da modeli iz TL i TB pristupa nisu sposobni samostalno riješiti problem istovremenog prepoznavanja i vremenske segmentacije aktivnosti na zadovoljavajući način. Tema sljedećeg odjeljka su upravo modeli specijalizirani za rješavanje ključnog problema predstavljenog u disertaciji.

5.3 Modeli za istovremeno prepoznavanje i vremensku segmentaciju

Pregled literature ukazao je na tri vrste arhitektura koje dominiraju u „*action segmentation*“ grupi istraživanja. Radi se o arhitekturama temeljenima na LSTM-u, dvosmjernom LSTM-u i dilatiranim 1D konvolucijskim slojevima. Kroz eksperimente istražena su svojstva novo razvijenih modela koji koriste gore navedene elemente u arhitekturi.

Modeli iz ovog odjeljka su kao ulaz dobivali podatke u obliku niza $\mathbf{X}_1^{\mathrm{T}} = \left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(\mathrm{T})}\right)$ ukupne duljine T, pri čemu je T u pravilu bio različit za svako opažanje iz razloga što su kao uzorci bili korišteni video zapisi različite duljine. Svaki član ulaznog niza u koraku t bio je vektor značajki $\boldsymbol{x}^{(t)} \in \mathbb{R}^{2048}$ za kadrove snimanja HE i Fokus, odnosno $\boldsymbol{x}^{(t)} \in \mathbb{R}^{4096}$ za kadar Concat. U eksperimentima su korišteni vektori značajki izvučeni nekim od tri moguća pristupa izvlačenja značajki. Zadatak modela bio je napraviti predikciju klase aktivnosti koja se odvija u svakom vremenskom koraku izlaznog niza $\hat{\mathbf{Y}}_1^{\mathrm{T}} = \left(\hat{\boldsymbol{y}}^{(1)}, \hat{\boldsymbol{y}}^{(2)}, \ldots, \hat{\boldsymbol{y}}^{(\mathrm{T})}\right)$ gdje je član predikcije izlaznog niza u koraku t vektor $\hat{\boldsymbol{y}}^{(t)} \in \mathbb{R}^{10}$ za kojeg vrijedi $\sum_{i=1}^{10} y_i^{(t)} = 1$.

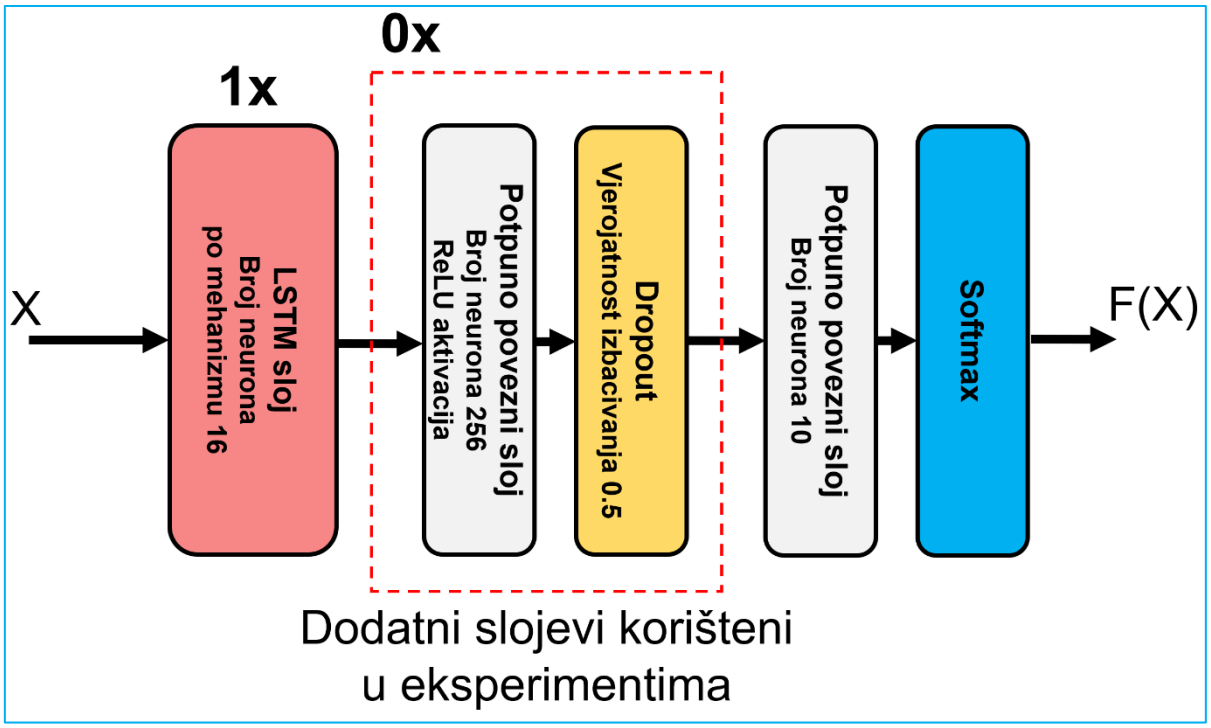
U naredna tri odjeljka ukratko će biti objašnjena tri tipa arhitektura korištena u eksperimentima, nakon čega će biti opisana metodologija koja je korištena kod učenja i izbora optimalnih hiperparametara modela te će cijelo poglavlje završiti s analizom učinkovitosti modela na temelju definirane metrike i razvijene procedure za usporedbu i izbor optimalnih modela.

5.3.1 Model temeljen na LSTM slojevima

Detaljan opis povratnih neuronskih mreža s LSTM slojevima dan je u odjeljku 2.3.4, a sažetak je da se radi o vrsti arhitekture koja je sposobna naučiti dugoročne vremenske zavisnosti u podatcima, jer su kod nje ublaženi problemi nestanka i eksplozije gradijenta. Prikaz arhitekture sa slike 5.13 sugerira da je u eksperimentima prvo definiran jedan ili više LSTM slojeva iza kojih slijedi jedan ili više potpuno povezanih slojeva, uz dropout sloj kojim se želi utjecati na pojavu prenaučenosti. Proces definiranja arhitekture vodila su sljedeća istraživačka pitanja:

1) Kakav će efekt na model imati veći broj LSTM slojeva, veći broj neurona po mehanizmima pojedinih LSTM slojeva i interakcija između ova dva hiperparametra?
2) Da li je nakon posljednjeg LSTM sloja, a prije sloja odgovornog za finalnu klasifikaciju, korisno uključiti i dodatni potpuno povezani sloj?

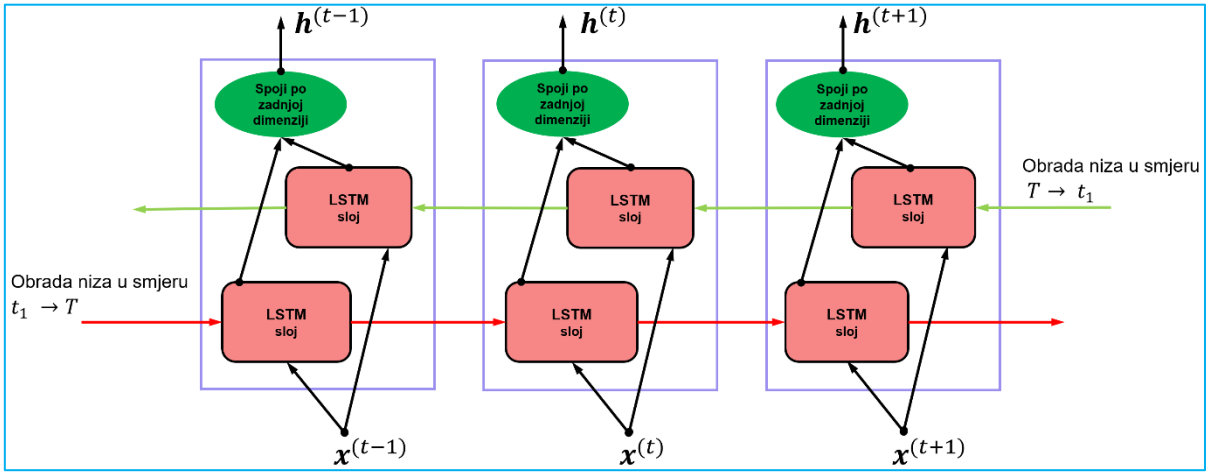
Konkretni hiperparametri s rasponima njihovih vrijednosti kod eksperimenata prikazani su u tablici 5.2, a određeni su na temelju literature i preliminarnih eksperimenata.



Slika 5.13 Arhitektura najboljeg LSTM modela

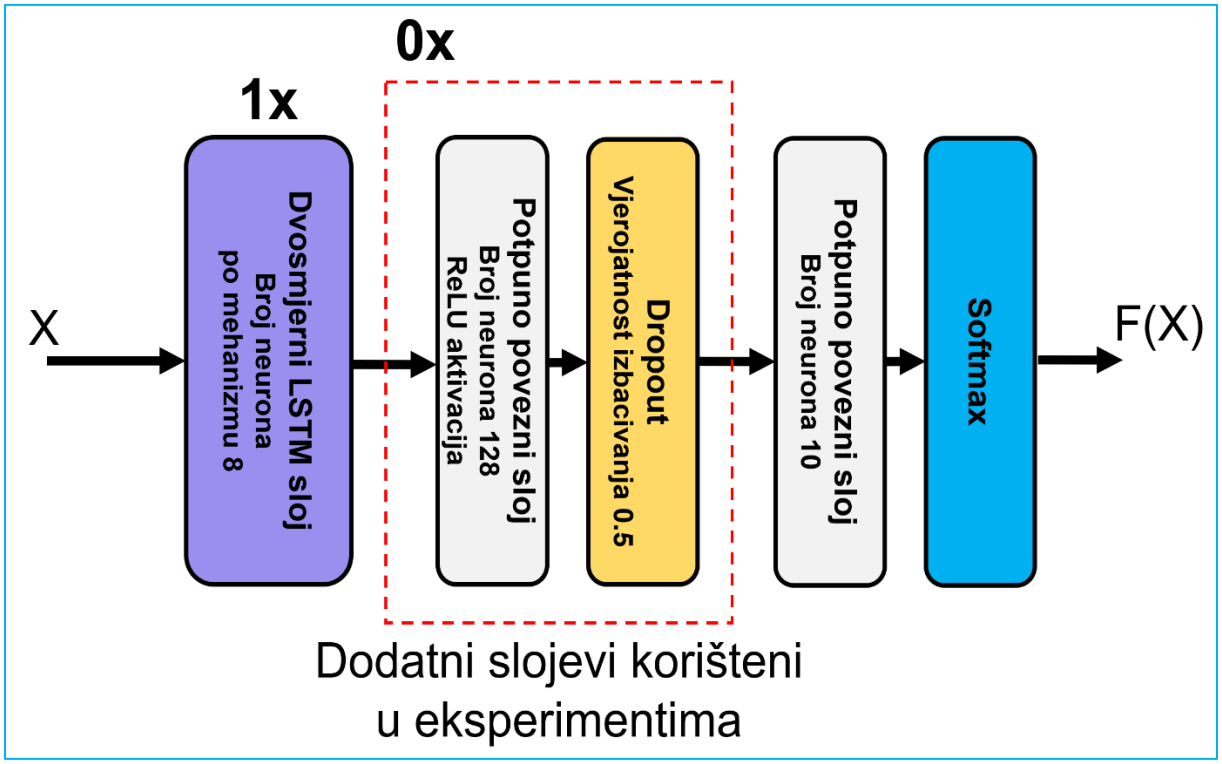
5.3.2 Model temeljen na dvosmjernim LSTM slojevima

Druga vrsta arhitekture koja je korištena u eksperimentima odnosi se na biLSTM slojeve. Glavno ograničenje povratnih neuronskih mreža je da predikciju u trenutnom vremenskom koraku rade na temelju trenutnog elementa ulaznog niza i svih prethodnih skrivenih stanja. Dvosmjerni LSTM omogućava da se u svakom vremenskom koraku koriste informacije iz cijelog niza, što je korisno svojstvo ako primjena modela nema ograničenja u vidu dostupnosti budućih vremenskih koraka. biLSTM je realiziran na način da se koriste dva paralelna LSTM sloja, pri čemu svaki obrađuje niz iz različitog smjera (vidi sliku 5.14).



Slika 5.14 Unutarnja struktura dvosmjernog LSTM-a za tri vremenska koraka

S obzirom da se unutar biLSTM sloja koriste dva paralelna LSTM sloja moguće je zaključiti da će se broj parametara udvostručiti. Izlazi iz paralelnih LSTM slojeva mogu se spajati na različite načine, ali ne postoje čvrsti teorijski dokazi koji je od tih načina najbolji. U disertaciji je korišteno spajanje naslaganjem izračunatih vektora značajki iz oba sloja.



Slika 5.15 Arhitektura najboljeg dvosmjernog LSTM modela

U izradi arhitekture sa slike 5.15 korištene su slične ideje i istraživačka pitanja kao i kod modela temeljenog na jednosmjernom LSTM-u, stoga su u eksperimentima podešavani hiperparametri iz tablice 5.2, koja je korištena i za model iz prethodnog odjeljka.

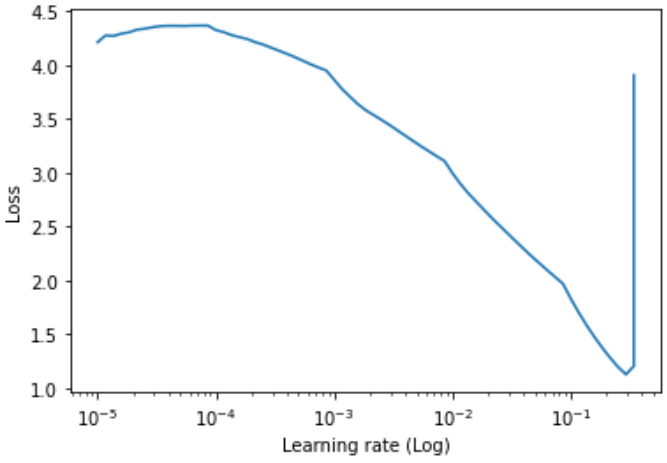
Tablica 5.2 Podešavani hiperparametri kod eksperimenata s LSTM i biLSTM modelima

Element algoritma	**Oznaka**	**Hiperparametar**	**Vrijednosti korištene u eksperimentima**
Model	LS	Broj LSTM/biLSTM slojeva	$\{1, 2, 3\}$
	LN	Broj neurona po mehanizmu LSTM/biLSTM sloja	$\{4, 8, 16, 32, 64, 128, 256, 512, 1024\}$
	PS	Broj dodatnih potpuno povezanih slojeva	$\{0, 1\}$
	PN	Broj neurona u dodatnim potpuno povezanim slojevima	$\{128, 256, 512\}$
	VDR	Vjerojatnost izbacivanja kod dropout sloja	$\{0;\ 0{,}1;\ 0{,}2;\ 0{,}5\}$

Element algoritma	Oznaka	Hiperparametar	Vrijednosti korištene u eksperimentima
Optim. metoda	OM	Optimizacijska metoda	{ADAM , SGD + faktor momenta od 0.9}
	SU	Stopa učenja	$\{5 \cdot 10^{-4}, 1 \cdot 10^{-4}, 1 \cdot 10^{-3}, 5 \cdot 10^{-2}, 1 \cdot 10^{-2}, 1 \cdot 10^{-1}, 1\}$
	RSU	Raspored stope učenja	{bez rasporeda, linearno smanjenje , ciklički raspored}
	VG	Veličina grupe opažanja	$\{1, 16, 32\}$
Funkcija gubitka	FG	Funkcija gubitka	{gubitak unakrsne entropije , gubitak unakrsne entropije + segmentacijski gubitak}
	REG	λ faktor regularizacije kod potpuno povezanih slojeva	$\{0;\ 5 \cdot 10^{-4}\}$

Pregledom tablice 5.2, moguće je uočiti dva elementa koji su prethodno spomenuti ali nisu detaljnije objašnjeni. Radi se o cikličkom rasporedu stope učenja i segmentacijskom gubitku.

Ciklički raspored stope učenja (engl. *cycle scheduling*) predložen je u radu [113] sa svrhom da se olakša izbor inicijalne stope učenje te optimizacijskoj metodi omogući bijeg iz loših lokalnih optimuma ili sedlastih točaka. Ovo se postiže na način da stopa učenja oscilira prema unaprijed definiranoj funkciji između minimalne i maksimalne vrijednosti. U istom radu predložen je i način kako odrediti minimalnu i maksimalnu vrijednost stope učenja, za što se koristi jednostavni algoritam povećavanja preliminarne vrijednosti stope učenja za fiksni faktor do uvjeta zaustavljanja. Nakon toga se pregledom grafa gubitka kao funkcije stope učenja odaberu minimalna i maksimalna vrijednost, tako da se pronađe prva vrijednost stope učenja za koju je gubitak počeo padati, a za maksimalnu vrijednost se odabere 10 puta manja vrijednost od one pri kojoj počinje divergencija. Na grafu sa slike 5.16 moguće je demonstrirati primjenu opisanog heurističkog pravila kojim su odabrane granice stope učenja od $1 \cdot 10^{-4}$ i $3 \cdot 10^{-2}$.



Slika 5.16 Graf gubitka kao funkcije stope učenja za izbor optimalnih granica stope učenja

Empirijski rezultati ukazuju na to da primjena opisane heuristike i cikličkog rasporeda stope učenja zahtjeva znatno manju količinu eksperimenata u odnosu na iscrpno traženje optimalne stope učenja za postizanje gotovo identične točnosti [113], zbog čega je odlučeno da će u eksperimentima, između ostalog, biti korišten i ovaj raspored stope učenja.

Segmentacijski gubitak je osmišljen u istraživanju [70] kako bi se riješio problem prekomjerne segmentacije koji se može pojaviti kod predikcije modela iz „*action segmentation*“ domene, a koristi se kao dodatak gubitku unakrsne entropije kako je pokazano u jednadžbi (5.1).

$$L = L_{CE} + \lambda L_{seg} \tag{5.1}$$

U izrazu (5.1) L_{CE} je gubitak unakrsne entropije, L_{seg} je segmentacijski gubitak, a λ je faktor kojim je određena relativna važnost ova dva gubitka, a obično se traži u intervalu $[0{,}05;\ 0{,}25]$. L_{seg} gubitak je temeljen na ograničenom srednjem kvadratu odstupanja (engl. *truncated mean squared error*). Konkretnije, segmentacijski gubitak za pojedini niz se računa prema izrazu (5.2), u kojem je T duljina niza, a C je ukupan broj različitih klasa. $\tilde{\Delta}_{t,k}$ je odstupanje između predikcije za vremenski korak t i klasu k te predikcije prethodnog vremenskog koraka za istu klasu koje je ograničeno konstantom τ kao što je prikazano u jednadžbi (5.3), pri čemu je τ hiperparametar kojeg je potrebno podesiti, a uobičajene vrijednosti su iz skupa $\{3, 4, 5\}$.

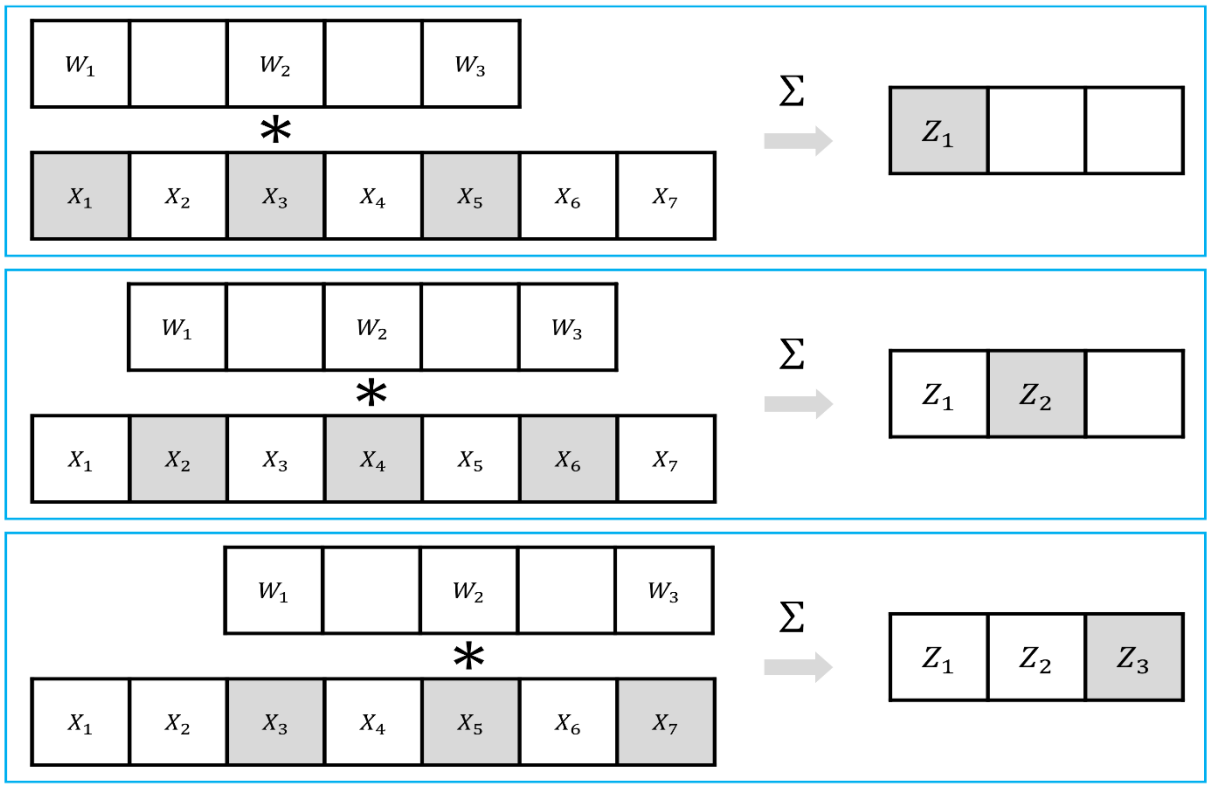
$$L_{seg} = \frac{1}{TC}\sum_{t}^{T}\sum_{k}^{C}\tilde{\Delta}_{t,k}^{2} \tag{5.2}$$

$$\tilde{\Delta}_{t,k} = \begin{cases} \Delta_{t,k} & za\ \Delta_{t,k} \leq \tau \\ \tau & inače \end{cases} \qquad \Delta_{t,k} = \left|\ln \hat{y}_k^{(t)} - \ln \hat{y}_k^{(t-1)}\right| \tag{5.3}$$

Intuicija je da ovaj gubitak kažnjava odstupanja u vrijednosti predikcija susjednih sličica, odnosno pokušava usmjeriti model da daje jednake predikcije susjednih sličica pod pretpostavkom da nije došlo do promjene klase. Ovaj gubitak može štetno djelovati na učenje modela u slučaju da hiperparametar τ nije pravilno podešen, jer će penalizirati model u situacijama kada on predviđa s visokom vjerojatnosti da je došlo do promjene klase. U eksperimentima su korištene varijante gubitka unakrsne entropije, sa i bez segmentacijskog gubitka.

5.3.3 Model temeljen na dilatiranim 1D konvolucijskim slojevima

Primjena dilatiranih 1D konvolucijskih slojeva u arhitekturi omogućava efikasno modeliranje vremenskih nizova, iz razloga što dilatiranje radi na principu povećanja jezgre kroz dodavanje praznina između pojedinih elemenata, zbog čega je uslijed proširenja receptivnog polja moguće uhvatiti duži vremenski kontekst bez povećanja broja parametara. Na slici 5.17 ilustriran je prolaz unaprijed za dilatirani 1D konvolucijski sloj, na kojem je korištena jezgra **W** dimenzija 1×3 s faktorom dilatacije $D = 2$, koja se pomiče po ulaznoj mapi **X** dimenzija 1×7 što rezultira izlaznom mapom **Z** dimenzija 1×3. U praksi se kod opisivanja 1D konvolucije u notaciji zanemaruje prva dimenzija tenzora **W**, **X** i **Z** koja je uvijek jednaka jedinici te će ova konvencija biti korištena u nastavku teksta. U općem slučaju 1D konvolucije, jezgra je tenzor 3. reda $\mathbf{W} \in \mathbb{R}^{N \times K \times D}$, a ulazna i izlazna mapa su matrice $\boldsymbol{X} \in \mathbb{R}^{W \times D}$ i $\boldsymbol{Z} \in \mathbb{R}^{J \times K}$.

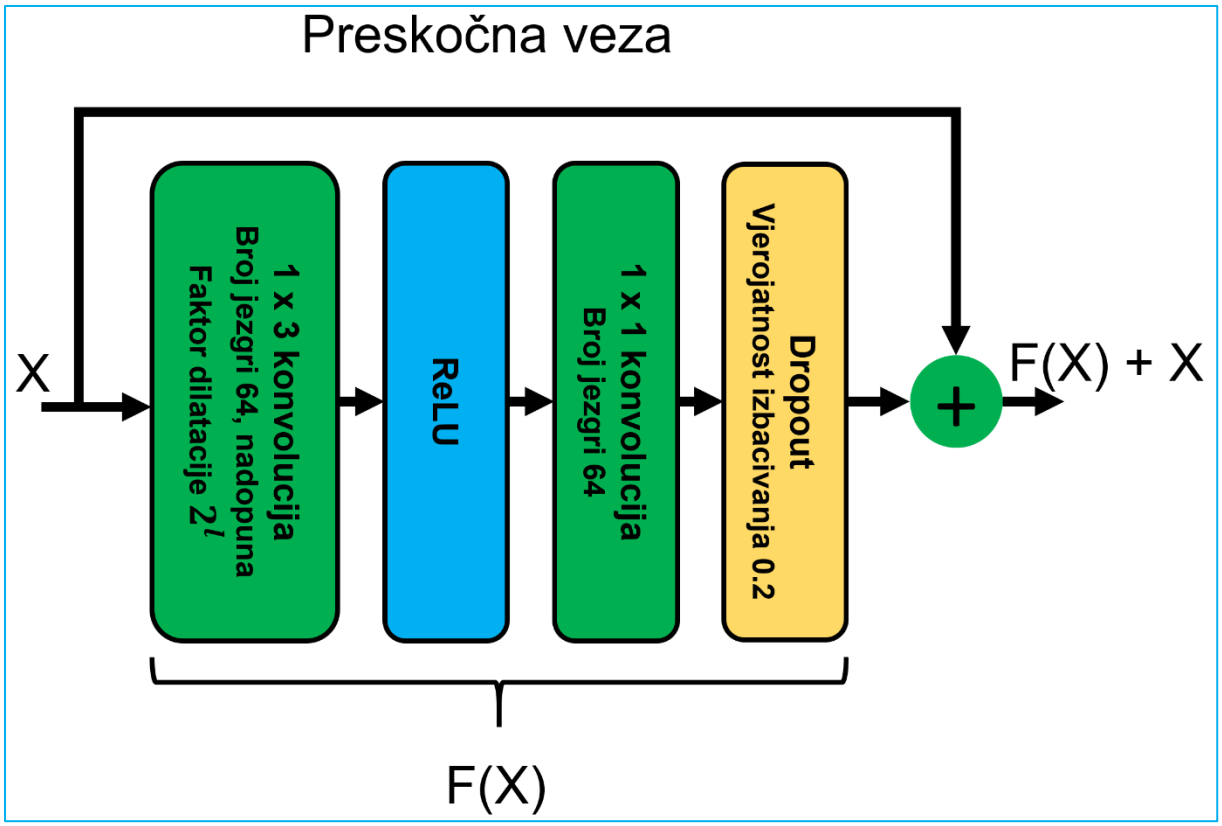


Slika 5.17 Prolaz unaprijed kroz jedan sloj dilatirane konvolucije s faktorom dilatacije 2

Izračun J dimenzije izlazne mape $\boldsymbol{Z}$ nakon primjene dilatirane 1D konvolucije na ulaznu mapu $\boldsymbol{X}$ slijedi formulu (5.4), gdje je P faktor dopunjavanja nulom, S je korak pomicanja jezgre, a $\lfloor \cdot \rfloor$ je operator koji vraća najmanju cjelobrojnu vrijednost (engl. *floor operator*).

$$J = \left\lfloor \frac{W + 2P - N - (N-1)(D-1)}{S} \right\rfloor + 1 \tag{5.4}$$

Konvolucijski modeli[25] iz ovog odjeljka temeljeni su na dvije vrste dilatiranih konvolucijskih blokova prepoznatih u istraživanjima [23,70]. Prvi od njih je dilatirani rezidualni blok koji je prikazan na slici 5.18. Ovaj blok započinje s dilatiranim konvolucijskim slojem dimenzije 3, pri čemu je faktor dilatacije D za l-ti sloj jednak 2^l, dok je broj jezgri bio hiperparametar koji je podešavan u eksperimentima. Ideja iza definiranja faktora dilatacije kao funkcije broja slojeva je ta da se s povećanjem dubine modela nastoji uhvatiti sve veći vremenski kontekst. Nakon prvog konvolucijskog sloja slijedi nelinearna transformacija ReLU aktivacijskom funkcijom i konvolucija dimenzije 1, čiji je broj jezgri bio hiperparametar koji je podešavan u eksperimentima. Svrha prethodno spomenutog konvolucijskog sloja je kontroliranje reprezentativnog kapaciteta modela kroz povećanje ili smanjenje broja jezgri. Dropout sloj je dodan s ciljem da se po potrebi utječe na prenaučenost modela kroz podešavanje vjerojatnosti izbacivanja. Dodatno je priključena i preskočna veza kako bi se olakšao proces učenja.

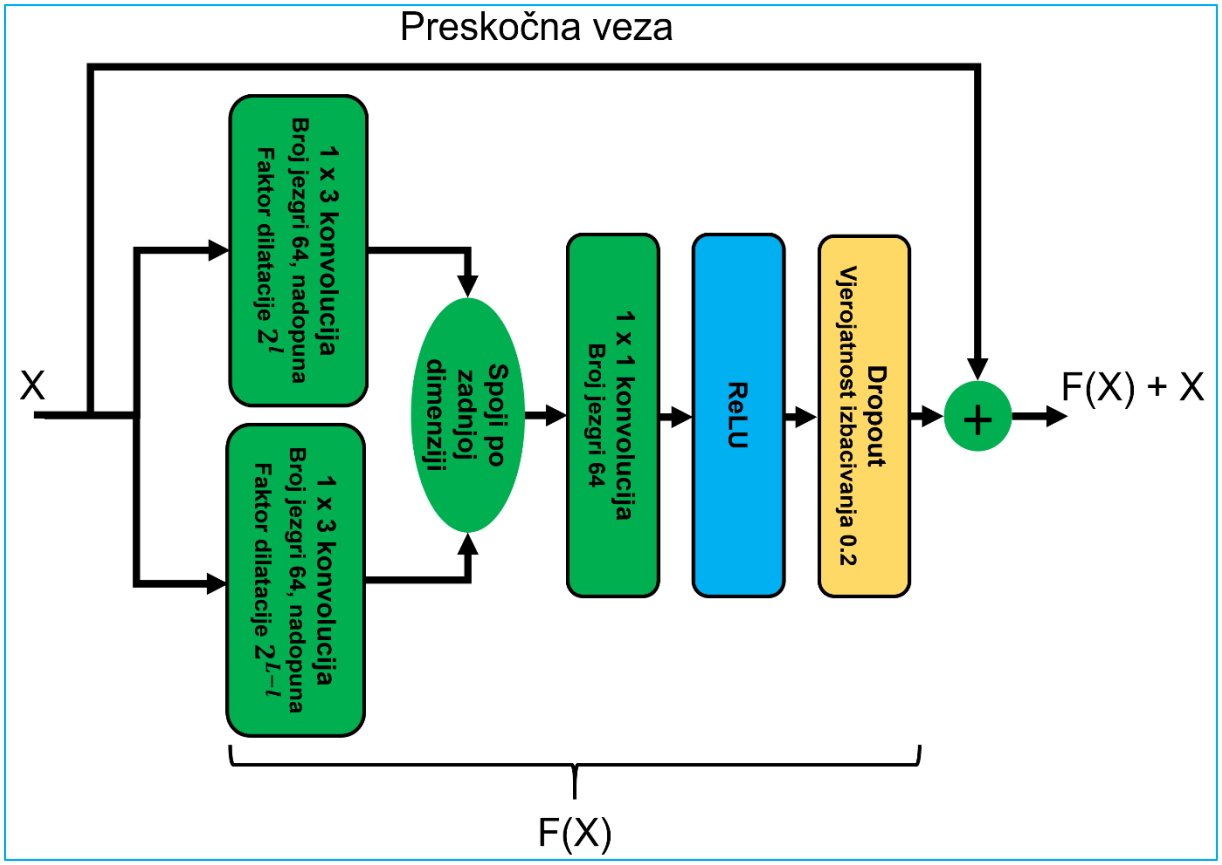


Slika 5.18 Rezidualni dilatirani blok prema [70]

Druga vrsta rezidualnog bloka koja je korištena kod razvijenih konvolucijskih modela je dualni dilatirani blok (vidi sliku 5.19). Glavna značajka ovog bloka je da koristi dvije grane dilatiranih

[25] U sljedećim odjeljcima za modele s ovom arhitekturom koristit će se oznaka CONV.

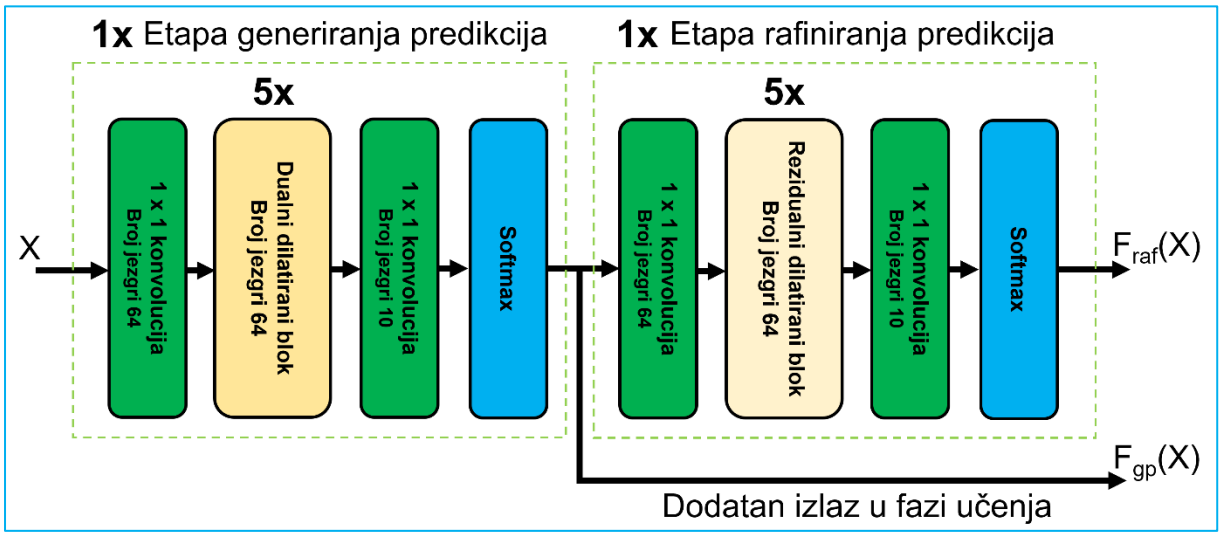
konvolucijskih slojeva dimenzije 3. U jednoj grani faktor dilatacije za l-ti sloj jednak je 2^l, a u drugoj grani jednak je 2^{L-l}, gdje je L oznaka zadnjeg sloja. Drugim riječima, u jednoj grani receptivno polje konvolucijskog sloja raste s dubinom modela, a u drugoj grani se receptivno polje smanjuje od najveće vrijednosti do najmanje. Definiranje ovakve arhitekture je potaknuto ciljem da se istodobno želi uhvatiti kratkoročni i dugoročni vremenski kontekst prisutan u podatcima. Broj jezgri konvolucijskih slojeva u obje grane je hiperparametar koji je podešavan u eksperimentima. Izlazi iz ove dvije grane se spajaju naslaganjem po dimenziji dubine izlaznih tenzora. S obzirom da arhitektura ovog bloka sadrži preskočnu vezu, nužno je bilo napraviti podešavanje dimenzionalnosti tenzora dobivenog spajanjem dvije konvolucijske grane, a za što je iskorištena konvolucija dimenzije 1. Izlazni tenzor iz posljednjeg konvolucijskog sloja prolazi kroz ReLU aktivacijsku funkciju s ciljem povećanja reprezentativnog kapaciteta modela, dok se dropout slojem po potrebi povećava ili smanjuje stupanj regularizacije.



Slika 5.19 Dualni dilatirani blok prema [23]

Konačno, arhitektura konvolucijskog modela koji sadrži dvije opisane vrste blokova (vidi sliku 5.20) može se podijeliti u dva dijela koji su u literaturi nazvani etapa za generiranje predikcija i etapa za rafiniranje predikcija [23]. Etapa za generiranje predikcija sastoji se od odgovarajućeg

broja dualnih dilatiranih blokova, kojima prethodi konvolucija dimenzije 1, za koju je već ranije spomenuto da služi za podešavanje dimenzije ulaznih podataka kroz definiranje broja jezgri. Nakon završnog dualnog dilatiranog bloka slijedi konvolucija dimenzije 1 čiji broj jezgri je uvjetovan brojem klasa aktivnosti u uzorku iz razloga što je upravo on odgovoran za generiranje predikcija. Broj dualnih dilatiranih blokova je hiperparametar koji je podešavan u eksperimentima. Preliminarni eksperimenti ukazali su na to da je samo jedna etapa generiranja predikcija dovoljna za modele razvijene na prikupljenom uzorku. Etapa rafiniranja predikcija slijedi slična načela kao i etapa generiranja predikcija samo što su umjesto dualnih dilatiranih blokova korišteni rezidualni dilatirani blokovi. Kao što sam naziv kaže, svrha etapa rafiniranja je da poboljšaju predikcije svake prethodne etape. Ovo je vidljivo iz toga da je ulaz u i-tu etapu rafiniranja tenzor predikcija dobiven iz softmax sloja etape rafiniranja $i-1$. Broj etapa rafiniranja te broj rezidualnih dilatiranih blokova bili su hiperparametri koji su podešavani u eksperimentima. Kod ove arhitekture nisu korišteni slojevi sažimanja jer bi oni doveli do smanjenja vremenske dimenzije čime bi se izgubile informacije o finim (kratkim) aktivnostima.



Slika 5.20 Arhitektura najboljeg konvolucijskog modela

Kod eksperimentiranja je istražen utjecaj broja dodatnih izlaza u procesu učenja te korištenja dijeljenih težina između etapa rafiniranja. Pretpostavka je bila da će veći broj izlaza pomoći u konvergenciji modela, pri čemu će su u fazi testiranja korištene predikcije samo iz izlaza posljednjeg sloja modela. Dijeljenje težina između etapa rafiniranja predloženo je u radu [70] gdje je argumentirano da je zadatak svake etape rafiniranja sličan pa ima smisla da dijele težine. Inicijalna pretpostavka u ovom istraživanju je bila da će dijeljenje težina pomoći u slučaju prenaučenosti te da će dovesti do brže konvergencije u odnosu na model iste dubine, ali bez

dijeljenih težina. S druge strane, također je bilo očekivano da bi dijeljenje težina potencijalno moglo smanjiti točnost modela uslijed smanjenog reprezentativnog kapaciteta. Popis svih hiperparametara koji su podešavani u eksperimentima s konvolucijskim modelima nalazi se u tablici 5.3. Metodologija kojom su tražene optimalne vrijednosti hiperparametara svih opisanih modela objašnjena je u nastavku.

Tablica 5.3 Podešavani hiperparametri kod eksperimenata s konvolucijskim modelima

Element algoritma	**Oznaka**	**Hiperparametar**	**Vrijednosti korištene u eksperimentima**
Model	DRS	Broj dilatiranih rezidualnih slojeva	$\{1, 5, 10\}$
	DDS	Broj dualnih dilatiranih slojeva	$\{1, 5, 11\}$
	ER	Broj etapa rafiniranja	$\{1, 2, 3, 4, 5, 6\}$
	DT	Dijeljenje težina između etapa rafiniranja	$\{\text{Ne}, \text{Da}\}$
	BI	Broj izlaza iz modela	{samo iz zadnjeg sloja, nakon svake etape}
	BJ	Broj jezgri u svakom konvolucijskom sloju	$\{64, 128\}$
	VDR	Vjerojatnost izbacivanja kod dropout sloja	$\{0;\ 0{,}1;\ 0{,}2;\ 0{,}5\}$
Optim. metoda	OM	Optimizacijska metoda	{ADAM, SGD + faktor momenta od 0,9}
	SU	Stopa učenja	$\{5 \cdot 10^{-4}, 1 \cdot 10^{-4}, 1 \cdot 10^{-3}, 5 \cdot 10^{-2}, 1 \cdot 10^{-2}, 1 \cdot 10^{-1}, 1\}$
	RSU	Raspored stope učenja	{bez rasporeda, linearno smanjenje, ciklički raspored}
	VG	Veličina grupe opažanja	$\{1, 16, 32\}$
Funkcija gubitka	FG	Funkcija gubitka	{gubitak unakrsne entropije, gubitak unakrsne entropije +segmentacijski gubitak}

5.3.4 Metodologija kod učenja i izbora optimalnih hiperparametara modela

U odjeljku 2.3.5 spomenuto je da je ključ procesa učenja u pronalasku odgovarajuće složenosti modela. Na složenost modela utječe odabir hiperparametara. U navedenom odjeljku opisan je utjecaj pojedinih hiperparametara na svojstva pristranosti i varijance modela, pri čemu je bitno pronaći ravnotežu između ove dvije krajnosti. Odnos pristranosti i varijance moguće je pratiti za vrijeme procesa učenja na temelju kretanja vrijednosti, trendova i odnosa između gubitka na skupu za učenje $J(\boldsymbol{W})_{train}$ i generalizacijskog gubitka $J(\boldsymbol{W})_{test}$, koji se tijekom podešavanja hiperparametara procjenjuje na skupu za validaciju $\mathcal{P}_{val}$. Kvalitetan model je onaj koji ima mali

generalizacijski gubitak. Također, rečeno je da postoji više različitih načina traženja hiperparametara. Kako su u pregledu literature prepoznate učinkovite arhitekture kod problema istovremenog prepoznavanja aktivnosti i vremenske segmentacije, stečena je dobra slika o inicijalnim postavkama hiperparametara modela iz domene, što otvara mogućnost za korištenje pristupa ručnog traženja hiperparametara.

Sve prethodno konstatirane činjenice čine osnovu metodologije koja je korištena u procesu učenja modela u okviru ovog istraživanja. Metodologija slijedi nekoliko jednostavnih principa i heuristika, koje proizlaze iz teorijskih i praktičnih saznanja opisanih u dijelu 2.3, a to su:

1) Inicijalni eksperiment započni s postavkama modela na temelju preporuka iz literature;
2) U procesu učenja pohrani vrijednosti parametara modela svakih 10 epoha te zaustavi eksperiment u slučaju da pad vrijednosti funkcije gubitka na skupu za validaciju stagnira kroz 5 uzastopnih epoha ili omjer između gubitka na validacijskom skupu i skupu za učenja prekorači definirani prag;
3) Generiraj graf metrike učinkovitosti i funkcije gubitka za podatke iz skupa za učenje i validacijskog skupa te analiziraj trendove pazeći na odnos pristranosti i varijance;
4) U sljedećem eksperimentu ažuriraj vrijednosti hiperparametara na temelju zaključaka o stanju pristranosti i varijance. Ako se radi o početnim eksperimentima koristi *princip ortogonalnosti*[26] uz fokus na ključne hiperparametre poput stope učenja i veličine modela po pitanju broja slojeva i neurona u slojevima;
5) Ponavljaj korake 2), 3) i 4) sve dok nije postignuta ciljana učinkovitost modela ili nije prekoračeno ograničenje vremenskog resursa u vidu maksimalnog broja eksperimenata ili maksimalnog dozvoljenog vremena učenja;
6) Kada je postignuta zadovoljavajuća vrijednost metrike učinkovitosti, a nije prekoračeno ograničenje vremenskog resursa, fokusiraj se na smanjenje broja parametara i skraćenje vremena konvergencije modela ponavljanjem koraka 5).

Opisana metodologija bit će demonstrirana u nastavku teksta na primjeru traženja optimalnog konvolucijskog modela na podatcima prikupljenima iz kadra HE i značajkama izvučenima pristupom TB. Sažetci eksperimenata za spomenuti model strukturirani su na način da sadrže sljedeće elemente:

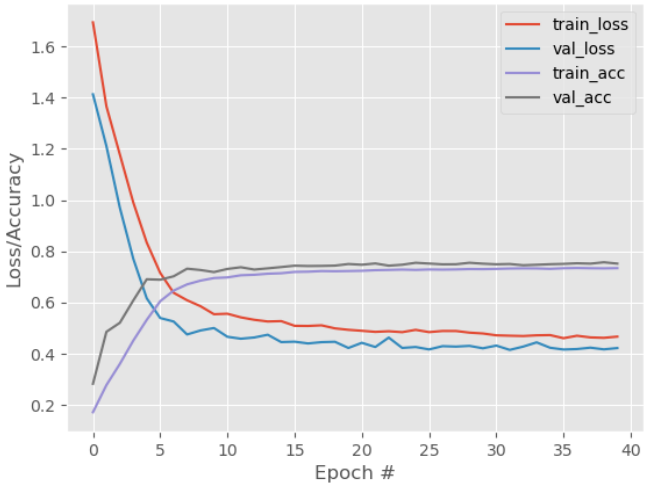
[26] Princip ortogonalnosti se odnosi na praksu mijenjanja vrijednosti samo **jednog** hiperparametra kako bi se ustanovio njegov utjecaj na funkciju gubitka i metriku učinkovitosti.

a) *Pregled vrijednosti korištenih hiperparametara*. Hiperparametri su organizirani po elementima algoritma pri čemu su korištene oznake iz tablice 5.3. S ciljem jednostavnijeg praćenja ažuriranih vrijednosti hiperparametara između eksperimenata, ažurirane vrijednosti su podcrtane.
b) *Rezultati točnosti*. Dane su vrijednosti točnosti na skupu za učenje i validaciju. Za određene eksperimente prikazani su grafovi točnosti i funkcije gubitka kao funkcija broja epoha, za oba skupa, kako bi se ukazalo na trendove u procesu učenja.
c) *Funkcionalne karakteristike modela*. Za ovaj dio korištene su informacije o broju parametara modela i vremenu učenja. Vrijeme učenja je iskazano preko ukupnog broja epoha prije prekida eksperimenta uslijed stagnacije pada funkcije gubitka na validacijskom skupu te vremena trajanja pojedine epohe.
d) *Zaključak eksperimenta*. Kratak zapis o trenutku kada je primijećena stagnacija procesa učenja te kakav je odnos pristranosti i varijance modela. Na temelju odnosa pristranosti i varijance donesene su odluke o tome na koje komponente algoritma je potrebno djelovati kako bi se potencijalno poboljšali rezultati modela, koristeći saznanja opisana u odjeljku 2.3.5.

Eksperiment 1

Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=3; **DT**=Ne, **BI**=samo zadnji sloj; **BJ**=64, **VDR**=0,5
Optim. metoda	**OM**=ADAM; **SU**=$5 \cdot 10^{-4}$; **RSU**=bez rasporeda; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=73,46%; $\boldsymbol{Točnost_{val}}$=75,24%
Funkcionalnost	**Broj parametara**=993.768; **Broj epoha**=20; **Trajanje epohe**=9 s

Zaključak: Nakon 20 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu, a gubitak na trening skupu jako sporo pada i točnost jako sporo raste (slika 5.21). Također, točnost na validacijskom skupu je veća nego na trening skupu uslijed korištenja regularizacije u vidu dropout tehnike. Rezultati ukazuju na to da problem nije u varijanci, već u visokoj pristranosti s obzirom da je omjer gubitaka oba skupa gotovo nepromijenjen kroz epohe, a nije ostvarena ciljana točnost od 95% (vidi odjeljak 5.2). Potrebno je stoga povećati broj slojeva i neurona i/ili promijeniti stopu učenja i optimizacijsku metodu i/ili smanjiti regularizaciju.



Slika 5.21 Gubitak i točnost kao funkcije broja epoha za eksperiment 1

Eksperiment 2

Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=3; **DT**=Ne, **BI**=samo zadnji sloj; **BJ**=128, **VDR**=0,5
Optim. metoda	**OM**=ADAM; **SU**=$5 \cdot 10^{-4}$; **RSU**=bez rasporeda; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=72,49%; $\boldsymbol{Točnost_{val}}$=75,44%
Funkcionalnost	**Broj parametara**=3.691.432; **Broj epoha**=12; **Trajanje epohe**=14 s

Zaključak: Nakon 12 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu, a gubitak na trening skupu jako sporo pada i točnost jako sporo raste. Rezultati ukazuju na to da problem nije u varijanci, već u visokoj pristranosti. Povećanje modela na način da s udvostruči broj jezgri konvolucijskih slojeva ne dovodi do poboljšanja učinkovitosti. Potrebno je povećati broj slojeva i/ili promijeniti stopu učenja i optimizacijsku metodu i/ili smanjiti regularizaciju.

Eksperiment 3

Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=5; **DT**=Ne, **BI**=samo zadnji sloj; **BJ**=64, **VDR**=0,5
Optim. metoda	**OM**=ADAM; **SU**=$5 \cdot 10^{-4}$; **RSU**=bez rasporeda; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=71,44%; $\boldsymbol{Točnost_{val}}$=74,24%
Funkcionalnost	**Broj parametara**=1.326.716; **Broj epoha**=12; **Trajanje epohe**=11 s

Zaključak: Nakon 12 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu, a gubitak na trening skupu jako sporo pada i točnost jako sporo raste. Rezultati ukazuju da problem nije u varijanci, već u visokoj pristranosti. Povećanje modela na način da se poveća broja etapa rafiniranja ne dovodi do poboljšanja učinkovitosti Moguće promjene hiperparametara uključuju promjenu stope učenja i optimizacijske metode i smanjenje regularizacije.

Eksperiment 4

Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=3; **DT**=Ne, **BI**=samo zadnji sloj; **BJ**=64, **VDR**=0,5
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$1 \cdot 10^{-1}$; **RSU**=linearno smanjenje; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=73,68%; $\boldsymbol{Točnost_{val}}$=75,17%
Funkcionalnost	**Broj parametara**=993.768; **Broj epoha**=16; **Trajanje epohe**=9 s

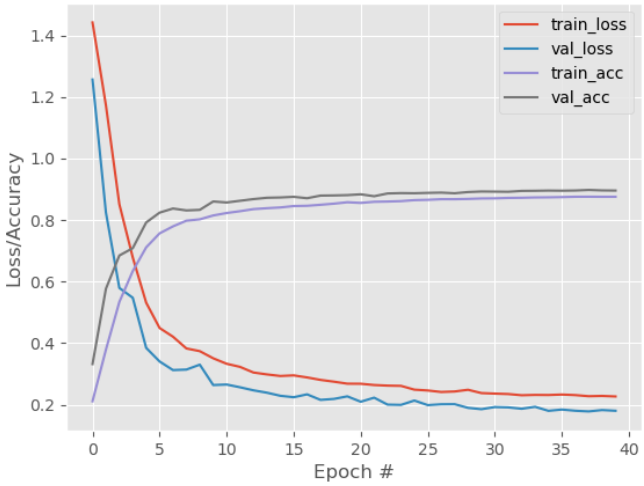
Zaključak: Nakon 16 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu, a gubitak na trening skupu jako sporo pada i točnost jako sporo raste. Rezultati ukazuju na to da problem nije u varijanci, već u visokoj pristranosti. Promjena optimizacijskog postupka uz veliku stopu učenja ne pomaže. Potrebno je smanjiti stopu učenja i/ili smanjiti regularizaciju.

Eksperiment 5

Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=3; **DT**=Ne, **BI**=samo zadnji sloj; **BJ**=64, **VDR**=0,2
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$1 \cdot 10^{-2}$; **RSU**=linearno smanjenje; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=87,80%; $\boldsymbol{Točnost_{val}}$=89,56%
Funkcionalnost	**Broj parametara**=993.768; **Broj epoha**=24; **Trajanje epohe**=9 s

Zaključak: Nakon 24 epohe stagnira rast točnosti i pad gubitka na validacijskom skupu, a gubitak na trening skupu jako sporo pada i točnost jako sporo raste (slika 5.22). Rezultati ukazuju na to da problem nije u varijanci, već u visokoj pristranosti. Ostvaren je prvi značajan napredak u vidu rasta točnosti s 75% na 89,6%. Faktori koji su utjecali na ovaj rezultat su niža stopa učenja i smanjenje stope izbacivanja dropout regularizacijske tehnike s 0,5 na 0,2.

Provjereno je što se događa ako se stopa učenja ponovno poveća na $1 \cdot 10^{-1}$ ili dropout tehnika ukloni. U slučaju povećanja stope učenja dolazi do divergencije u procesu učenja, a u slučaju uklanjanja regularizacije nema nikakvih dodatnih pozitivnih pomaka kod učinkovitosti. S obzirom da dolazi jako rano do stagnacije pada gubitka, bit će uklonjeno linearno smanjenje stope učenja.

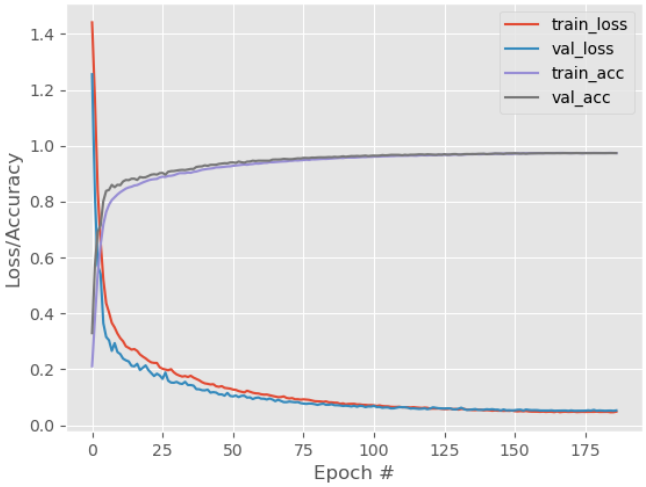


Slika 5.22 Gubitak i točnost kao funkcije broja epoha za eksperiment 5

Eksperiment 6

Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=3; **DT**=Ne, **BI**=samo zadnji sloj; **BJ**=64, **VDR**=0,2
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$1 \cdot 10^{-2}$; **RSU**=bez rasporeda; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=97,37%; $\boldsymbol{Točnost_{val}}$=97,35%
Funkcionalnost	**Broj parametara**=993.768; **Broj epoha**=130; **Trajanje epohe**=9 s

Zaključak: Nakon 130 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu (slika 5.23). Ovi rezultati ukazuju na to da je linearno smanjenje stope učenja loše djelovalo na proces učenja na način da je dovodilo do rane stagnacije. Prebačena je ciljana točnost od 95% na validacijskom skupa, ali rezultati ukazuju da i dalje nije prisutan problem visoke varijance, stoga i dalje ima smisla pokušavati povećati točnost. Potrebno je povećati model i/ili mijenjati stopu učenja.



Slika 5.23 Gubitak i točnost kao funkcije broja epoha za eksperiment 6

Eksperiment 7

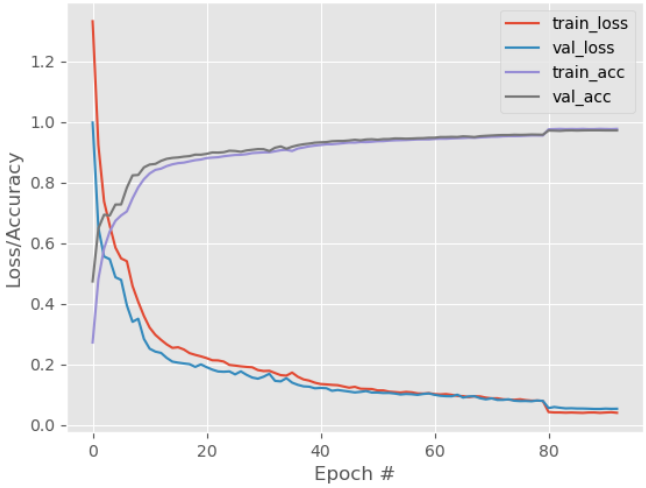
Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=3; **DT**=Ne, **BI**=samo zadnji sloj; **BJ**=64, **VDR**=0,2
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$[1 \cdot 10^{-2}, 5 \cdot 10^{-2}]$; **RSU**=ciklički raspored; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=91,52%; $\boldsymbol{Točnost_{val}}$=90,74%
Funkcionalnost	**Broj parametara**=993.768; **Broj epoha**=80; **Trajanje epohe**=9 s

Zaključak: Nakon 80 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu. Ovaj eksperiment je pokazao da ciklički raspored stope učenja ne pomaže u kombinaciji s postojećim postavom hiperparametara jer dolazi do smanjenja točnosti. Potrebno je povećati model i ukloniti ciklički raspored stope učenja.

Eksperiment 8

Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=3; **DT**=Ne, **BI**=samo zadnji sloj; **BJ**=128, **VDR**=0,2
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$1 \cdot 10^{-2}$; **RSU**=bez rasporeda; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=97,75%; $\boldsymbol{Točnost_{val}}$=97,25%
Funkcionalnost	**Broj parametara**=3.691.432; **Broj epoha**=80; **Trajanje epohe**=15 s

Zaključak: Nakon 80 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu (slika 5.24). Ovaj eksperiment je pokazao da 3,5 puta veći broj parametara ne dovodi do značajnog poboljšanja u točnosti u odnosu na model s 64 jezgre u konvolucijskim slojevima. U 80 epohi manualno je spuštena stopa učenja na $1 \cdot 10^{-4}$. Kako i dalje nema jasnih dokaza da model ima problema s prenaučenosti, potrebno je probati poboljšati učinkovitost modela na način da se poveća broj slojeva i smanji broj jezgri.

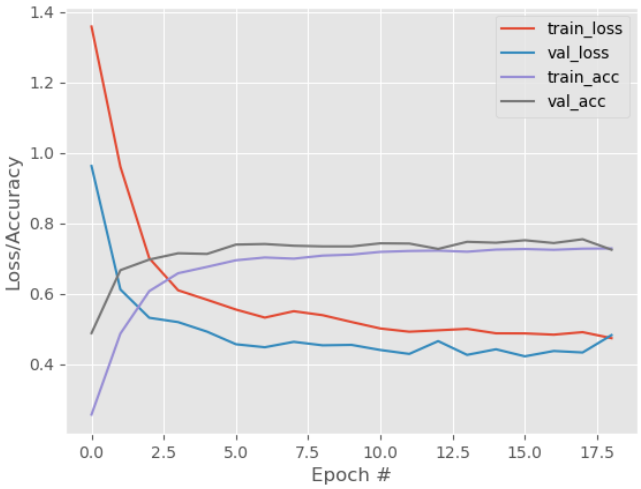


Slika 5.24 Gubitak i točnost kao funkcije broja epoha za eksperiment 8

Eksperiment 9

Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=6; **DT**=Ne, **BI**=samo zadnji sloj; **BJ**=64, **VDR**=0,2
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$1 \cdot 10^{-2}$; **RSU**=bez rasporeda; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=72,82%; $\boldsymbol{Točnost_{val}}$=72,47%
Funkcionalnost	**Broj parametara**=1.493.190; **Broj epoha**=10; **Trajanje epohe**=14 s

Zaključak: Nakon 10 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu (slika 5.25). Ovaj eksperiment je pokazao da veća dubina, ali 2 puta manji broj parametara u odnosu na prethodni eksperiment, dovode do brze stagnacije procesa učenja. Potrebno se vratiti na prethodne postavke hiperparametara i provjeriti je li moguće poboljšati učinkovitost primjenom drugačije funkcije gubitka.



Slika 5.25 Gubitak i točnost kao funkcije broja epoha za eksperiment 9

Eksperiment 10

Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=3; **DT**=Ne, **BI**=samo zadnji sloj; **BJ**=64, **VDR**=0,2
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$1 \cdot 10^{-2}$; **RSU**=bez rasporeda; **VG**=32
Funkcija gubitka	Unakrsna entropija + segmentacijski gubitak
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=98,00%; $\boldsymbol{Točnost_{val}}$=97,56%
Funkcionalnost	**Broj parametara**=993.768; **Broj epoha**=240; **Trajanje epohe**=9 s

Zaključak: Nakon 240 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu. Ovaj eksperiment je pokazao da nova funkcija gubitka ne donosi znatno poboljšanje točnosti na validacijskom skupu, a proces učenja je sporiji u odnosu na model iz eksperimenta 6 za 84%. Kako je već ranije prebačena ciljana točnost, u sljedećim eksperimentima fokus će biti na poboljšanju funkcionalnih karakteristika modela uz očuvanje postojeće točnosti. Cilj poboljšanja funkcionalnih karakteristika moguće je ostvariti smanjenjem broja parametara modela, a u ovom specifičnom slučaju to je moguće postići manjim brojem etapa rafiniranja ili dijeljenjem težina u etapama rafiniranja ili manjim brojem slojeva u etapi rafiniranja i etapi generiranja inicijalnih predikcija. Bržu konvergenciju moguće je postići bolje podešenim elementima optimizacijskog postupka, ali i dodatcima na arhitekturi poput izlaza iz svake etape modela, a posljedično će i manji broj parametara utjecati na ubrzavanje procesa učenja.

Eksperiment 11

Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=2; **DT**=Ne, **BI**=samo zadnji sloj; **BJ**=64, **VDR**=0,2
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$1 \cdot 10^{-2}$; **RSU**=bez rasporeda; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=98,32%; $\boldsymbol{Točnost_{val}}$=97,82%
Funkcionalnost	**Broj parametara**=827.294; **Broj epoha**=200; **Trajanje epohe**=7 s

Zaključak: Nakon 200 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu. Ovaj eksperiment je pokazao da manji broj etapa u odnosu na eksperimente 6 i 8 rezultira boljom točnosti na validacijskom skupu, uz 16% manji broj parametara u odnosu na model iz eksperimenta 6, ali uz 20% duže vrijeme učenja u odnosu na spomenuti model. Potrebno je provjeriti da li je korisno napraviti dodatno smanjenje broja etapa rafiniranja.

Eksperiment 12

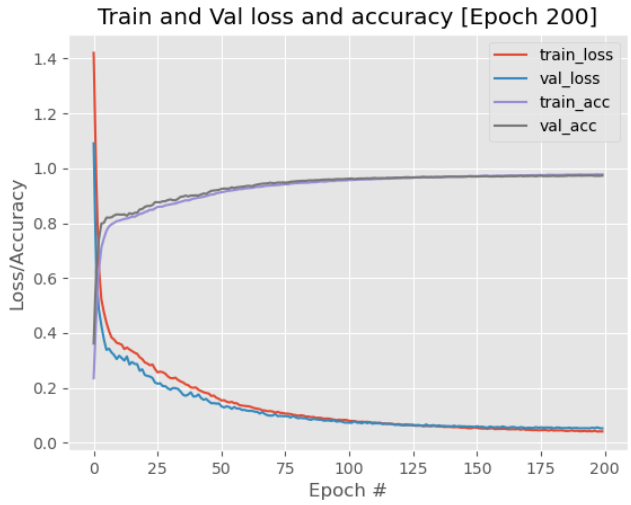
Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=1; **DT**=Ne, **BI**=samo zadnji sloj; **BJ**=64, **VDR**=0,2
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$1 \cdot 10^{-2}$; **RSU**=bez rasporeda; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=72,85%; $\boldsymbol{Točnost_{val}}$=73,83%
Funkcionalnost	**Broj parametara**=660.820; **Broj epoha**=20; **Trajanje epohe**=6 s

Zaključak: Nakon 20 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu. Ovaj eksperiment je pokazao da jedna etapa rafiniranja (mala dubina) nije dostatna za ostvarenje ciljane točnosti u kombinaciji s postojećim vrijednostima hiperparametara. Umjesto da se smanji dubina potrebno je probati uvesti dijeljenje parametara između etapa rafiniranja.

Eksperiment 13

Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=2; **DT**=Da, **BI**=samo zadnji sloj; **BJ**=64, **VDR**=0,2
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$1 \cdot 10^{-2}$; **RSU**=bez rasporeda; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=97,77%; $\boldsymbol{Točnost_{val}}$=97,32%
Funkcionalnost	**Broj parametara**=660.820; **Broj epoha**=160; **Trajanje epohe**=7 s

Zaključak: Nakon 160 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu (slika 5.26). Ovaj eksperiment je pokazao da model s dvije etape rafiniranja, uz dijeljenje težina, rezultira sličnom točnosti kao i model iz eksperimenta 6, ali uz 33% manje parametara. Potrebno je provjeriti da li dijeljenje težina i veći broj etapa rafiniranja mogu utjecati na povećanje točnosti.



Slika 5.26 Gubitak i točnost kao funkcije broja epoha za eksperiment 13

Eksperiment 14

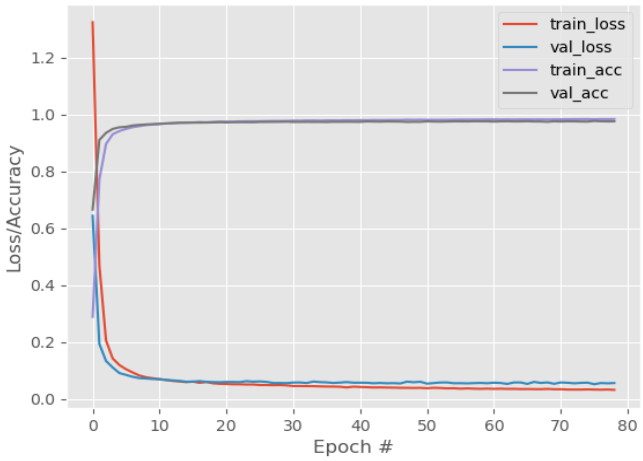
Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=4; **DT**=Da, **BI**=samo zadnji sloj; **BJ**=64, **VDR**=0,2
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$1 \cdot 10^{-2}$; **RSU**=bez rasporeda; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=97,42%; $\boldsymbol{Točnost_{val}}$=97,09%
Funkcionalnost	**Broj parametara**=660.820; **Broj epoha**=240; **Trajanje epohe**=10 s

Zaključak: Nakon 240 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu. Ovaj eksperiment je pokazao da model s četiri etape rafiniranja, uz dijeljenje težina, rezultira nešto manjom točnosti od modela iz eksperimenta 13, ali uz 114% duže vrijeme učenja u odnosu na taj model. Potrebno je provjeriti da li korištenje izlaza iz modela za svaku etapu može ubrzati proces učenja.

Eksperiment 15

Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=2; **DT**=Da, **BI**=svaka etapa; **BJ**=64, **VDR**=0,2
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$1 \cdot 10^{-2}$; **RSU**=bez rasporeda; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=98,37%; $\boldsymbol{Točnost_{val}}$=97,70%
Funkcionalnost	**Broj parametara**=660.820; **Broj epoha**=40; **Trajanje epohe**=7 s

Zaključak: Nakon 40 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu (slika 5.27). Ovaj eksperiment je pokazao da model s dvije etape rafiniranja, uz dijeljenje težina, i izlazom iz svake etape, rezultira neznatno većom točnosti od modela iz eksperimenta 13, ali uz 75% kraće vrijeme učenja u odnosu na taj model. Potrebno je provjeriti da li korištenje rasporeda stope učenja može poboljšati točnost modela s postojećim postavom hiperparametara.

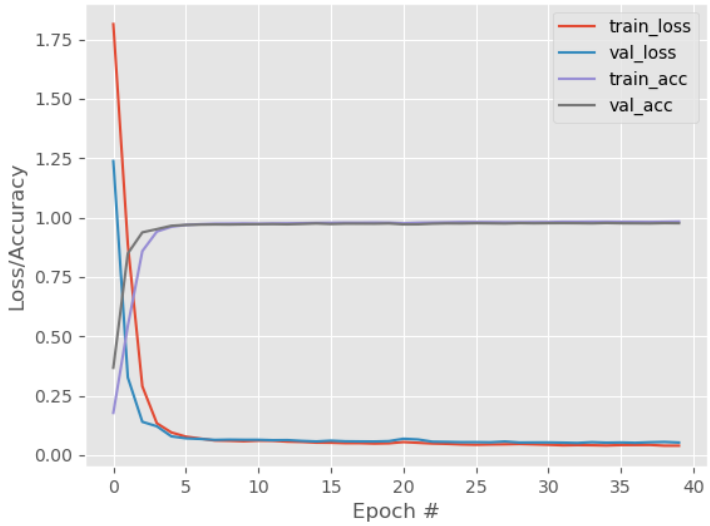


Slika 5.27 Gubitak i točnost kao funkcije broja epoha za eksperiment 15

Eksperiment 16

Dio algoritma	**Hiperparametri**
Model	**DRS**=10; **DDS**=11; **ER**=2; **DT**=Da, **BI**=svaka etapa; **BJ**=64, **VDR**=0,5
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$[1 \cdot 10^{-4}, 1 \cdot 10^{-2}]$; **RSU**=ciklički raspored; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=97,86%; $\boldsymbol{Točnost_{val}}$=97,41%
Funkcionalnost	**Broj parametara**=660.820; **Broj epoha**=10; **Trajanje epohe**=7 s

Zaključak: Nakon 10 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu (slika 5.28). Ovaj eksperiment je pokazao da dodatak cikličkog rasporeda stope učenja ubrzava konvergenciju za 75% u odnosu na model iz eksperimenta 15, uz sličnu točnost. Potrebno je provjeriti da li je uz ovaj postav hiperparametara moguće smanjiti broj slojeva unutar etape generiranja inicijalnih predikcija i etapa za rafiniranje predikcija.



Slika 5.28 Gubitak i točnost kao funkcije broja epoha za eksperiment 16

Eksperiment 17

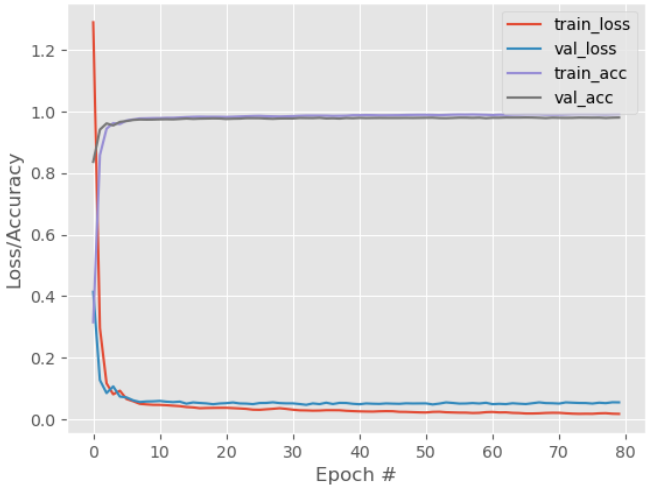
Dio algoritma	**Hiperparametri**
Model	**DRS**=5; **DDS**=11; **ER**=2; **DT**=Da, **BI**=svaka etapa; **BJ**=64, **VDR**=0,5
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$[1 \cdot 10^{-4}, 1 \cdot 10^{-2}]$; **RSU**=ciklički raspored; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=97,83%; $\boldsymbol{Točnost_{val}}$=97,41%
Funkcionalnost	**Broj parametara**=578.260; **Broj epoha**=20; **Trajanje epohe**=6 s

Zaključak: Nakon 20 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu. Ovaj eksperiment je pokazao da smanjenje broja dilatiranih rezidualnih slojeva s 10 na 5, ne umanjuje točnost modela, a istovremeno dovodi do smanjenja broja parametara za 12% u odnosu na model iz eksperimenta 16. U sljedećem eksperimentu potrebno je provjeriti da li je moguće smanjiti broj dualnih dilatiranih slojeva u etapi za generiranje predikcija, a s obzirom da se u etapama rafiniranja dijele težine moguće je probati i s korištenjem samo jedne etape.

Eksperiment 18

Dio algoritma	**Hiperparametri**
Model	**DRS**=5; **DDS**=5; **ER**=1; **DT**=Da, **BI**=svaka etapa; **BJ**=64, **VDR**=0,5
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$[1 \cdot 10^{-4}, 1 \cdot 10^{-2}]$; **RSU**=ciklički raspored; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=98,11%; $\boldsymbol{Točnost_{val}}$=97,59%
Funkcionalnost	**Broj parametara**=380.500; **Broj epoha**=40; **Trajanje epohe**=3 s

Zaključak: Nakon 40 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu (slika 5.29). Ovaj eksperiment je pokazao da je samo pet slojeva u etapi za generiranje predikcija uz samo jednu etapu rafiniranja s pet slojeva, dovoljno za postizanje točnosti koja je usporediva s onom modela iz eksperimenta 17, a s jednakim vremenom učenja i 34% manjim brojem parametara. Potrebno je provjeriti da li je moguća daljnja redukcija broja parametara u spomenute dvije etape.



Slika 5.29 Gubitak i točnost kao funkcije broja epoha za eksperiment 18

Eksperiment 19

Dio algoritma	**Hiperparametri**
Model	**DRS**=1; **DDS**=1; **ER**=1; **DT**=Da, **BI**=svaka etapa; **BJ**=64, **VDR**=0,5
Optim. metoda	**OM**=SGD+moment 0.9; **SU**=$[1 \cdot 10^{-4}, 1 \cdot 10^{-2}]$; **RSU**=ciklički raspored; **VG**=32
Funkcija gubitka	Unakrsna entropija
Rezultati eksperimenta	
Učinkovitost	$\boldsymbol{Točnost_{train}}$=92,82%; $\boldsymbol{Točnost_{val}}$=92,57%
Funkcionalnost	**Broj parametara**=182.612; **Broj epoha**=40; **Trajanje epohe**=2 s

Zaključak: Nakon 40 epoha stagnira rast točnosti i pad gubitka na validacijskom skupu. Ovaj eksperiment je pokazao da model s jednim slojem u etapi za generiranje predikcija i jednim slojem u etapi rafiniranja, nedovoljan za postizanje željene točnosti.

Na temelju provedenih eksperimenata moguće je zaključiti da je u 18. eksperimentu pronađen najbolji model, jer ima optimalnu kombinaciju točnosti, broja parametara i vremena učenja. Na sličan način provedeni su eksperimenti za preostalih 26 modela.

Po završenom procesu učenja i odabiru najboljih modela na temelju rezultata ostvarenih na skupu za validaciju $\mathcal{P}_{val}$, sljedeći korak je dati završnu ocjenu modela. Za završnu ocjenu modela upotrijebljen je skup podataka koji nije korišten ni za učenje ni za izbor hiperparametara kako bi se dobila nepristrana procjena generalizacijske sposobnosti modela što je jedna od tema sljedećeg odjeljka.

5.4 Evaluacija i izbor optimalnih modela

U ovom dijelu rada bit će prikazani rezultati evaluacije razvijenih modela za istovremeno prepoznavanje i vremensku segmentaciju aktivnosti na skupu podataka za testiranje $\mathcal{P}_{test}$. Kao što je pokazano u prethodnom odjeljku, proces razvoja modela bio je primarno vođen postignutom točnosti na skupu za validaciju $\mathcal{P}_{val}$ uz sekundarni cilj ograničavanja broja parametara i vremena potrebnog za učenje modela. S obzirom da je kreirano 27 modela koji se razlikuju po pitanju načina prikupljanja ulaznih podataka, pristupu korištenom kod izvlačenja značajki i arhitekturi, razvijena je procedura na temelju koje je moguće provesti usporedbu i izbor najboljeg modela. Modeli su ocijenjeni na temelju četiri elementa:

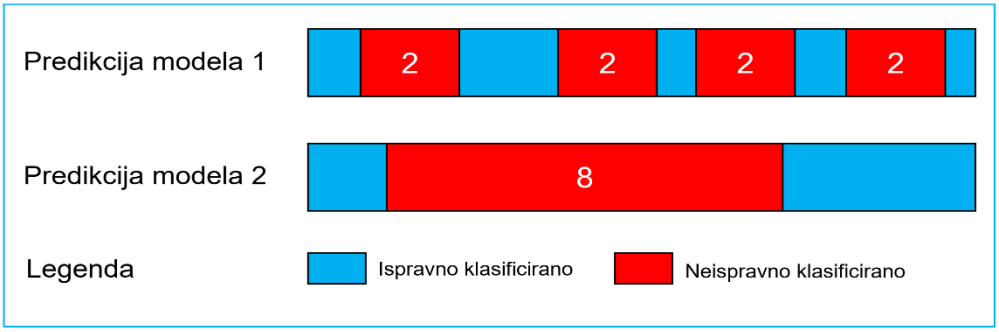
- Točnost po pojedinačnoj sličici (engl. *frame wise accuracy*).
- Segmentacijska F1 metrika uz definiran minimalan prag preklapanja [24].
- Veličina modela s aspekta broja parametara.
- Vrijeme učenja potrebno za konvergenciju.

Razlog izbora ove četiri komponente za usporedbu modela potaknut je ciljem da se pronađe balans između učinkovitosti modela i njegovih funkcionalnih karakteristika. Konkretno, u praktičnim uvjetima, a ovisno od domene primjene, često je potrebno napraviti kompromis između najtočnijih modela veće složenosti te kompaktnijih modela koji možda imaju i nižu točnost, ali su pogodniji za integraciju u postojeći tehnički sustav. U nastavku će biti detaljnije opisane četiri komponente procedure za evaluaciju te će biti uvedena metrika koja je kompozit tih komponenti.

Osnovna metrika koja se gotovo uvijek koristi u literaturi kod problema *„action segmentation“*-a je točnost po pojedinačnoj sličici. Spomenuta metrika mjeri udio točno klasificiranih sličica u video zapisu u odnosu na ukupan broj sličica prema izrazu (5.5).

$$Točnost(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{y^{(t)} = \hat{y}^{(t)}\} \tag{5.5}$$

U jednadžbi (5.5) $\mathbf{y} = \left(y^{(1)}, y^{(2)}, \ldots, y^{(\mathrm{T})}\right)$ je niz stvarnih oznaka za čije elemente vrijedi $y^{(t)} \in \{1, 2, \ldots, \mathrm{C}\}$, a $\hat{\mathbf{y}} = \left(\hat{y}^{(1)}, \hat{y}^{(2)}, \ldots, \hat{y}^{(\mathrm{T})}\right)$ je niz predikcija oznaka za čije elemente također vrijedi $\hat{y}^{(t)} \in \{1, 2, \ldots, \mathrm{C}\}$, pri čemu je $\mathbf{1}\{\cdot\}$ indikatorska funkcija koja vraća 1 ako je jednakost istinita, a 0 inače. Nedostatak ove metrike je da modeli slične točnosti mogu imati velike kvalitativne razlike u pogledu prepoznatih segmenata u video zapisu što je zorno prikazano na slici 5.30.



Slika 5.30 Usporedba izlaza dva modela s aspekta točnosti i segmentiranosti

Slika 5.30 pokazuje da će model koji je prepoznao devet segmenata i model koji je prepoznao tri segmenta, u video zapisu koji ima samo jedan segment, na temelju točnosti biti jednako kažnjeni. Opisano svojstvo metrike točnosti je problematično kod ocjenjivanja učinkovitosti modela u različitim primjenama pa tako i u slučaju studija vremena. U radu [24] predložena je segmentacijska F1@IoU metrika uz minimalan prag preklapanja između stvarnog segmenta i predikcije segmenta. Ova metrika ima tri svojstva koja je čine pogodnom za evaluaciju modela u domeni „*action segmentation*“-a. Prvo svojstvo je da kažnjava greške prekomjerne segmentiranosti, a drugo svojstvo je da ne kažnjava manje pomake u vremenu koji mogu biti posljedica varijabilnosti i subjektivnosti kod označavanja, dok je završno svojstvo da rezultati ovise o broju aktivnosti, a ne njihovom trajanju. Pseudokod algoritma za izračuna F1@IoU metrike izveden je na temelju analize izvornog koda povezanog uz rad [24] te je dan u tablici 5.4.

Tablica 5.4 Pseudokod algoritma za izračun segmentacijske F1 metrike uz minimalni prag preklapanja za jedno opažanje

Zadaj: $\mathbf{O} = (O_1, O_2, \dots, O_N)$ // stvarne oznake segmenata

Zadaj: $\mathbf{P} = (P_1, P_2, \dots, P_M)$ // predikcija oznaka segmenata

Zadaj: $\mathbf{S_O} = \left\{\left(s_o^{(1)}, e_o^{(1)}\right), \left(s_o^{(2)}, e_o^{(2)}\right), \dots, \left(s_o^{(N)}, e_o^{(N)}\right)\right\}$ // stvarne granice segmenata

Zadaj: $\mathbf{S_P} = \left\{\left(s_p^{(1)}, e_p^{(1)}\right), \left(s_p^{(2)}, e_p^{(2)}\right), \dots, \left(s_p^{(M)}, e_p^{(M)}\right)\right\}$ // predikcija granica segmenata

Zadaj: $prag$ // minimalni prag preklapanja između predikcije i stvarnog segmenta

Inicijaliziraj: $TP = 0, FP = 0, FN = 0$

Inicijaliziraj: $\boldsymbol{n_O} = \mathbf{0}_{N,1}$ // indikator da li je stvarni segment već uparen s nekom od predikcija, nul vektor dimenzije jednake broju stvarnih segmenata,

Za $i = 1, 2, \dots, M$

  Inicijaliziraj: $\boldsymbol{IoU} = \mathbf{0}_{N,1}$ // spremnik IoU-a za P_i segment u odnosu na sve stvarne segmente

  Za $j = 1, 2, \dots, N$

    Ako $P_i = O_j$ **onda** // samo u slučaju da su predikcija i oznaka iste računaj IoU

      $$IoU_j = \frac{\mathrm{S_P^{(i)}} \cap \mathrm{S_O^{(j)}}}{\mathrm{S_P^{(i)}} \cup \mathrm{S_O^{(j)}}}$$

    Kraj

  Kraj

  $idx = \underset{j}{\operatorname{argmax}}\ \boldsymbol{IoU}$ // vrati indeks elementa koji ima najveći IoU

  Ako $IoU_{idx} \geq prag$ **i** $n_{O\,idx} = 0$ **onda** // ako IoU trenutne predikcije segmenta zadovoljava prag, te stvarni segment s kojim trenutna predikcija ima najveći preklop nije već zauzet, ažuriraj TP

    $TP \leftarrow TP + 1$

    $n_{O\,idx} = 1$

  Inače

    $FP \leftarrow FP + 1$

  Kraj

Kraj

$FN = N - \sum_j^N n_{O\,j}$ // svaki neupareni stvarni segment je FN

$$Preciznost = \frac{TP}{TP + FP}$$

$$Odziv = \frac{TP}{TP + FN}$$

$$F1@IoU = 2\frac{Preciznost \cdot Odziv}{Preciznost + Odziv}$$

Vrati $F1@IoU$

Iz opisa algoritma slijedi da metrika zahtjeva kao ulaz granice segmenta, $\mathbf{S_O}$ i $\mathbf{S_P}$, i njihove oznake **O** i **P**, te minimalnu vrijednost preklapanja između stvarnog segmenta i predikcije segmenta. Preklapanje se računa na temelju omjera između presjeka i unije stvarnog segmenta i predikcije segmenta (engl. *Intersection over Union–IoU*) što je pokazano u 7. koraku algoritma. U literaturi je najčešće korišten prag od minimalno 50% preklapanja, stoga je i u disertaciji korištena ova vrijednost. Kako u općem slučaju broj stvarnih segmenata N i predikcije segmenata M može biti različit, definirano je da stvarni segment može biti uparen samo s jednom predikcijom segmenta (2. korak), pri čemu predikcija mora imati istu oznaku kao i stvarni segment (6. korak). U slučaju da predikcija segmenta zadovoljava minimalni preklop sa stvarnim segmentom, te taj stvarni segment već nije uparen s nekom drugom predikcijom (11. korak), predikcija će biti proglašena stvarno pozitivnom (engl. *true positive–TP*), inače će predikcija biti proglašena lažno pozitivnom (engl. *false positive–FP*). U slučaju da neki stvarni segment nije uparen ni s jednom predikcijom segmenta, takav stvarni segment bit će uračunat kao lažno negativan (engl. *false negative–FN*). Na temelju broja stvarno pozitivnih i lažno pozitivnih, računa se preciznost (engl. *precision*) modela što je pokazano u 19. koraku algoritma. Preciznost predstavlja udio ispravno pozitivno klasificiranih segmenata u skupu svih koji su pozitivno klasificirani. U sljedećem koraku algoritma izračunat je odziv (engl. *recall*) modela koji predstavlja udio ispravno pozitivno klasificiranih segmenata u skupu svih koji su stvarno pozitivni. U posljednjem koraku algoritma definirana je F1@IoU metrika koja je dobivena kao harmonijska sredina (engl. *harmonic mean*) preciznosti i odziva. Razlog korištenja harmonijske sredine, umjesto aritmetičke sredine, leži u tome da harmonijska sredina daje veću težinu nižim vrijednostima stoga će model imati visoku F1 metriku samo kada su i preciznost i odziv visoki. Svojstva F1@IoU metrike te njene razlike u odnosu na točnost moguće je pokazati na primjerima. Za stvarni niz $\mathbf{y} = (0, 0, 0, 0, 1, 1, 1, 1)$ koji se sastoji od dva segmenta trajanja četiri vremenske jedinice uz povezanu predikciju $\hat{\mathbf{y}} = (0, 1, 0, 0, 1, 1, 1, 1)$, vrijednost F1 metrike uz prag od 50% bit će 66,6% dok će metrika točnosti biti 87,5%. Ovaj primjer ilustrira da je F1 stroža metrika od točnosti u slučaju prekomjerne segmentiranosti. Razlog niske vrijednosti metrike posljedica je činjenice da je model generirao dva segmenta s oznakom nula i dva segmenta s oznakom jedan. Za stvarni niz $\mathbf{y} = (0, 0, 1, 1, 2, 2, 2, 2)$ koji se sastoji od tri segmenta pri čemu prva dva traju dvije vremenske jedinice, a posljednji četiri vremenske jedinice uz povezanu predikciju $\hat{\mathbf{y}} = (0, 1, 1, 1, 2, 2, 2, 2)$, vrijednost F1 metrike uz prag od 50% bit će 100% dok će točnost biti 87,5%. Iz prethodnog primjera je vidljivo da F1 ne penalizira male vremenske pomake. Nadalje, moguće je zaključiti da su metrika točnosti i F1 komplementarne jer kažnjavaju različite greške modela što je i razlog zašto će se u razvijenoj

proceduri koristiti obje metrike. U okviru disertacije razmatrana je i metrika srednje prosječne preciznosti[27] (engl. *mean average precision*–mAP) koja je puno češće korištena u literaturi iz domene „*action detection*"-a. Spomenuta metrika slična je F1@IoU metrici na način da isto koristi minimalan prag preklapanja između predikcije i stvarnih segmenata, dok se razlikuje po tome da mAP koristi informaciju o vjerojatnosti oznake segmenta kako bi se kreirala rangirana lista predikcija. Rezultati mAP metrike značajno ovise o načinu na koji je izračunata vjerojatnost segmenta [24], pri čemu se najčešće koristi ili prosječna vrijednost vjerojatnosti oznaka sličica u segmentu ili maksimalna vrijednost vjerojatnosti oznake u segmentu. U eksperimentima u okviru ovog istraživanja pokazalo se da mAP metrika nije dovoljno dobra za finu detekciju aktivnosti te daje rezultate slične metrici točnosti, stoga je isključena iz procedure.

Na temelju inicijalnih eksperimenata kao funkcionalne metrike modela izabrane su vrijeme učenja i broj parametara. Broj parametara modela obuhvaća sve elemente modela koji su podložni adaptaciji u procesu učenja. Vrijeme učenja je produkt broja epoha i vremena izvođenja jedne epohe. Izabrane funkcionalne metrike su u djelomičnoj pozitivnoj korelaciji, što se očituje u tome da se s rastom broja parametara produljuje vrijeme trajanja pojedinačne epohe. Međutim, opisana veza između broja parametara i vremena učenja nije jednolična uslijed primjene različitih arhitektura koje obrađuju ulazne podatke na različit način. Također, usporedbom vremena učenja modela s jednakim brojem parametara, ali različite dubine, uočena su dodatna odstupanja od prethodno spomenute veze između funkcionalnih metrika. Finalno, ove razlike su dodatno potencirane činjenicom da vrijeme konvergencije algoritma ne ovisi samo o broju parametara već o složenim interakcijama između modela, funkcije cilja i optimizacijske metode. Navedeni argumenti su razlog zašto su obje metrike dio procedure za evaluaciju modela.

U tablici 5.5 prikazani su rezultati 27 najboljih modela iz svake grupe pristupa po pitanju prethodno opisana četiri elementa procedure. Odabir najboljeg rješenja na temelju različitih kriterija je problem iz domene višekriterijske optimizacije. U ovom radu definirana je jednostavna procedura koja rezultira agregiranim pokazateljem na temelju kojeg je moguća brza usporedba ukupne učinkovitosti različitih modela.

[27] Pogledati modul „*metrics*" u biblioteci „*phd_lib*"

Tablica 5.5 Rezultati evaluacije najboljih modela iz svake od 27 grupa

Oznaka modela	Arhitektura	Pristup izvlačenju značajki	Kadar	Broj epoha	Trajanje epohe [s]	Vrijeme učenja [s]	Broj parametara	Točnost	F1@50
BFEHE	biLSTM	FE	HE	40	11	440	25.194.506	90,38	76,14
CFEHE	CONV	FE	HE	160	5	800	627.860	95,66	86,60
LFEHE	LSTM	FE	HE	80	7	560	2.362.890	87,82	68,19
BFEF	biLSTM	FE	Fokus	80	3	240	264.650	96,33	88,30
CFEF	CONV	FE	Fokus	160	3	480	380.500	96,97	90,49
LFEF	LSTM	FE	Fokus	80	3	240	266.698	95,63	84,94
BFEC	biLSTM	FE	Concat	40	6	240	1.057.674	96,18	85,59
CFEC	CONV	FE	Concat	80	5	400	511.572	97,00	90,34
LFEC	LSTM	FE	Concat	120	5	600	528.842	95,55	84,73
BTLHE	biLSTM	TL	HE	10	2	20	131.818	95,53	84,48
CTLHE	CONV	TL	HE	20	3	60	380.500	96,75	89,85
LTLHE	LSTM	TL	HE	40	2	80	132.330	96,21	86,76
BTLF	biLSTM	TL	Fokus	40	2	80	131.818	97,52	92,45
CTLF	CONV	TL	Fokus	10	3	30	380.500	97,50	92,63
LTLF	LSTM	TL	Fokus	40	2	80	132.330	97,29	91,17
BTLC	biLSTM	TL	Concat	20	3	60	262.890	97,78	93,16
CTLC	CONV	TL	Concat	10	5	50	511.572	97,74	93,46
LTLC	LSTM	TL	Concat	20	3	60	263.402	97,56	91,15
BTBHE	biLSTM	TB	HE	160	2	320	135.018	97,37	92,10
CTBHE	CONV	TB	HE	40	3	120	380.500	97,60	92,48
LTBHE	LSTM	TB	HE	80	2	160	132.330	97,01	88,70
BTBF	biLSTM	TB	Fokus	40	2	80	131.818	97,24	90,39
CTBF	CONV	TB	Fokus	40	3	120	380.500	97,41	92,33
LTBF	LSTM	TB	Fokus	40	2	80	132.330	96,94	88,89
BTBC	biLSTM	TB	Concat	20	3	60	262.890	97,66	91,80
CTBC	CONV	TB	Concat	40	5	200	511.572	97,84	93,64
LTBC	LSTM	TB	Concat	20	3	60	263.402	97,47	90,75

Izračun jedinstvenog pokazatelja učinkovitosti modela sastoji se od nekoliko koraka. U prvom koraku sva četiri elementa procedure svedena su u isti raspon vrijednosti. Razlog za provedbu ovog koraka je uklanjanje utjecaja različitih mjernih jedinica na važnost pojedine komponente. Ovo je najlakše uočiti na primjeru usporedbe raspona vrijednosti broja parametara i raspona vrijednosti preostalih elemenata. Bez odgovarajućeg skaliranja, preostale tri metrike imale bi marginalan utjecaj na jedinstveni pokazatelj. Razmatrani su različiti pristupi za provedbu ovog koraka poput npr. normalizacije[28], ali je konačno odabrana solucija kojom se rezultati svih tipova modela na određenoj metrici dovode u interval $[0, 1]$. Drugim riječima, ako se svi rezultati ostvareni na određenoj metrici skupe u vektor $\boldsymbol{x}$, onda će na svaku komponentu navedenog vektora biti primijenjen izraz (5.6), u kojem su $min(\cdot)$ i $max(\cdot)$ funkcije koje vraćaju najmanju i najveću vrijednost vektora.

$$x_i^{std} = \frac{x_i - min(\boldsymbol{x})}{max(\boldsymbol{x}) - min(\boldsymbol{x})} \tag{5.6}$$

Dva su razloga utjecala na ovaj odabir. Prvi razlog je povezan s time što u općem slučaju odabrane funkcionalne metrike nemaju referentnu donju i gornju granicu raspona vrijednosti, kao što je imaju točnost i F1@IoU metrika pa je interval $[0, 1]$ odabran kao neutralno rješenje. Drugi razlog odabira postupka skaliranja u raspon $[0, 1]$ je taj da će njime biti osigurana separacija između modela s obzirom da će sigurno postojati rezultati na donjoj i gornjoj granici raspona vrijednosti metrike. Drugi korak koji je proveden s ciljem razvoja jedinstvenog pokazatelja bio je reflektiranje vrijednosti funkcionalnih metrika. Konkretno, ako se svi skalirani rezultati pojedinih modela na odabranoj metrici naslažu u vektor $\boldsymbol{x}^{std}$, onda će na svaku komponentu biti primijenjen izraz $x_i^r = 1 - x_i^{std}$. Ova manipulacija napravljena je s ciljem ujednačavanja interpretacije svih metrika. Preciznije, na ovaj način veće vrijednosti metrike uvijek su povezane s boljom učinkovitosti modela. S obzirom da su vrijednosti funkcionalne metrike bile u intervalu $[0, 1]$, nakon reflektiranja moguće ih je tumačiti kao inverz broja parametara i vremena učenja, stoga će se u nastavku teksta koristiti nazivi „*kompaktnost*“ modela i „*brzina učenja*“, u smislu da su modeli s većom vrijednosti funkcionalne metrike kompaktniji („manji“) i brže konvergiraju. Sljedeći korak definiranja jedinstvenog pokazatelja je odabir veze između četiri odabrane metrike. Za ovaj korak odlučeno je da će se koristiti težinski prosjek. Formalno, ako se rezultat pojedine metrike, koji je pripremljen na temelju dva prethodno opisana koraka, označi s M_i, a važnost pojedine metrike

[28] Normalizirani podatci su oni koji imaju aritmetičku sredinu jednaku nuli i jediničnu standardnu devijaciju

je određena koeficijentom w_i, onda jednadžba (5.7) predstavlja težinski prosjek metrika. U kontekstu evaluacije modela, opisani težinski prosjek interpretira se kao jedinstveni pokazatelj koji je nazvan „*standardizirana učinkovitost*“ – STU.

$$STU = \frac{\sum_i w_i M_i}{\sum_i w_i} \tag{5.7}$$

Kombiniranje metrika na temelju težinskog prosjeka motivirano je ciljem da razvijeni pokazatelj bude prilagodljiv zahtjevima domene u kojoj će model biti korišten. Zbog toga što su sve komponente STU pokazatelja skalirane u raspon $[0, 1]$ i vrijednosti STU pokazatelja bit će u istom rasponu.

Razvijena procedura i STU pokazatelj iskorišteni su da se napravi analiza učinkovitosti modela. Analiza je organizirana u dva dijela. U prvom dijelu zasebno su analizirani modeli naučeni na podatcima s istog kadra pri čemu je na podatke primijenjen isti pristup izvlačenja značajki. Cilj je bio utvrditi koja je arhitektura najpogodnija za određenu vrstu ulaznih podataka. U drugom dijelu analize izračunate su standardizirane vrijednosti četiri komponente metrike za sve modele zajedno te je na temelju njih dobiven STU pokazatelj koji je u ovom kontekstu funkcija učinkovitosti modela, ali i korištenih značajki. Svrha ovog dijela analize bila je prepoznati optimalnu kombinaciju ulaznih podataka i modela.

5.4.1 Analiza učinkovitosti modela razvijenih na istim ulaznim značajkama

Ovaj dio obuhvaća devet grupa modela za koje su u analizi dani:

a) *Vrijednosti STU pokazatelja*. STU pokazatelj za svaki model u grupi izračunat je na tri različita načina. Prvi način tretira sve četiri metrike kao jednako važne, tj. svakoj komponenti pridodan je jednak koeficijent težine w_i iz čega slijedi da je za ovaj slučaj STU pokazatelj ekvivalentan aritmetičkoj sredini četiri metrike svakog modela. Kod druga dva načina izračuna STU pokazatelja, ispitani su scenariji kada je dva puta veća težina stavljena na metrike učinkovitosti te kada je dva puta veća težina stavljena na funkcionalne metrike.
b) *Grafovi standardiziranih vrijednosti četiri elementa procedure za evaluaciju modela*. U ovom dijelu analize četiri vrste metrike su skalirane u interval $[0, 1]$ na temelju rezultata tri vrste modela unutar pojedine grupe, nakon čega je napravljeno reflektiranje funkcionalnih metrika kako bi se dobile standardizirane vrijednosti kompaktnosti i brzine učenja.

c) *Tablice optimalnih hiperparametara.* Prethodno su u tablicama 5.2 i 5.3 pokazani hiperparametri LSTM, biLSTM i konvolucijskih modela koji su korišteni kod eksperimenata. Dio optimalnih hiperparametara je isti kod svih 27 modela stoga ti hiperparametri neće biti prikazani u tablicama koje slijede u nastavku teksta. Radi se o sljedećim hiperparametrima kod LSTM i biLSTM modela:

- Broj dodatnih potpuno povezanih slojeva (PS)
- Broj neurona u dodatnim potpuno povezanim slojevima (PN)
- Vjerojatnost izbacivanja kod dropout sloja (VDR)

Rezultati eksperimenata su pokazali da dodatni potpuno povezani slojevi ne doprinose boljoj učinkovitosti LSTM i biLSTM modela već negativno utječu na brzinu učenja i kompaktnost te brzo dovode modele u područje prenaučenosti. U slučaju konvolucijskih modela hiperparametri koji su isti kod svih modela su:

- Broj etapa rafiniranja (ER)
- Dijeljenje težina između etapa rafiniranja (DT)
- Broj izlaza iz modela (BI)

Dijeljenje težina između etapa rafiniranja je relevantno samo u slučaju da model koristi veći broj etapa, dok najbolji konvolucijski modeli razvijeni u sklopu disertacije imaju samo jednu etapu rafiniranja. Eksperimenti su pokazali da veći broj izlaza iz konvolucijskih modela pozitivno utječe na brzinu učenja, stoga je kod svih modela korištena ova opcija. Sva tri tipa arhitektura postižu bolje rezultate kada se kao optimizacijska metoda koristi stohastički gradijentni spust u kombinaciji s momentom umjesto ADAM metode. Ovaj fenomen je potencijalno povezan s činjenicom da je uz SGD metodu najčešće korišten i ciklički raspored upravljanja stopom učenja koji nije bio učinkovit u kombinaciji s ADAM metodom. Iako su u eksperimentima korištene tri veličine mini grupa opažanja (VG), na kraju se pokazalo da je najbolje koristiti maksimalnu veličinu grupe koja može biti obrađena grafičkim procesorom. Za značajke razvijene na pojedinačnim kadrovima (HE i Fokus) korištena je veličina grupe od 32 opažanja, a za fuziju oba kadra (Concat) to je bila grupa s 16 opažanja. Završno, kao funkcija gubitka kod svih modela korištena je unakrsna entropija. Istražen je i utjecaj kombiniranja unakrsne entropije sa segmentacijskim gubitkom, ali rezultati nisu opravdali primjenu ovog dodatka. Mogući razlog zašto segmentacijski gubitak nije doveo do boljih rezultata je utjecaj dva dodatna hiperparametra samog gubitka koji nisu bili adekvatno prilagođeni ulaznim podatcima.

d) *Segmentacijski grafovi za opažanja na kojima su ostvareni najgori i najbolji rezultati po pitanju u F1@IoU metrike*. Svrha ovih grafova je da ukažu na kvalitativne karakteristike modela te da se prepoznaju aktivnosti kod kojih modeli najčešće griješe. Boje korištene na grafu služe za razlikovanje 10 aktivnosti iz snimanog procesa.

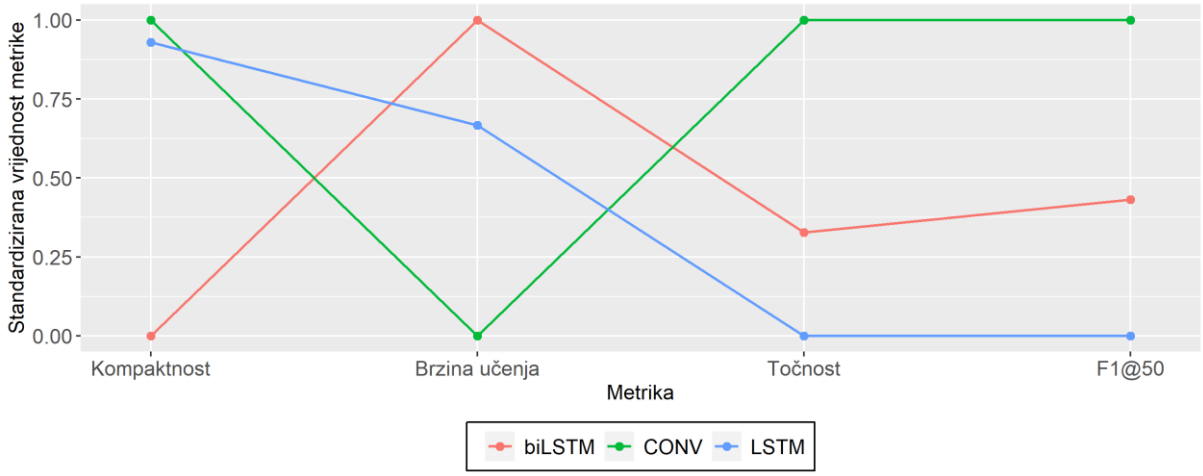
Analiza modela za kadar HE i pristup izvlačenja značajki FE

Usporedba modela razvijenih na značajkama dobivenima FE pristupom, iz podataka prikupljenih u HE kadru snimanja, na temelju STU pokazatelja izračunatog s tri različite konfiguracije težinskih koeficijenata dana je u tablici 5.6.

Tablica 5.6 Tri načina izračuna STU pokazatelja modela za kadar HE i pristup FE

Arhitektura	STU		
	Težište na učinkovitosti	Ujednačena važnost svih metrika	Težište na funkcionalnosti
biLSTM	0,4196	0,4397	0,4598
CONV	0,8333	0,7500	0,6666
LSTM	0,2660	0,3990	0,5320

Zaključak 1: CONV model je najbolji neovisno od toga na čemu je težište STU pokazatelja. S druge strane kada je težište na funkcionalnim metrikama, LSTM model je bolji od biLSTM modela, dok je u slučaju težišta na točnost i F1 metrici bolji biLSTM model.



Slika 5.31 Standardizirane vrijednosti 4 metrike kod modela za kadar HE i pristup FE

Graf sa slike 5.31 sugerira da je CONV model najbolji po pitanju tri elementa za ovu vrstu ulaznih podataka, pri čemu je njegova slabost duže vrijeme učenja u odnosu na konkurentske modele.

Zaključak 2: Razlozi zašto CONV model ima najbolju vrijednost STU pokazatelja su ti da je po pitanju apsolutne vrijednosti točnosti za 5 bodova bolji od sljedećeg modela, a po pitanju F1 metrike je za 10 bodova bolji od sljedećeg modela, te ima 40 puta manje parametara u odnosu na najveći model iz ove grupe (vidi tablicu 5.5).

Kako je na grafu sa slike 5.31 uočeno da LSTM ima bolju kompaktnost od biLSTM modela, dok biLSTM model brže konvergira, dodatno je analizirano zašto STU pokazatelj s težištem na funkcionalnim metrikama preferira LSTM model.

Zaključak 3: LSTM model je bolji od biLSTM modela bez obzira što je sporiji od njega kod konvergencije, jer ima puno bolju kompaktnost uslijed 12 puta manjeg broja parametara (vidi tablicu 5.5).

U nastavku su prikazane optimalne vrijednosti ključnih hiperparametara za sva tri tipa modela.

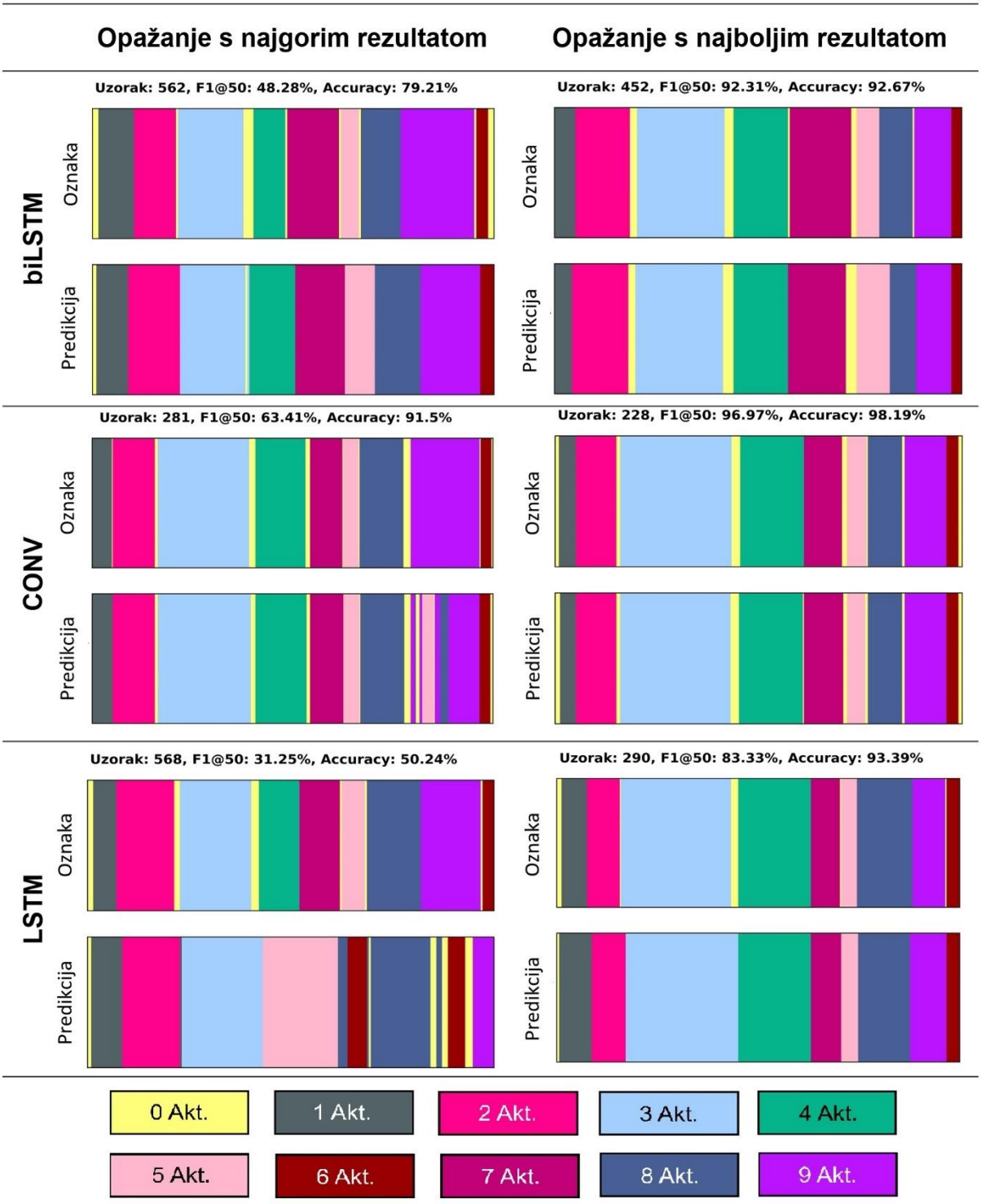
Tablica 5.7 Optimalni hiperparametri LSTM i biLSTM modela za kadar HE i pristup FE

Arhitektura	**LS**	**LN**	**SU**	**RSU**	**REG**
LSTM	1	256	$5 \cdot 10^{-2}$	Bez rasporeda	$5 \cdot 10^{-4}$
biLSTM	1	1024	$5 \cdot 10^{-4}$	Linearno smanjenje	0

Tablica 5.8 Optimalni hiperparametri konvolucijskog modela za kadar HE i pristup FE

Arhitektura	**DRS**	**DDS**	**BJ**	**VDR**	**SU**	**RSU**
CONV	10	10	64	0,2	$1 \cdot 10^{-3}$	Bez rasporeda

Na slici 5.32 prikazan je segmentacijski graf za opažanja na kojima su modeli iz ove grupe ostvarili najbolje i najgore rezultate.



Slika 5.32 Segmentacijski graf za opažanja na kojima su ostvareni najgori i najbolji rezultati modela za kadar HE i pristup FE

Zaključak 4: biLSTM model griješi kod prijelaza između aktivnosti, jer tada najčešće dolazi do pojave pozadinske klase aktivnosti, s kojom imaju problema svi tipovi modela. Opažanje 562 izvodi operater O4 koji na specifičan način radi 5. aktivnost (puno brže, vidi odjeljak 4.2.5), pa model kasni u prepoznavanju početka 8. aktivnosti. biLSTM model kasni i kod prijelaza iz 8. u

9. aktivnost. CONV model je imao najvećih problema na opažanju 281 za operatera O3, gdje je 9. aktivnost pogrešno prepoznao kao aktivnost 5 ili 8. Razlozi za ovo su da je operater prvo imao problema s uzimanjem kopče iz kutije što je vizualno nalikovalo na početak 5. aktivnosti, a zatim je spustio ruke na poprečnu stranicu što izgleda kao početak 8. aktivnosti. LSTM je imao najvećih problema na opažanju 568 kojeg izvodi O4, iako kod ovog opažanja nema nikakvih specifičnosti osim prijelaza iz 5. u 8. aktivnost. Najvjerojatniji razlog za ovaj rezultat je što se radi o najslabijem modelu od svih 27 završnih modela.

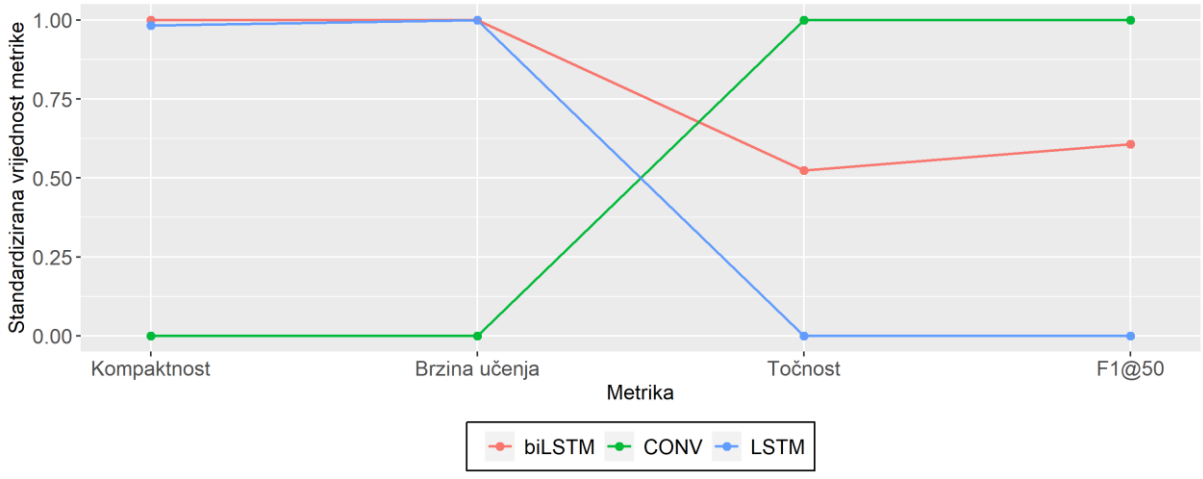
Analiza modela za kadar Fokus i pristup izvlačenja značajki FE

Usporedba modela razvijenih na značajkama dobivenima FE pristupom, iz podataka prikupljenih u Fokus kadru snimanja, na temelju STU pokazatelja izračunatog s tri različite konfiguracije težinskih koeficijenata dana je u tablici 5.9.

Tablica 5.9 Tri načina izračuna STU pokazatelja modela za kadar Fokus i pristup FE

Arhitektura	STU		
	Težište na učinkovitosti	Ujednačena važnost svih metrika	Težište na funkcionalnosti
biLSTM	0,7103	0,7827	0,8552
CONV	0,6666	0,5000	0,3333
LSTM	0,3308	0,4956	0,6608

Zaključak 1: biLSTM model je najbolji neovisno od toga na čemu je težište STU pokazatelja. S druge strane kada je težište na funkcionalnim metrikama, LSTM model je bolji od CONV modela, dok je u slučaju težišta na točnosti i F1 metrici bolji CONV model.



Slika 5.33 Standardizirane vrijednosti 4 metrike kod modela za kadar Fokus i pristup FE

Graf sa slike 5.33 sugerira da je biLSTM model najbolji po pitanju funkcionalnih metrika za ovu vrstu ulaznih podataka, pri čemu je nešto slabiji od CONV modela s aspekta točnosti i F1 metrike.

Zaključak 2: Razlozi zašto biLSTM model ima najbolju vrijednost STU pokazatelja su ti da mu je potrebno 2 puta manje epoha do konvergencije te ima 1,4 puta manje parametara od CONV modela, dok mu je točnost samo za 0,5 bodova manja, a F1 metrika za 2 boda manja od tog istog modela (vidi tablicu 5.5).

U nastavku su prikazane optimalne vrijednosti ključnih hiperparametara za sva tri tipa modela.

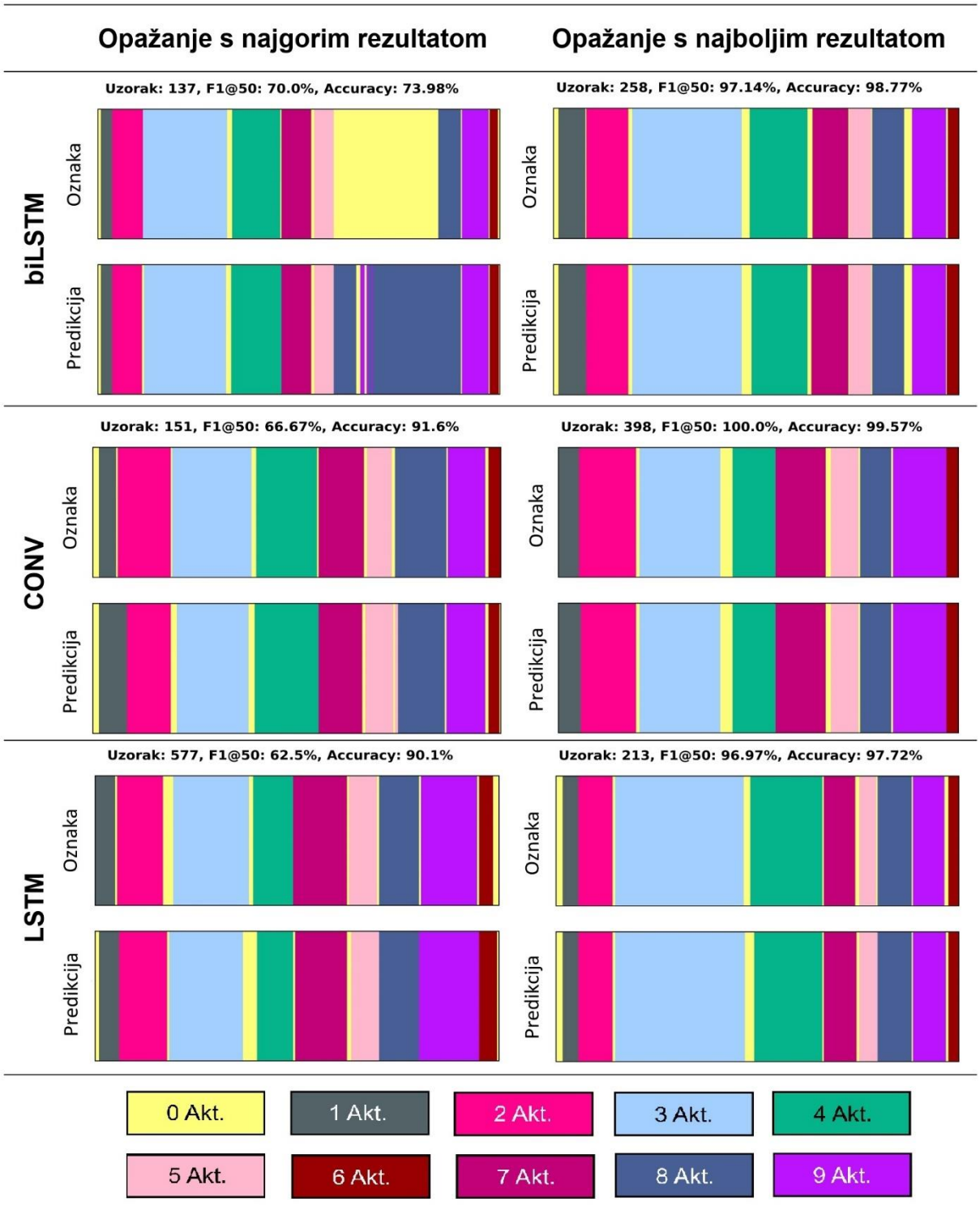
Tablica 5.10 Optimalni hiperparametri LSTM i biLSTM modela za kadar Fokus i pristup FE

Arhitektura	**LS**	**LN**	**SU**	**RSU**	**REG**
LSTM	1	32	$[3 \cdot 10^{-3}, 1 \cdot 10^{-1}]$	ciklički	0
biLSTM	1	16	$[2 \cdot 10^{-3}, 2 \cdot 10^{-1}]$	ciklički	0

Tablica 5.11 Optimalni hiperparametri konvolucijskog modela za kadar Fokus i pristup FE

Arhitektura	**DRS**	**DDS**	**BJ**	**VDR**	**SU**	**RSU**
CONV	5	5	64	0,2	$[1 \cdot 10^{-4}, 3 \cdot 10^{-3}]$	ciklički

Na slici 5.34 prikazan je segmentacijski graf za opažanja na kojima su modeli iz ove grupe ostvarili najbolje i najgore rezultate.



Slika 5.34 Segmentacijski graf za opažanja na kojima su ostvareni najgori i najbolji rezultati modela za kadar Fokus i pristup FE

Zaključak 3: biLSTM model najviše griješi na opažanju 137 koje izvodi operater O1 na način da pozadinske aktivnosti prepoznaje kao 8. i 9. aktivnost. Ove aktivnosti se odnose na umetanje kopči na lijevoj gornjoj i donjoj strani rešetke. Razlog ove greške je taj što operater ima

problema s odvajanjem nekoliko kopči koje su međusobno zaglavile, pri čemu pomiče ruke na pozicije koje sugeriraju da će početi s 8. ili 9. aktivnosti. Ovo opažanje je problematično kod većine modela. CONV i LSTM modeli griješe kod prijelaza između aktivnosti što je povezano s pozadinskim aktivnostima koje se u Fokus kadru najčešće ne mogu vidjeti.

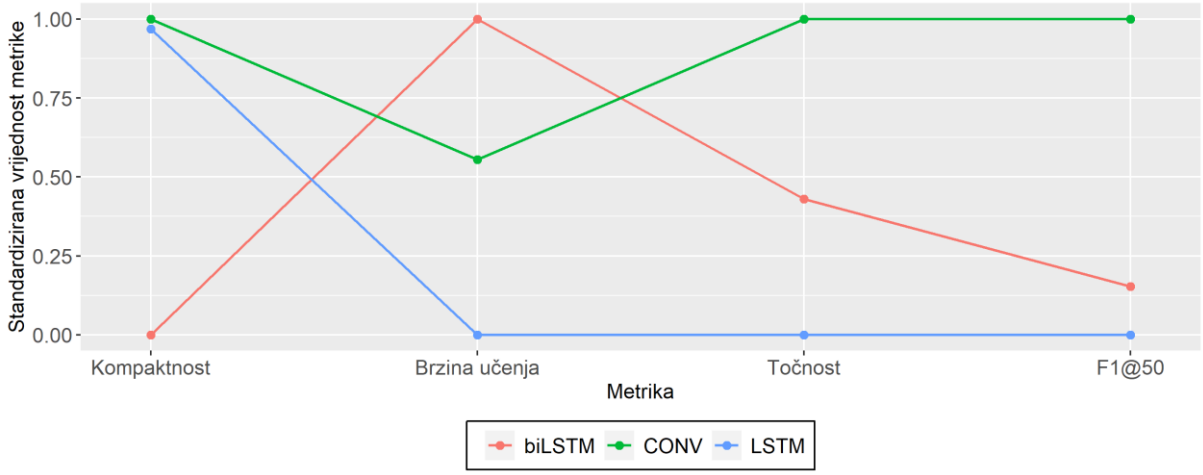
Analiza modela za kadar Concat i pristup izvlačenja značajki FE

Usporedba modela razvijenih na značajkama dobivenima FE pristupom, iz podataka prikupljenih u Concat kadru snimanja, na temelju STU pokazatelja izračunatog s tri različite konfiguracije težinskih koeficijenata dana je u tablici 5.12.

Tablica 5.12 Tri načina izračuna STU pokazatelja modela za kadar Concat i pristup FE

Arhitektura	STU		
	Težište na učinkovitosti	Ujednačena važnost svih metrika	Težište na funkcionalnosti
biLSTM	0,3611	0,3958	0,4306
CONV	0,9259	0,8888	0,8518
LSTM	0,1614	0,2421	0,3228

Zaključak 1: CONV model je najbolji neovisno od toga na čemu je težište STU pokazatelja, dok je biLSTM uvijek bolji izbor od LSTM modela za ovu vrstu ulaznih podataka.



Slika 5.35 Standardizirane vrijednosti 4 metrike kod modela za kadar Concat i pristup FE

Graf sa slike 5.35 sugerira da je CONV model najbolji po pitanju tri elementa za ovu vrstu ulaznih podataka, pri čemu je njegova slabost nešto sporije učenje u odnosu na biLSTM model.

Zaključak 2: Razlozi zašto CONV model ima najbolju vrijednost STU pokazatelja su ti da je po pitanju apsolutne vrijednosti točnosti za 1 bod bolji od sljedećeg modela, a po pitanju F1

metrike je za 5 bodova bolji od sljedećeg modela, te ima 2 puta manje parametara u odnosu na biLSTM model (vidi tablicu 5.5).

Kako je na grafu sa slike 5.35 uočeno da LSTM ima bolju kompaktnost od biLSTM modela, dok biLSTM model brže konvergira, dodatno je analizirano zašto STU pokazatelja s težištem na funkcionalnim metrikama preferira biLSTM model.

Zaključak 3: biLSTM model je bolji od LSTM modela bez obzira što ima 2 puta veći broj parametara, jer je 2,5 puta brži (vidi tablicu 5.5).

U nastavku su prikazane optimalne vrijednosti ključnih hiperparametara za sva tri tipa modela.

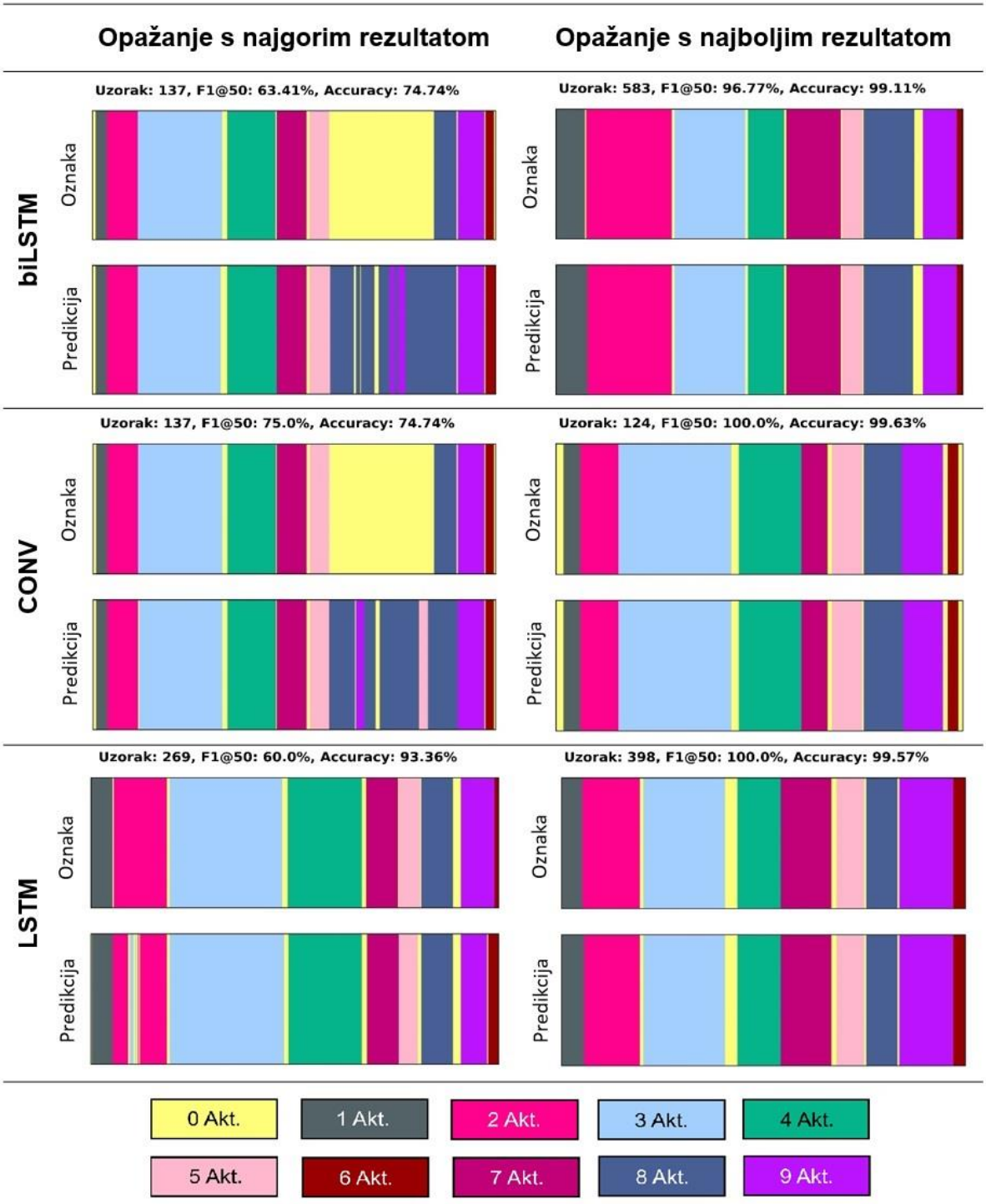
Tablica 5.13 Optimalni hiperparametri LSTM i biLSTM modela za kadar Concat i pristup FE

Arhitektura	**LS**	**LN**	**SU**	**RSU**	**REG**
LSTM	1	32	$[2 \cdot 10^{-3}, 4 \cdot 10^{-2}]$	ciklički	0
biLSTM	1	32	$[8 \cdot 10^{-3}, 1 \cdot 10^{-1}]$	ciklički	0

Tablica 5.14 Optimalni hiperparametri konvolucijskog modela za kadar Concat i pristup FE

Arhitektura	**DRS**	**DDS**	**BJ**	**VDR**	**SU**	**RSU**
CONV	5	5	64	0,2	$[9 \cdot 10^{-5}, 2 \cdot 10^{-3}]$	ciklički

Na slici 5.36 prikazan je segmentacijski graf za opažanja na kojima su modeli iz ove grupe ostvarili najbolje i najgore rezultate.



Slika 5.36 Segmentacijski graf za opažanja na kojima su ostvareni najgori i najbolji rezultati modela za kadar Concat i pristup FE

Zaključak 3: Prilikom korištenja podataka s oba kadra biLSTM i CONV modeli najviše griješe na opažanju 137, pri čemu su razlozi pojave grešaka povezani s ovim opažanjem već ranije opisani. LSTM model ima problema s opažanjem 269 kojeg izvodi operater O3, gdje model za vrijeme trajanja 2. aktivnosti pogrešno prepoznaje 3. aktivnost i pozadinske aktivnosti. Analiza

ovog opažanja ukazuje da su razlozi koji dovode do zabune od strane modela slični kao i kod opažanja 137. Konkretno, prilikom uzimanja kopče iz kutije dolazi do zaglavljivanja nekoliko kopči koje je potrebno razdvojiti prije umetanja u rešetku. Ovo opažanje je također problematično kod većine modela.

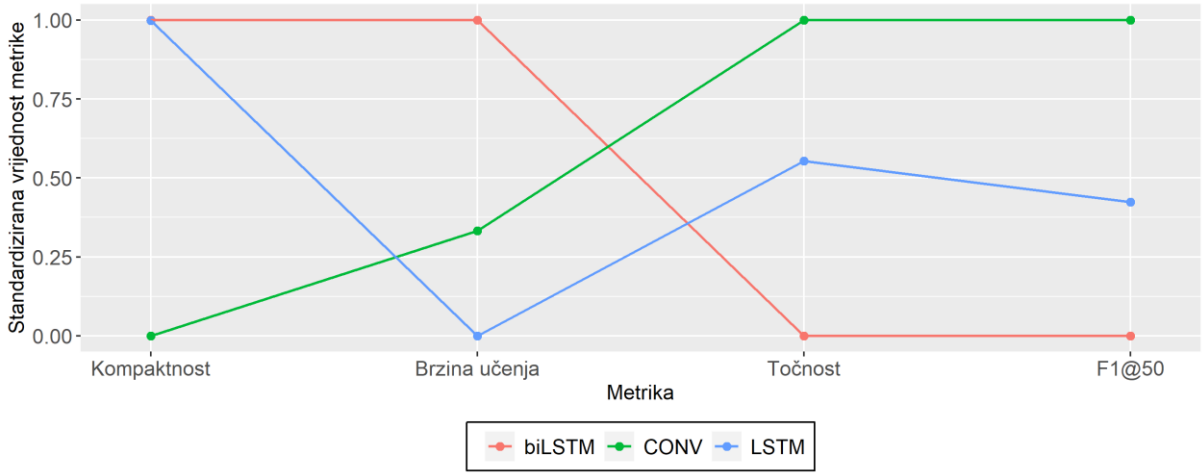
Analiza modela za kadar HE i pristup izvlačenja značajki TL

Usporedba modela razvijenih na značajkama dobivenima TL pristupom, iz podataka prikupljenih u HE kadru snimanja, na temelju STU pokazatelja izračunatog s tri različite konfiguracije težinskih koeficijenata dana je u tablici 5.15.

Tablica 5.15 Tri načina izračuna STU pokazatelja modela za kadar HE i pristup TL

Arhitektura	STU		
	Težište na učinkovitosti	Ujednačena važnost svih metrika	Težište na funkcionalnosti
biLSTM	0,3333	0,5000	0,6666
CONV	0,7222	0,5833	0,4444
LSTM	0,4921	0,4938	0,4955

Zaključak 1: CONV model je najbolji kada je težište na točnosti i F1 metrici te ga u ovom slučaju slijedi LSTM model, dok su biLSTM i LSTM bolji kada je težište na funkcionalnosti. U slučaju da su sve komponente STU pokazatelja podjednako važne CONV model je najbolji.



Slika 5.37 Standardizirane vrijednosti 4 metrike kod modela za kadar HE i pristup TL

Graf sa slike 5.37 sugerira da je CONV model najbolji po pitanju metrika učinkovitosti, dok je biLSTM bolji s aspekta funkcionalnih metrika za ovu vrstu ulaznih podataka.

Zaključak 2: Razlozi zašto CONV model ima najbolju vrijednost STU pokazatelja kada je težište na točnosti i F1 metrici su ti da je po pitanju apsolutne vrijednosti točnosti za 1 bod bolji

od biLSTM modela, a po pitanju F1 metrike je za 2,5 boda bolji od biLSTM modela. S druge strane biLSTM je bolji kada STU pokazatelj preferira funkcionalne metrike zbog toga jer je 3 puta brži od sljedećeg modela te ima skoro 3 puta manje parametara od CONV modela (vidi tablicu 5.5).

U nastavku su prikazane optimalne vrijednosti ključnih hiperparametara za sva tri tipa modela.

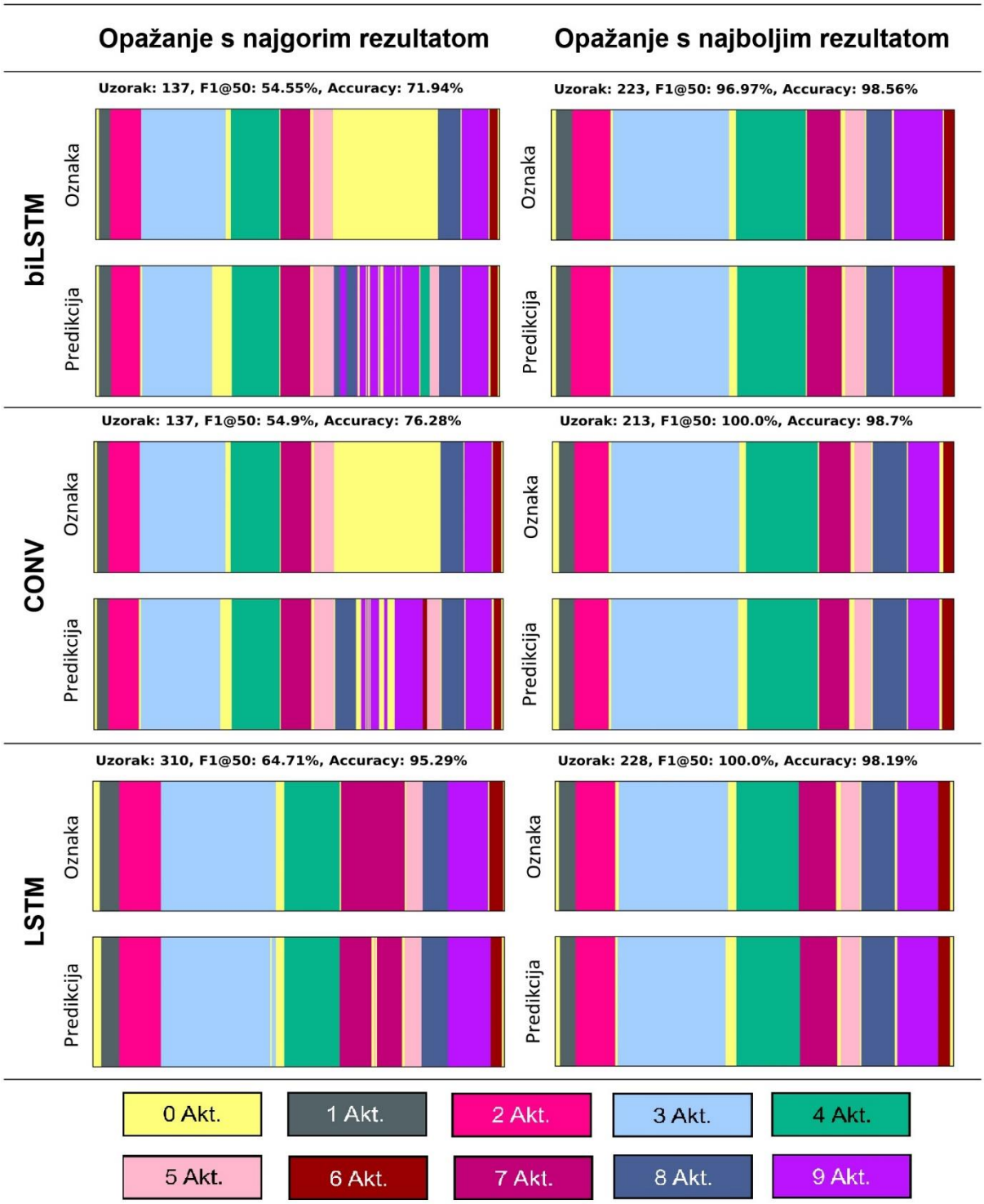
Tablica 5.16 Optimalni hiperparametri LSTM i biLSTM modela za kadar HE i pristup TL

Arhitektura	**LS**	**LN**	**SU**	**RSU**	**REG**
LSTM	1	16	$[1 \cdot 10^{-2}, 1]$	ciklički	0
biLSTM	1	8	$[5 \cdot 10^{-2}, 6 \cdot 10^{-1}]$	ciklički	0

Tablica 5.17 Optimalni hiperparametri konvolucijskog modela za kadar HE i pristup TL

Arhitektura	**DRS**	**DDS**	**BJ**	**VDR**	**SU**	**RSU**
CONV	5	5	64	0,5	$[1 \cdot 10^{-3}, 9 \cdot 10^{-2}]$	ciklički

Na slici 5.38 prikazan je segmentacijski graf za opažanja na kojima su modeli iz ove grupe ostvarili najbolje i najgore rezultate.



Slika 5.38 Segmentacijski graf za opažanja na kojima su ostvareni najgori i najbolji rezultati modela za kadar HE i pristup TL

Zaključak 3: biLSTM i CONV modeli najviše griješe na opažanju 137, pri čemu su razlozi pojave grešaka povezani s ovim opažanjem već ranije opisani. LSTM model ima problema s opažanjem 310 kojeg izvodi operater O4, gdje model za vrijeme trajanja 7. aktivnosti pogrešno prepoznaje pozadinske aktivnosti. Analiza ovog opažanja ukazuje da je razlog koji dovodi do

zabune modela taj da operater za vrijeme trajanja 7.aktivnosti krene razvrstavati kopče koje je odložio na radnom stolu.

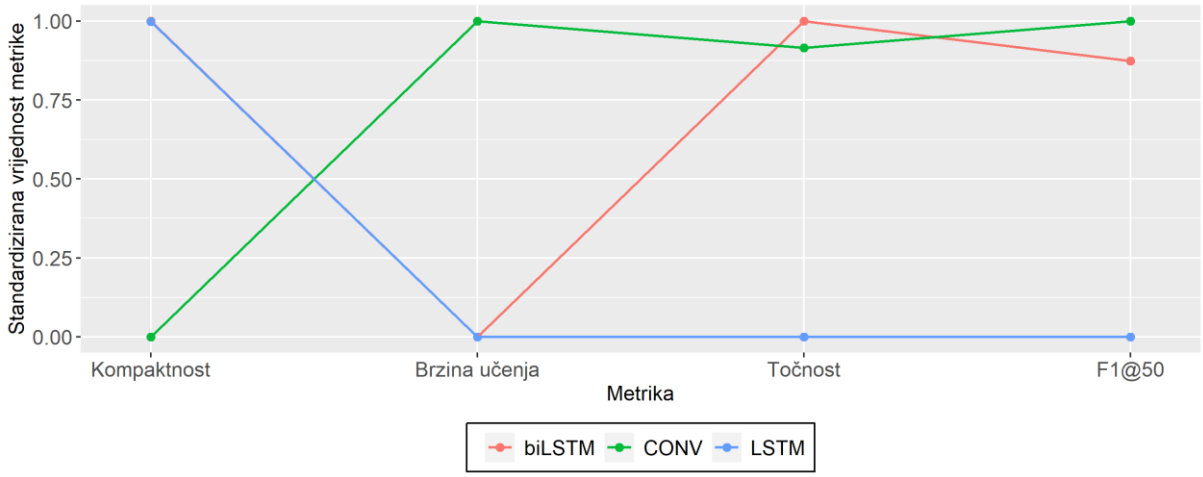
Analiza modela za kadar Fokus i pristup izvlačenja značajki TL

Usporedba modela razvijenih na značajkama dobivenima TL pristupom, iz podataka prikupljenih u Fokus kadru snimanja, na temelju STU pokazatelja izračunatog s tri različite konfiguracije težinskih koeficijenata dana je u tablici 5.18.

Tablica 5.18 Tri načina izračuna STU pokazatelja modela za kadar Fokus i pristup TL

Arhitektura	STU		
	Težište na učinkovitosti	Ujednačena važnost svih metrika	Težište na funkcionalnosti
biLSTM	0,7913	0,7185	0,6457
CONV	0,8049	0,7287	0,6525
LSTM	0,1663	0,2495	0,3326

Zaključak 1: CONV model je najbolji neovisno od toga na čemu je težište STU pokazatelja, pri čemu je biLSTM neznatno lošiji za ovu vrstu podataka.



Slika 5.39 Standardizirane vrijednosti 4 metrike kod modela za kadar Fokus i pristup TL

Graf sa slike 5.39 sugerira da je CONV model najbolji po pitanju dva elementa STU pokazatelja, pri čemu je njegova slabija strana kompaktnost, dok je biLSTM model nešto bolji od CONV modela po pitanju točnosti, a malo slabiji po pitanju F1 metrike.

Zaključak 2: Razlozi zašto CONV i biLSTM modeli imaju gotovo jednaku vrijednost STU pokazatelja su ti da je po pitanju brzine konvergencije CONV model 2,6 puta brži, a biLSTM ima 2,9 puta manje parametara, dok su im točnost i F1 metrika neznatno različite (vidi tablicu 5.5).

U nastavku su prikazane optimalne vrijednosti ključnih hiperparametara za sva tri tipa modela.

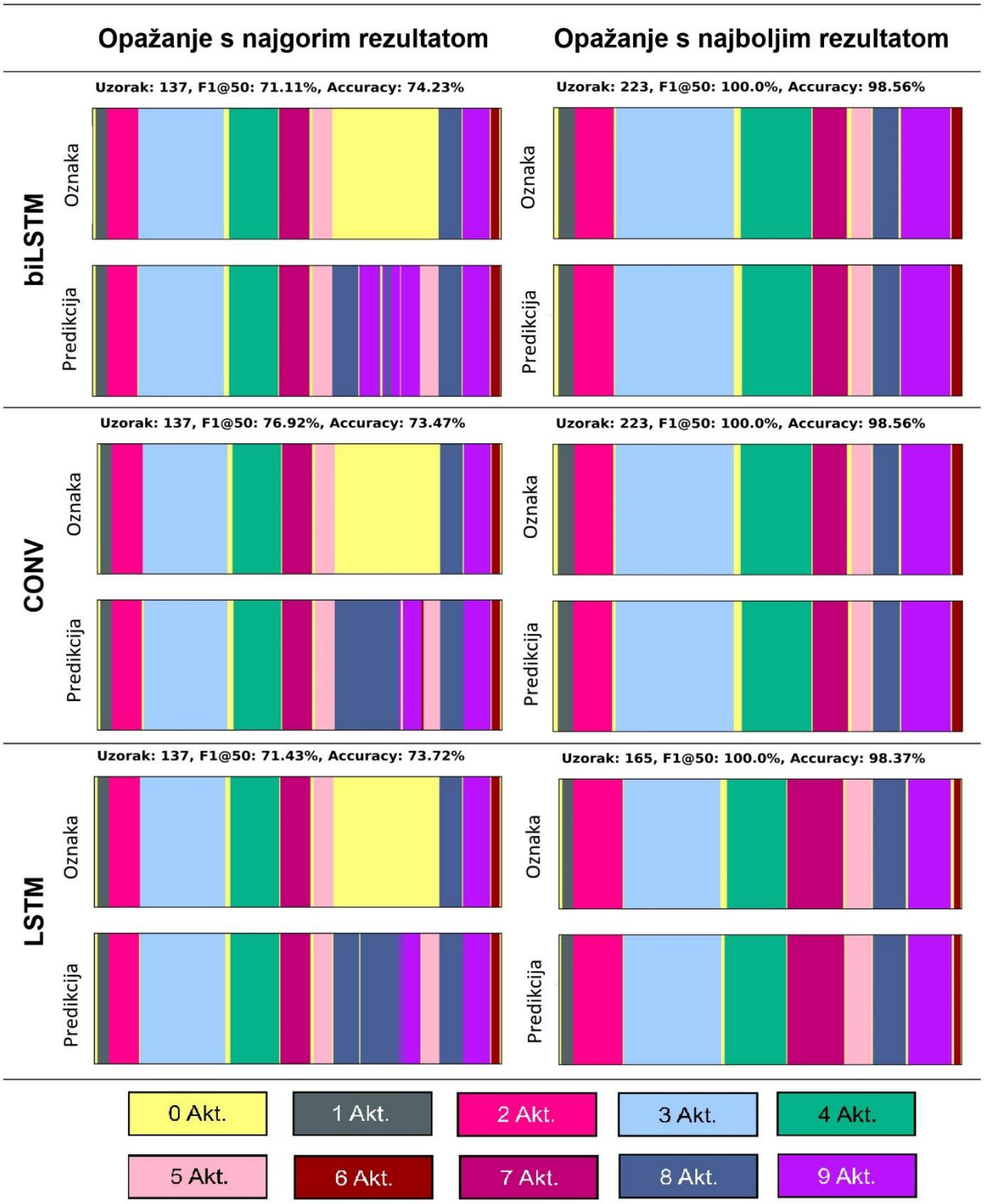
Tablica 5.19 Optimalni hiperparametri LSTM i biLSTM modela za kadar Fokus i pristup TL

Arhitektura	**LS**	**LN**	**SU**	**RSU**	**REG**
LSTM	1	16	$[1 \cdot 10^{-2}, 1]$	ciklički	0
biLSTM	1	8	$[3 \cdot 10^{-2}, 1]$	ciklički	0

Tablica 5.20 Optimalni hiperparametri konvolucijskog modela za kadar Fokus i pristup TL

Arhitektura	**DRS**	**DDS**	**BJ**	**VDR**	**SU**	**RSU**
CONV	5	5	64	0,5	$[1 \cdot 10^{-3}, 9 \cdot 10^{-2}]$	ciklički

Na slici 5.40 prikazan je segmentacijski graf za opažanja na kojima su modeli iz ove grupe ostvarili najbolje i najgore rezultate.



Slika 5.40 Segmentacijski graf za opažanja na kojima su ostvareni najgori i najbolji rezultati modela za kadar Fokus i pristup TL

Zaključak 3: Za ovu vrstu ulaznih podataka sva tri tipa modela najviše griješe na opažanju 137, pri čemu su razlozi pojave grešaka povezani s ovim opažanjem ranije opisani.

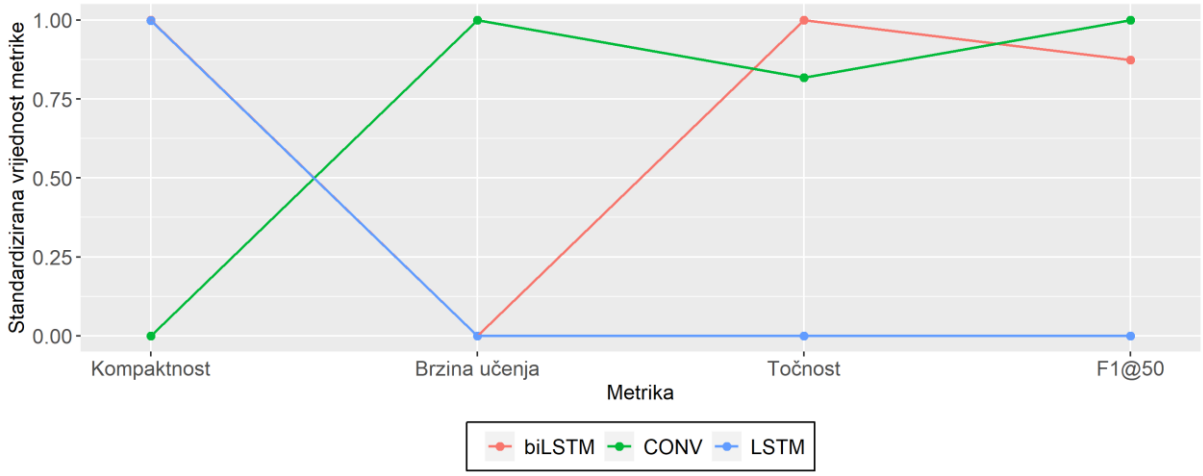
Analiza modela za kadar Concat i pristup izvlačenja značajki TL

Usporedba modela razvijenih na značajkama dobivenima TL pristupom, iz podataka prikupljenih u Concat kadru snimanja, na temelju STU pokazatelja izračunatog s tri različite konfiguracije težinskih koeficijenata dana je u tablici 5.21.

Tablica 5.21 Tri načina izračuna STU pokazatelja modela za kadar Concat i pristup TL

Arhitektura	STU		
	Težište na učinkovitosti	Ujednačena važnost svih metrika	Težište na funkcionalnosti
biLSTM	0,7910	0,7183	0,6455
CONV	0,7727	0,7045	0,6364
LSTM	0,1663	0,2495	0,3326

Zaključak 1: biLSTM model je najbolji neovisno od toga na čemu je težište STU pokazatelja, dok je CONV model neznatno lošiji za ovu vrstu podataka.



Slika 5.41 Standardizirane vrijednosti 4 metrike kod modela za kadar Concat i pristup TL

Graf sa slike 5.41 sugerira da je biLSTM model najbolji za metriku kompaktnosti i točnosti, a CONV model je najbolji kod brzine učenja i F1 metrike.

Zaključak 2: Razlozi zašto biLSTM model ima nešto bolju vrijednost STU pokazatelja od CONV modela su ti da je biLSTM 2 puta manji, dok je CONV 1,2 puta brži, a točnost i F1 metrika su im neznatno različite (vidi tablicu 5.5).

U nastavku su prikazane optimalne vrijednosti ključnih hiperparametara za sva tri tipa modela.

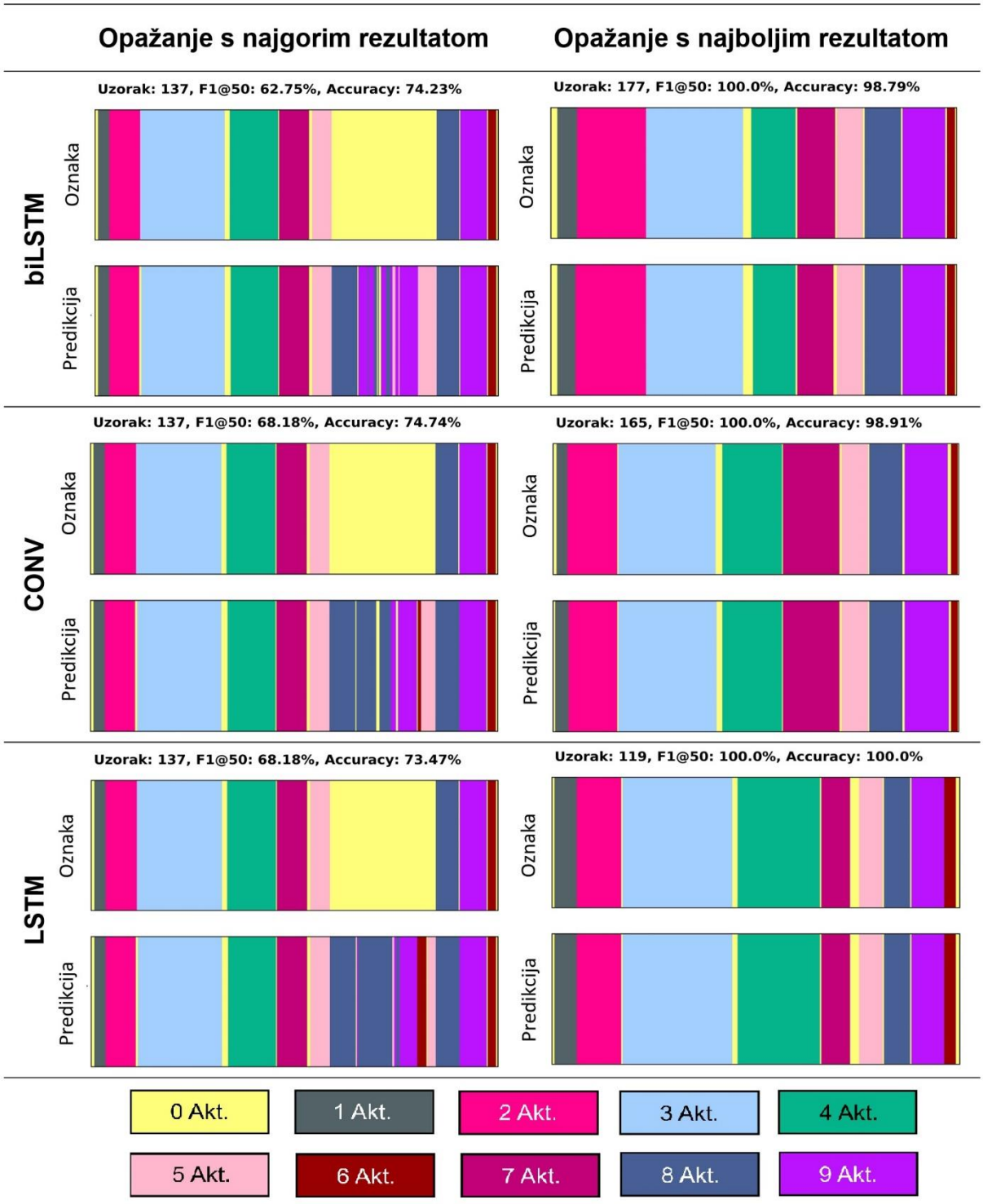
Tablica 5.22 Optimalni hiperparametri LSTM i biLSTM modela za kadar Concat i pristup TL

Arhitektura	**LS**	**LN**	**SU**	**RSU**	**REG**
LSTM	1	16	$[1 \cdot 10^{-2}, 1]$	ciklički	0
biLSTM	1	8	$[1 \cdot 10^{-2}, 6 \cdot 10^{-1}]$	ciklički	0

Tablica 5.23 Optimalni hiperparametri konvolucijskog modela za kadar Concat i pristup TL

Arhitektura	**DRS**	**DDS**	**BJ**	**VDR**	**SU**	**RSU**
CONV	5	5	64	0,5	$[1 \cdot 10^{-4}, 9 \cdot 10^{-2}]$	ciklički

Na slici 5.42 prikazan je segmentacijski graf za opažanja na kojima su modeli iz ove grupe ostvarili najbolje i najgore rezultate.



Slika 5.42 Segmentacijski graf za opažanja na kojima su ostvareni najgori i najbolji rezultati modela za kadar Concat i pristup TL

Zaključak 3: Za ovu vrstu ulaznih podataka sva tri tipa modela najviše griješe na opažanju 137, pri čemu su razlozi pojave grešaka povezani s ovim opažanjem ranije opisani.

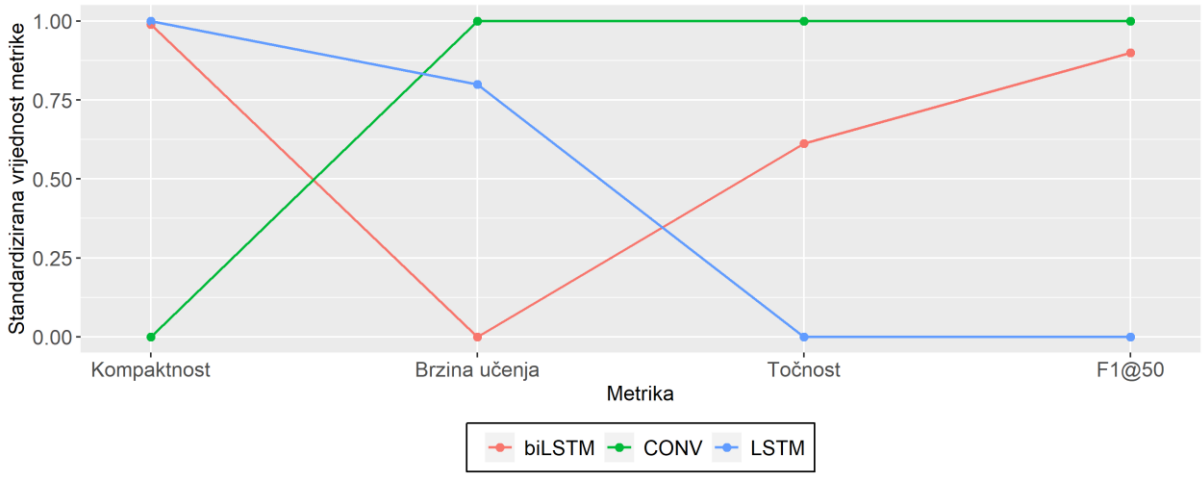
Analiza modela za kadar HE i pristup izvlačenja značajki TB

Usporedba modela razvijenih na značajkama dobivenima TB pristupom, iz podataka prikupljenih u HE kadru snimanja, na temelju STU pokazatelja izračunatog s tri različite konfiguracije težinskih koeficijenata dana je u tablici 5.24.

Tablica 5.24 Tri načina izračuna STU pokazatelja modela za kadar HE i pristup TB

Arhitektura	STU		
	Težište na učinkovitosti	Ujednačena važnost svih metrika	Težište na funkcionalnosti
biLSTM	0,6689	0,6253	0,5817
CONV	0,8333	0,7500	0,6667
LSTM	0,3000	0,4500	0,6000

Zaključak 1: CONV model je najbolji neovisno od toga na čemu je težište STU pokazatelja. S druge strane kada je težište na funkcionalnim metrikama, LSTM model je nešto bolji od biLSTM modela, dok je u slučaju težišta na točnosti i F1 metrici puno bolji biLSTM model.



Slika 5.43 Standardizirane vrijednosti 4 metrike kod modela za kadar HE i pristup TB

Graf sa slike 5.43 sugerira da je CONV model najbolji po pitanju tri elementa za ovu vrstu ulaznih podataka, pri čemu je njegova slaba strana kompaktnost.

Zaključak 2: Razlozi zašto CONV model ima najbolju vrijednost STU pokazatelja su ti da je po pitanju apsolutne vrijednosti F1 metrike za 4 boda bolji od LSTM modela te je 2,6 puta brži od biLSTM modela (vidi tablicu 5.5).

U nastavku su prikazane optimalne vrijednosti ključnih hiperparametara za sva tri tipa modela.

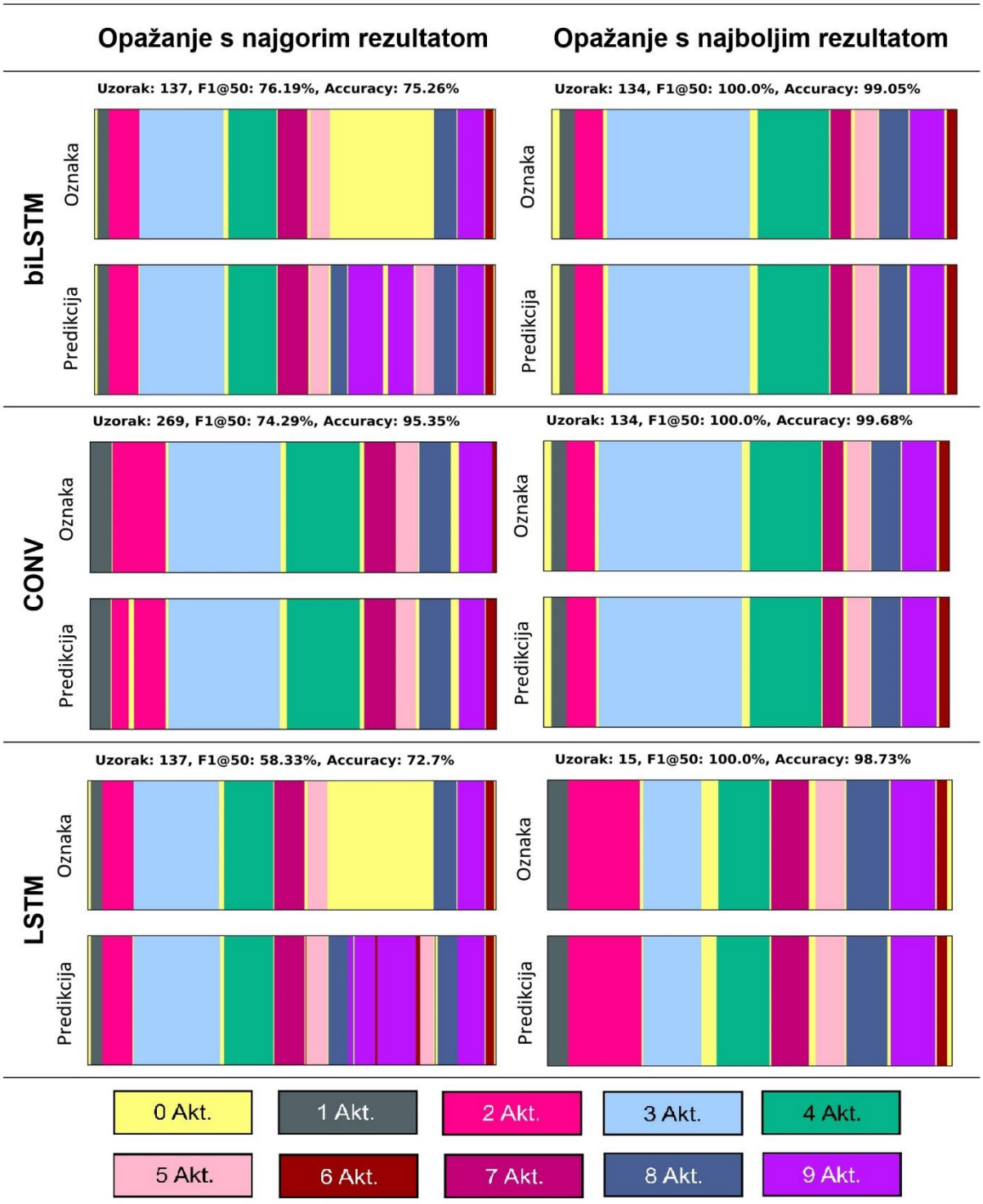
Tablica 5.25 Optimalni hiperparametri LSTM i biLSTM modela za kadar HE i pristup TB

Arhitektura	**LS**	**LN**	**SU**	**RSU**	**REG**
LSTM	1	16	$[1 \cdot 10^{-2}, 1]$	ciklički	0
biLSTM	3	8	$[4 \cdot 10^{-2}, 5 \cdot 10^{-1}]$	ciklički	0

Tablica 5.26 Optimalni hiperparametri konvolucijskog modela za kadar HE i pristup TB

Arhitektura	**DRS**	**DDS**	**BJ**	**VDR**	**SU**	**RSU**
CONV	5	5	64	0,5	$[1 \cdot 10^{-4}, 1 \cdot 10^{-2}]$	ciklički

Na slici 5.44 prikazan je segmentacijski graf za opažanja na kojima su modeli iz ove grupe ostvarili najbolje i najgore rezultate.



Slika 5.44 Segmentacijski graf za opažanja na kojima su ostvareni najgori i najbolji rezultati modela za kadar HE i pristup TB

Zaključak 3: Za ovu vrstu ulaznih podataka biLSTM i LSTM modeli najviše griješe na opažanju 137, a CONV model na opažanju 269, pri čemu su razlozi grešaka povezani s ovim opažanjima već ranije opisani.

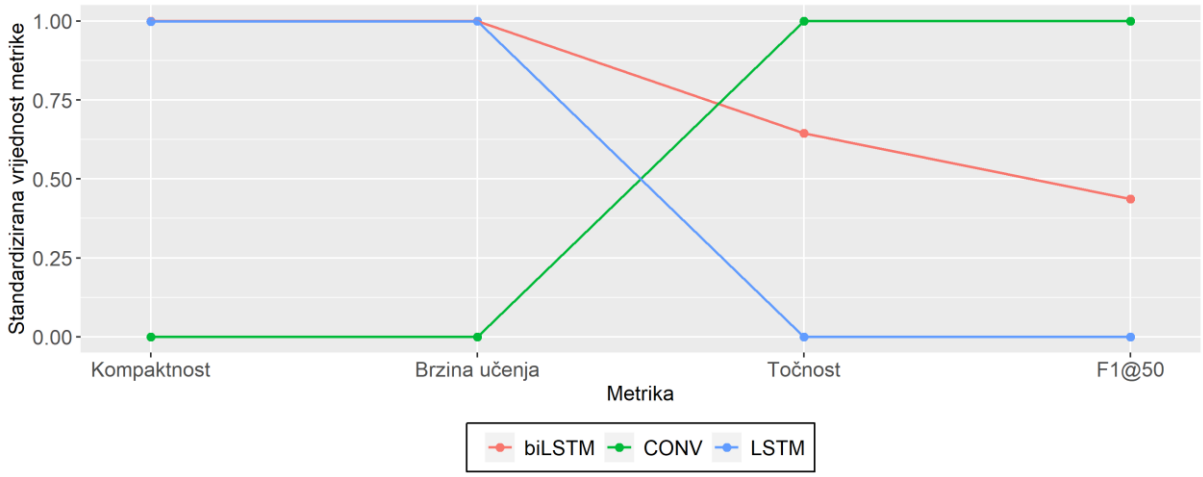
Analiza modela za kadar Fokus i pristup izvlačenja značajki TB

Usporedba modela razvijenih na značajkama dobivenima TB pristupom, iz podataka prikupljenih u Fokus kadru snimanja, na temelju STU pokazatelja izračunatog s tri različite konfiguracije težinskih koeficijenata dana je u tablici 5.27.

Tablica 5.27 Tri načina izračuna STU pokazatelja modela za kadar Fokus i pristup TB

Arhitektura	STU		
	Težište na učinkovitosti	Ujednačena važnost svih metrika	Težište na funkcionalnosti
biLSTM	0,6939	0,7704	0,8469
CONV	0,6667	0,5000	0,3333
LSTM	0,3330	0,4995	0,6659

Zaključak 1: biLSTM model je najbolji neovisno od toga na čemu je težište STU pokazatelja. S druge strane kada je težište na funkcionalnim metrikama, LSTM model je bolji od CONV modela, dok je u slučaju težišta na točnosti i F1 metrici bolji LSTM model.



Slika 5.45 Standardizirane vrijednosti 4 metrike kod modela za kadar Fokus i pristup TB

Graf sa slike 5.45 sugerira da je biLSTM model najbolji po pitanju funkcionalnih metrika za ovu vrstu ulaznih podataka, pri čemu je nešto lošiji od CONV modela kod metrike točnosti i F1 metrike.

Zaključak 2: Razlozi zašto biLSTM model ima najbolju vrijednost STU pokazatelja su ti da je 1,5 puta brži te ima skoro 3 puta manje parametara od CONV modela, dok je razlika između njih po pitanju točnosti neznatna (vidi tablicu 5.5).

U nastavku su prikazane optimalne vrijednosti ključnih hiperparametara za sva tri tipa modela.

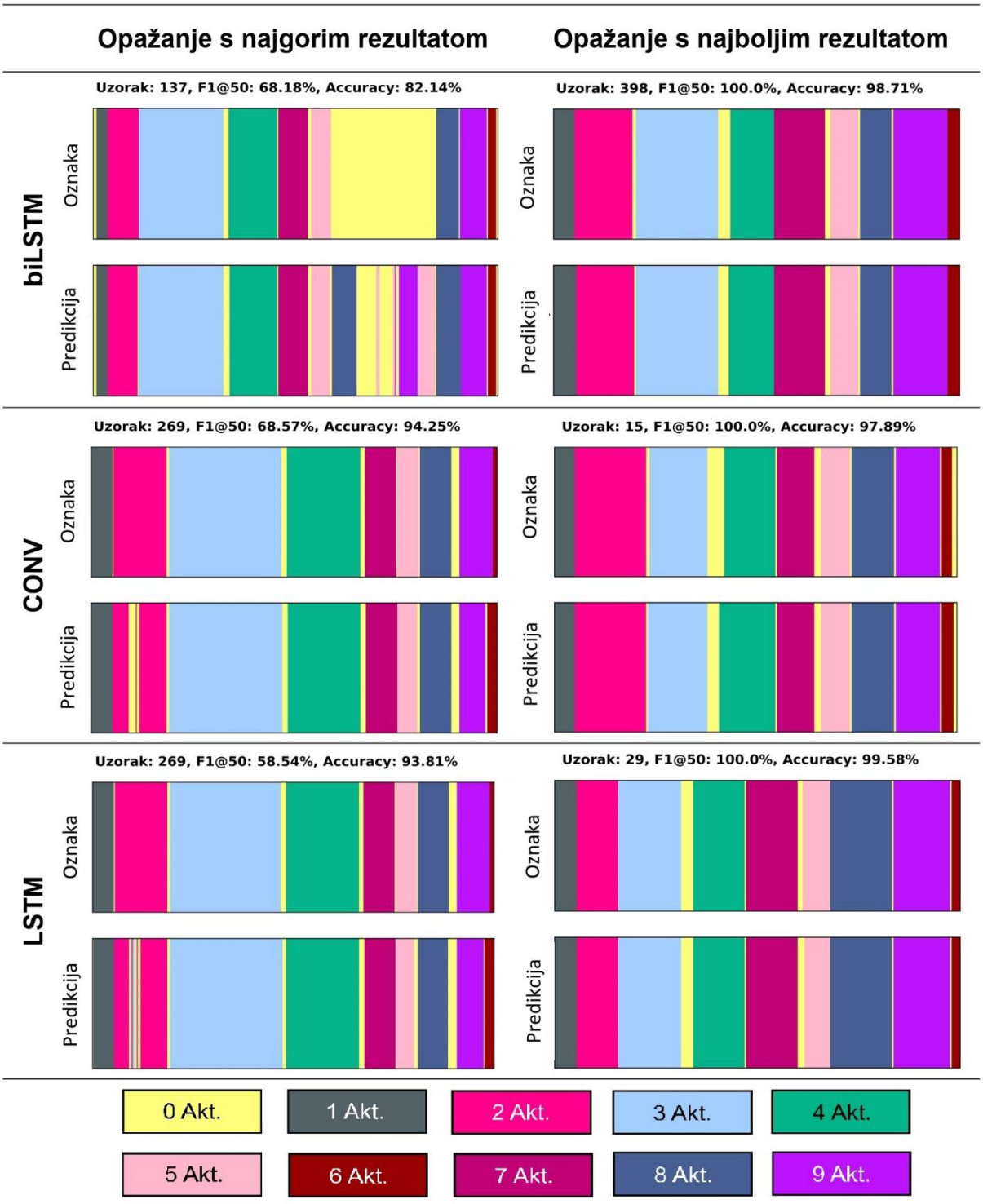
Tablica 5.28 Optimalni hiperparametri LSTM i biLSTM modela za kadar Fokus i pristup TB

Arhitektura	**LS**	**LN**	**SU**	**RSU**	**REG**
LSTM	1	16	$[1 \cdot 10^{-2}, 1]$	ciklički	0
biLSTM	1	8	$[1 \cdot 10^{-1}, 1]$	ciklički	0

Tablica 5.29 Optimalni hiperparametri konvolucijskog modela za kadar Fokus i pristup TB

Arhitektura	**DRS**	**DDS**	**BJ**	**VDR**	**SU**	**RSU**
CONV	5	5	64	0,5	$[1 \cdot 10^{-4}, 5 \cdot 10^{-2}]$	ciklički

Na slici 5.46 prikazan je segmentacijski graf za opažanja na kojima su modeli iz ove grupe ostvarili najbolje i najgore rezultate.



Slika 5.46 Segmentacijski graf za opažanja na kojima su ostvareni najgori i najbolji rezultati modela za kadar Fokus i pristup TB

Zaključak 3: Za ovu vrstu ulaznih podataka CONV i LSTM modeli najviše griješe na opažanju 269, a biLSTM model na opažanju 137, pri čemu su razlozi grešaka povezani s ovim opažanjima već ranije opisani.

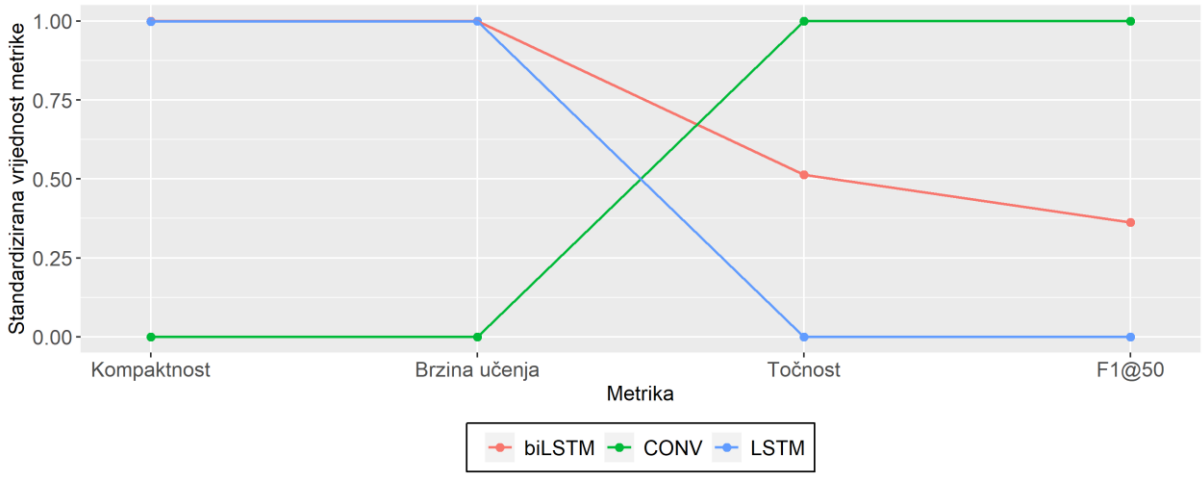
Analiza modela za kadar Concat i pristup izvlačenja značajki TB

Usporedba modela razvijenih na značajkama dobivenima TB pristupom, iz podataka prikupljenih u Concat kadru snimanja, na temelju STU pokazatelja izračunatog s tri različite konfiguracije težinskih koeficijenata dana je u tablici 5.30.

Tablica 5.30 Tri načina izračuna STU pokazatelja modela za kadar Concat i pristup TB

Arhitektura	STU		
	Težište na učinkovitosti	Ujednačena važnost svih metrika	Težište na funkcionalnosti
biLSTM	0,6255	0,7191	0,8127
CONV	0,6667	0,5000	0,3333
LSTM	0,3330	0,4995	0,6659

Zaključak 1: biLSTM model je najbolji u slučaju jednake važnosti elemenata STU pokazatelja i kada je težište na funkcionalnim metrikama. S druge strane, kada je težište na točnosti i F1 metrici bolji je CONV model.



Slika 5.47 Standardizirane vrijednosti 4 metrike kod modela za kadar Concat i pristup TB

Graf sa slike 5.47 sugerira da je biLSTM model najbolji po pitanju funkcionalnih metrika za ovu vrstu ulaznih podataka, pri čemu je lošiji od CONV modela kod metrike točnosti i F1 metrike.

Zaključak 2: Razlozi zašto biLSTM model ima najbolju vrijednost STU pokazatelja, kada je važnost svih elemenata jednaka, su ti da je 3,3 puta brži te ima 2 puta manje parametara od CONV modela, dok je razlika između njih po pitanju točnosti neznatna (vidi tablicu 5.5). CONV model je najbolji kada je težište na učinkovitosti zato jer je po pitanju apsolutne vrijednosti F1 metrike za 2 boda bolji od biLSTM i za 3 boda bolji od LSTM modela.

U nastavku su prikazane optimalne vrijednosti ključnih hiperparametara za sva tri tipa modela.

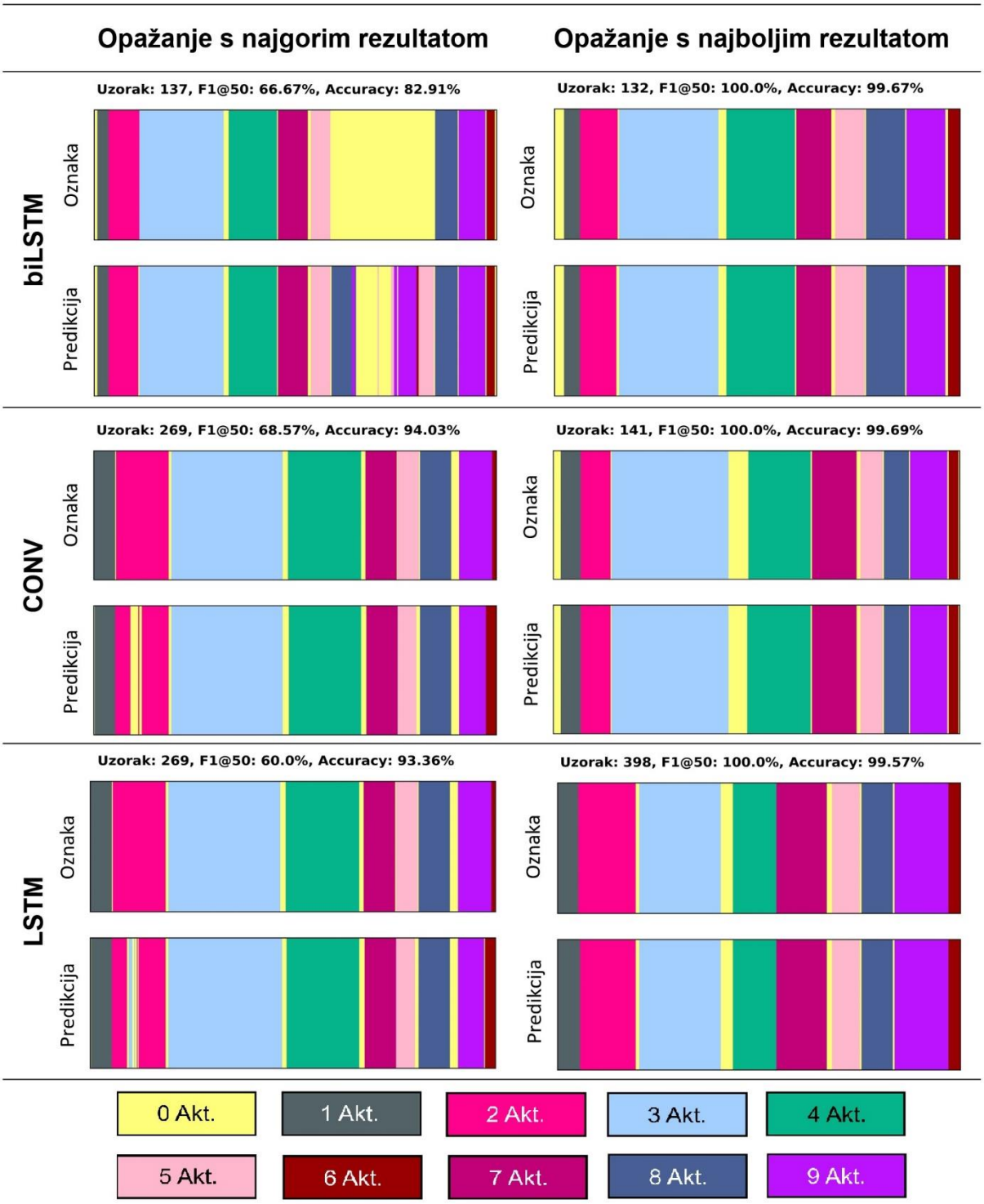
Tablica 5.31 Optimalni hiperparametri LSTM i biLSTM modela za kadar Concat i pristup TB

Arhitektura	**LS**	**LN**	**SU**	**RSU**	**REG**
LSTM	1	16	$[1 \cdot 10^{-2}, 5 \cdot 10^{-1}]$	ciklički	0
biLSTM	1	8	$[1 \cdot 10^{-2}, 6 \cdot 10^{-1}]$	ciklički	0

Tablica 5.32 Optimalni hiperparametri konvolucijskog modela za kadar Concat i pristup TB

Arhitektura	**DRS**	**DDS**	**BJ**	**VDR**	**SU**	**RSU**
CONV	5	5	64	0,2	$[1 \cdot 10^{-4}, 1 \cdot 10^{-2}]$	ciklički

Na slici 5.48 prikazan je segmentacijski graf za opažanja na kojima su modeli iz ove grupe ostvarili najbolje i najgore rezultate.

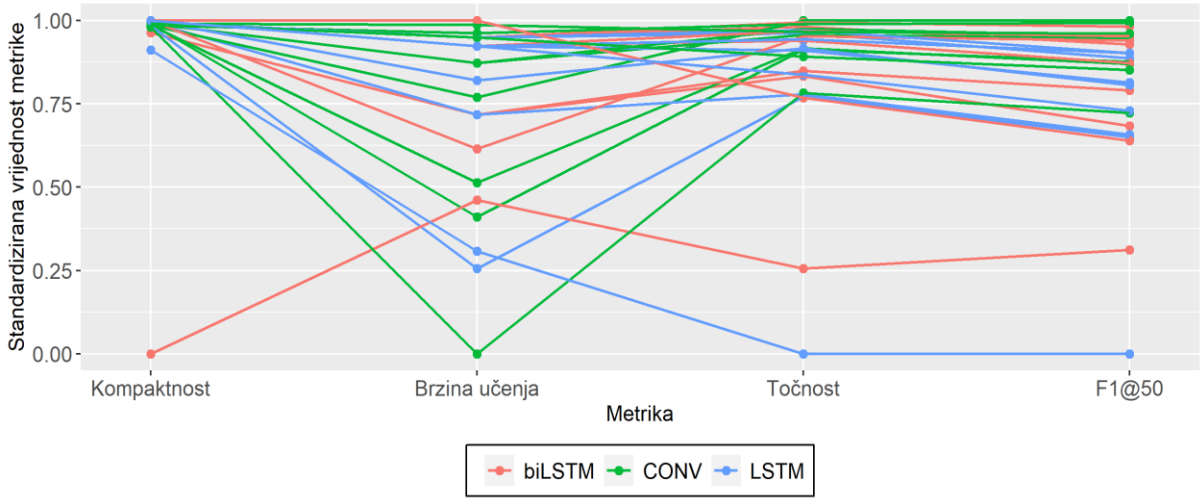


Slika 5.48 Segmentacijski graf za opažanja na kojima su ostvareni najgori i najbolji rezultati modela za kadar Concat i pristup TB

Zaključak 3: Za ovu vrstu ulaznih podataka CONV i LSTM modeli najviše griješe na opažanju 269, a biLSTM model na opažanju 137, pri čemu su razlozi grešaka povezani s ovim opažanjima već ranije opisani.

5.4.2 Izbor optimalne kombinacije ulaznih podataka i modela

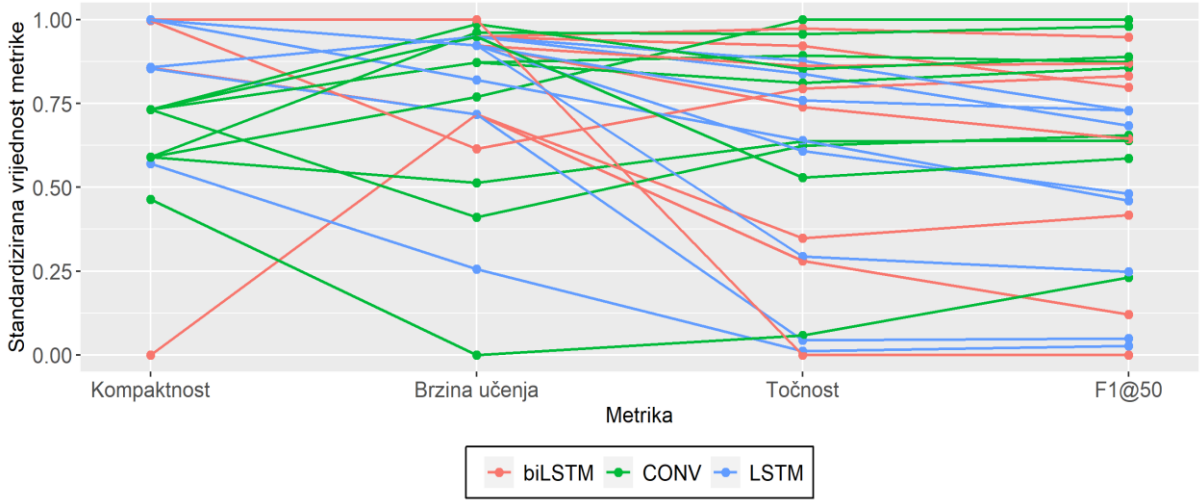
U ovom dijelu analize paralelno je uspoređeno svih 27 modela. Kao što je ranije spomenuto, za ovako postavljenu analizu vrijednost STU pokazatelja više ne ovisi isključivo o učinkovitosti modela veći i o ulaznim podatcima, s obzirom da svi modeli nemaju jednake početne uvjete. Zbog navedenog, interpretacija rezultata ići će u smjeru prepoznavanja najbolje kombinacije kadra snimanja podataka, pristupa izvlačenju značajki i modela. Na slici 5.49 prikazane su standardizirane vrijednosti četiri komponente STU pokazatelja izračunate na temelju rezultata svih 27 modela. Na temelju ovog grafa zaključeno je da dva modela imaju značajne vrijednosti metrika izvan raspona ostatka podataka, a uslijed toga ostali modelu djeluju znatno bolji, što se vidi na temelju toga da je većina krivulja po pitanju kompaktnosti, točnosti i F1 metrike u gornjem dijelu grafa. Konkretno radi se o biLSTM i LSTM modelu (oznake BFEHE i LFEHE) koji su razvijeni na temelju podataka iz kadra HE postupkom izvlačenja značajki FE.



Slika 5.49 Standardizirane vrijednosti 4 metrike svih 27 modela

Graf 5.49 sugerira da su spomenuti modeli najslabiji iz aspekta točnosti, F1 metrike i kompaktnosti. Pregledom tablice 5.5 vidljivo je da BFEHE model ima 25 puta više parametara, a LFEHE model 2 puta više parametara od modela koji su u uobičajenom rasponu podataka. Točnost i F1 metrika LFEHE modela je 87,82% i 68,19%, dok su ove metrike za BFEHE model 90,38% i 76,14%. Kada se dodatno analizira i konvolucijski model (oznaka CFEHE) razvijen na prethodno spomenutoj vrsti podataka, dolazi se do zaključka da kombinacija HE kadra i FE pristupa podataka rezultira najslabijim modelima. Ovo je očekivano ponašanje, ako se u obzir uzme da je izvlačenje značajki napravljeno na temelju FE modela koji je prednaučen na različitim podatcima od predmetnog skupa podataka iz proizvodnje, pri čemu nije rađeno fino podešavanje. Općenito, rezultati iz tablice 5.5 ukazuju da su modeli razvijeni na značajkama iz

FE pristupa najslabiji. Zaključak uvodnog dijela analize je da će modeli BFEHE i LFEHE biti isključeni iz daljnje evaluacije i traženja najbolje kombinacije ulaznih podataka i modela. Graf standardiziranih vrijednosti elemenata STU metrike bez isključenih modela je na slici 5.50.



Slika 5.50 Standardizirane vrijednosti 4 metrike bez "BFEHE" i "LFEHE" modela

Na gornjoj slici su malo jasniji trendovi po pitanju vrijednosti 4 metrike, pri čemu je generalni zaključak da su modeli s CONV arhitekturom bolji po pitanju točnosti i F1 metrike, dok biLSTM i LSTM modeli imaju prednost kod funkcionalnih metrika. Ovo ponašanje je kontra intuitivno s aspekta brzine učenja jer je inicijalna hipoteza bila da će biLSTM i LSTM modeli zbog sekvencijalnog pristupa obradi podataka biti sporiji od CONV arhitekture. Dva su moguća objašnjenja spomenutog fenomena. Prvo objašnjenje je da su biLSTM i LSTM modeli bili u pravilu kompaktniji od CONV modela, izuzev isključenih BFEHE i LFEHE modela, a drugo je da su trebali manji broj epoha do konvergencije iako je vrijeme obrade podataka po pojedinačnoj epohi bilo slično (vidi tablicu 5.5). Također očekivalo se da će biLSTM i LSTM modeli biti točniji od CONV modela, ali se primjena dilatiranih konvolucijskih slojeva pokazala učinkovitijim rješenjem. Potencijalni razlozi leže u činjenici da je korištenje dvije vremenske grane odgovarajući mehanizam za prepoznavanje aktivnosti iz analiziranog uzorka. Nastavak analize slijedit će rezultate iz tablice 5.33. Prvo će biti spomenute tri najbolje kombinacije ulaznih podataka i modela za svaki od tri načina izračuna STU pokazatelja, a završno će biti spomenuta najbolja kombinacija za pojedinačne komponente metrike.

Tablica 5.33 Modeli sortirani po STU pokazatelju uz jednaku važnost četiri metrike (bez "BFEHE" i "LFEHE" modela)

Oznaka modela	Arhitektura	Pristup izvlačenju značajki	Kadar	Kompaktnost	Brzina učenja	Točnost	F1@50	STU (jednako)	STU (funkcionalnost)	STU (učinkovitost)
BTLC	biLSTM	TL	Concat	0,8584	0,9487	0,9737	0,9477	**0,9321**	0,9226	0,9417
BTLF	biLSTM	TL	Fokus	1,0000	0,9231	0,8621	0,8696	**0,9137**	0,9297	0,8978
BTBC	biLSTM	TB	Concat	0,8584	0,9487	0,9212	0,7988	**0,8818**	0,8890	0,8745
CTLC	CONV	TL	Concat	0,5898	0,9615	0,9562	0,9796	**0,8718**	0,8397	0,9038
CTLF	CONV	TL	Fokus	0,7314	0,9872	0,8534	0,8898	**0,8654**	0,8634	0,8675
LTLC	LSTM	TL	Concat	0,8579	0,9487	0,8774	0,7281	**0,8530**	0,8698	0,8363
LTLF	LSTM	TL	Fokus	0,9994	0,9231	0,7593	0,7298	**0,8529**	0,8890	0,8168
CTBHE	CONV	TB	HE	0,7314	0,8718	0,8928	0,8733	**0,8423**	0,8288	0,8559
CTBC	CONV	TB	Concat	0,5898	0,7692	1,0000	1,0000	**0,8398**	0,7863	0,8932
LTBC	LSTM	TB	Concat	0,8579	0,9487	0,8381	0,6843	**0,8322**	0,8559	0,8086
BTBF	biLSTM	TB	Fokus	1,0000	0,9231	0,7396	0,6449	**0,8269**	0,8718	0,7820
CTBF	CONV	TB	Fokus	0,7314	0,8718	0,8118	0,8565	**0,8179**	0,8125	0,8233
BTBHE	biLSTM	TB	HE	0,9965	0,6154	0,7943	0,8320	**0,8096**	0,8084	0,8108
LTBF	LSTM	TB	Fokus	0,9994	0,9231	0,6083	0,4810	**0,7530**	0,8224	0,6835
LTBHE	LSTM	TB	HE	0,9994	0,8205	0,6389	0,4600	**0,7297**	0,7898	0,6696
CTLHE	CONV	TL	HE	0,7314	0,9487	0,5295	0,5859	**0,6989**	0,7459	0,6518
LTLHE	LSTM	TL	HE	0,9994	0,9231	0,2932	0,2482	**0,6160**	0,7311	0,5009
CFEF	CONV	FE	Fokus	0,7314	0,4103	0,6236	0,6555	**0,6052**	0,5938	0,6167
CFEC	CONV	FE	Concat	0,5898	0,5128	0,6367	0,6395	**0,5947**	0,5802	0,6092
BFEF	biLSTM	FE	Fokus	0,8565	0,7179	0,3479	0,4171	**0,5849**	0,6523	0,5174
BTLHE	biLSTM	TL	HE	1,0000	1,0000	0,0000	0,0000	**0,5000**	0,6667	0,3333
LFEF	LSTM	FE	Fokus	0,8543	0,7179	0,0438	0,0497	**0,4164**	0,5397	0,2932
BFEC	biLSTM	FE	Concat	0,0000	0,7179	0,2801	0,1211	**0,2798**	0,3062	0,2534
LFEC	LSTM	FE	Concat	0,5712	0,2564	0,0110	0,0272	**0,2164**	0,2822	0,1507
CFEHE	CONV	FE	HE	0,4642	0,0000	0,0591	0,2313	**0,1887**	0,2031	0,1742

Kada je važnost komponenti STU pokazatelja ujednačena, tri najbolja modela su biLSTM modeli pod oznakama BTLC, BTLF i BTBC pri čemu je BTLC balansirano rješenje između funkcionalnosti i učinkovitosti dok druga dva daju malo veći naglasak na funkcionalnost. Prvi i treći model koriste podatke iz oba kadra (Concat) što je u skladu s očekivanjem da će korištenje informacija iz različitih kadrova pozitivno utjecati na učinkovitost modela. Prvi i drugi model su koristili TL pristup za izvlačenje značajki dok je treći koristio TB pristup. Ovo znači da korištenje vlastitih podataka za fino podešavanje prednaučenog modela (TL), te korištenje takvog modela za izvlačenje deskriptivnih značajki, dovodi do boljih rezultata prema STU pokazatelju u odnosu na izvlačenje podataka primjenom modela koji je naučen samo na vlastitim podatcima (TB). Pozitivan utjecaj korištenja TL pristupa za izvlačenje značajki vidljiv je i na primjeru modela rangiranih od 4. do 7. mjesta prema STU pokazatelju. Opisani rezultati govore u prilog tehnici prijenosa znanja s finim podešavanjem (vidi odjeljak 5.2) u razvoju modela dubokog strojnog učenja kao učinkovitog rješenja u uvjetima male količine podataka. Kada je težište STU pokazatelja na funkcionalnim metrikama, isti modeli, kao i kod ujednačene važnosti komponenti STU pokazatelja, čine grupu od tri najbolja. Međutim, sada BTLF model zauzima prvo mjesto, a LSTM model oznake LTLF ima jednaku vrijednost STU pokazatelja kao i BTBC model. Ovo je sukladno trendovima opaženima na slici 5.50. Kada su točnost i F1 metrika imperativne, prva tri modela prema STU pokazatelju su BTLC, CTLC i BTLF, pri čemu je na četvrtom mjestu konvolucijski model CTBC koji je s aspekta točnosti i F1 metrike najbolji u odnosu na sve druge modele, ali zbog slabe funkcionalne metrike, nije prema STU pokazatelju u prva tri modela. Spomenuti CTBC model ukazuje na to da su sama točnost i F1 metrika najbolje u slučaju korištenja modela za izvlačenje značajki koji je razvijen samo na vlastitim podatcima uz korištenje informacija iz oba kadra snimanja.

Završni dio analize povezan je uz apsolutne vrijednosti metrika modela iz tablice 5.5. Model s najvećom točnosti i F1@IoU metrikom, kao što je ranije spomenuto, je konvolucijski model CTBC s točnosti od 97,84% i segmentacijskom F1 metrikom od 93,64%. Model s najboljim funkcionalnim metrikama je biLSTM model BTLHE s 131.818 parametara i vremenom učenja od 20 sekundi. Prosječna točnost svih modela je 96,37%, a prosječna vrijednost F1 metrike je 88,57%, dok se broj parametara modela kreće oko 330.000 uz vrijeme učenja od 190 s. U slučaju da je točnost imperativ, konvolucijski modeli predstavljaju odgovarajuće rješenje, a ako je težište na funkcionalnosti, biLSTM modeli su se pokazali kao bolja opcija na korištenom skupu podataka. Iako je nezahvalno uspoređivati modele razvijene na različitim skupovima podataka, bit će spomenuti rezultati najboljeg modela iz literature koji je primjenjivan u

proizvodnim uvjetima kako bi se stekao dojam o učinkovitosti postojećih modela u odnosu na modele iz disertacije. Model iz rada [18] daje najbolje rezultate u odnosu na sve ostale iz domene proizvodnje. Spomenuti model ostvario je preciznost od 86% i odziv od 89%, iz čega slijedi da je točnost 87,5%, ali uz bitnu napomenu da su kod ocjene modela izbačene pozadinske aktivnosti. Pozadinske aktivnosti bitno utječu na točnost modela kao što je pokazano kod analize u prethodnom odjeljku. Također, kao što je ranije objašnjeno, metrika točnosti ne govori ništa o segmentiranosti aktivnosti, a u pravilu je gotovo uvijek niža od točnosti. Nadalje, spomenuti model koristi kao ulaz značajke izvučene fiksnim algoritmom koje su ulaz 2D konvolucijskog modela, a u odjeljku 5.2 je pokazano da 2D konvolucijski modeli rezultiraju prekomjerno segmentiranim modelima što implicira niže vrijednosti F1@IoU metrike. Sve navedene činjenice govore u prilog modelima razvijenima u disertaciji kao kvalitetnijem rješenju, ali isto tako i u prilog pristupima razvoja ulaznih značajki primjenom modela dubokog strojnog učenja. Na temelju svih podataka navedenih u analizi, moguće je zaključiti da funkcionalne karakteristike i učinkovitost modela, razvijenih na realnim podatcima u okviru ovog istraživanja, stvaraju pretpostavke za unaprjeđenje postojećih pristupa studiju vremena primjenom novo razvijenih modela.

6. ZAKLJUČAK

Ovo istraživanje bilo je potaknuto autorovim iskustvima kod provedbe studija vremena u uvjetima realnih poslovnih i proizvodnih procesa. Prepoznati su nedostatci u vidu značajnog opterećenja koje je na analitičaru prilikom konvencionalnog pristupa studiju vremena, a koje zahtjeva manualnu obradu prikupljenih podataka što je vremenski intenzivan proces. Na temelju spomenutog, moguće je zaključiti da takvi pristupi ograničavaju brzu i čestu provedbu analize, koja je uz to i pod utjecajem subjektivnosti analitičara. Prije provedbe samog istraživanja razmatrana je šira slika s pozicije svrsishodnosti unapređenja postupaka studija vremena u kontekstu postojećih trendova u poslovnim i proizvodnim sustavima. Kako je fokus studija vremena čovjek i njegova interakcija s okruženjem, a tehnološki trendovi djeluju disruptivno na ulogu čovjeka unutar sustava, razmatrana je potreba za novim pristupima studiju vremena. Nakon što je utvrđeno da postoji niz poslovnih i proizvodnih procesa u kojima se tehnički ne može, ili barem nije isplativo, zamijeniti čovjeka s tehnologijom, bilo je jasno da je nužno na učinkovit i objektivan način pratiti i vrednovati utjecaj ljudskog faktora na cijeli sustav. Ustanovljena je veza između studija vremena i domene računalnog vida preko problema istovremenog prepoznavanja i vremenske segmentacije aktivnosti, što je usmjerilo ovu disertaciju prema istraživanju i razvoju modela sposobnog za rješavanje ovog problema u okviru proizvodnih procesa.

U ovom poglavlju napravljen je osvrt na proces realizacije postavljenih ciljeva istraživanja i ostvarene znanstvene doprinose, iza kojeg slijedi sažetak prepoznatih ograničenja istraživanja na temelju kojega su posljedično definirane smjernice za daljnja istraživanja.

6.1 Osvrt na znanstvene doprinose i hipotezu istraživanja

Vodeći se gore napisanom motivacijom i utvrđenim smjerom istraživanja, napravljen je inicijalni pregled literature o postojećim pristupima vezanima za problem istovremenog prepoznavanja i vremenske segmentacije aktivnosti. Pregled je ukazao na to da su suvremeni pristupi ovom problemu zasnovani na modelima dubokog strojnog učenja, te da postoji nekolicina istraživanja provedena u kontekstu proizvodnje, pri čemu su ta istraživanja temeljena na modelima strojnog učenja s plitkom strukturom koji su naučeni na manualno razvijenim značajkama. Osim toga, pregled je ukazao da postoji potreba za modelima veće učinkovitosti koji su razvijeni na podatcima prikupljenima u realnim uvjetima. Na temelju stečenih spoznaja

iz pregleda literature postavljeni su ciljevi istraživanja te doprinosi koji će biti ostvareni njihovom realizacijom. Doprinosi ostvareni disertacijom su:

1) Izrada novog modela za istovremeno prepoznavanje i vremensku segmentaciju niza ljudskih aktivnosti u realnom proizvodnom procesu temeljenog na dubokom strojnom učenju.
2) Razvijena procedura za testiranje učinkovitosti modela na prikupljenom skupu podataka koja će omogućiti usporedbu novo razvijenih modela.

Kako bi modeli mogli biti izrađeni prikupljeni su podatci u obliku video zapisa iz realnog proizvodnog sustava iz dva različita kadra snimanja. U odjeljku 5.2 predstavljena su tri pristupa izvlačenja značajki koji su primijenjeni na ulazne podatke iz dva različita kadra i fuziju podataka s oba kadra, na temelju kojih je razvijeno 27 vrsta modela za istovremeno prepoznavanje i vremensku segmentaciju aktivnosti korištenjem tri vrste arhitektura opisanih u odjeljku 5.3. Metodologija korištena kod razvoja modela opisana je u odjeljku 5.3.4. Usporedbe i izbor optimalnih modela provedeni su u dijelu 5.4 primjenom novo razvijene procedure zasnovane na definiranim metrikama učinkovitosti i funkcionalnim karakteristikama modela koje su ugrađene u jedinstveni pokazatelj učinkovitosti.

Osim opisanih primarnih doprinosa, dodatnim doprinosima mogu se smatrati razvijena programska biblioteka *phd_lib* za efikasnu primjenu dubokog strojnog učenja na podatcima u obliku video zapisa te prikupljeni i označeni skup podataka iz realnog proizvodnog procesa opisan u poglavljima 3 i 4. Prikupljeni skup podataka može služiti za razvoj i testiranje novih pristupa iz domene istovremenog prepoznavanja i vremenske segmentacije aktivnosti primjenom računalnog vida s obzirom da se svojim karakteristikama razlikuje od postojećih javno dostupnih skupova podataka na kojima su razvijani modeli iz literature.

Postavljena hipoteza istraživanja bila je:

„Na temelju podataka prikupljenih iz realnog proizvodnog procesa moguće je razviti model temeljen na računalnom vidu i dubokom strojnom učenju koji ima sposobnost prepoznavanja i vremenske segmentacije niza ljudskih aktivnosti te će se njegovom primjenom unaprijediti postupci studija vremena i analize produktivnosti ljudskog faktora.“

Pregledom tablice 5.5 u kojoj su prikazani rezultati razvijenih modela na skupu za učenje te na temelju ostvarenih doprinosa i ciljeva istraživanja može se smatrati da je hipoteza istraživanja potvrđena. Konkretno, rezultati koji podupiru potvrdu hipoteze su ti da je prosječna točnost svih razvijenih modela 96,37%, a prosječna vrijednost segmentacijske F1 metrike uz prag od 0,5 je

88,57%. Nadalje, najbolji model ostvario je točnost od 97,84% i F1 rezultat od 93,64%. Uzimajući u obzir sve navedene rezultate može se reći da su stvorene pretpostavke za unaprjeđenje postojećih pristupa studiju vremena primjenom novo razvijenih modela.

6.2 Ograničenja provedenog istraživanja i smjernice za daljnja istraživanja

Analiza ograničenja istraživanja usmjerena je na elemente dva glavna znanstvena doprinosa: model i proceduru testiranja modela. Uz utvrđena ograničenja dana su razmišljanja o potrebnim budućim istraživanjima kojima će se nastojati otkloniti prepoznata ograničenja. Prepoznata ograničenja s povezanim smjernicama za daljnja istraživanja su:

- *Ograničenje 1: Razvoj modela u dva koraka*. Većina postojećih istraživanja, a provedeno istraživanje iz disertacije nije iznimka, iz područja istovremenog prepoznavanja i vremenske segmentacije aktivnosti kod učenja modela koriste pristup temeljen na dva odvojena korak. U prvom koraku se obično provodi izvlačenje značajki, a u drugom klasifikacija i segmentacija. Teorija dubokog strojnog učenja ukazuje na korisnost integriranog učenja značajki i finalne klasifikacije ili regresije. Iako postoje određena tehnička ograničenja kod praktične provedbe integriranog učenja na podatcima u obliku video zapisa, ovaj pristup ima potencijal za razvoj modela veće točnosti, stoga je ovo jedan od smjerova budućih istraživanja.
- *Ograničenje 2*: *Ručno traženje optimalnih hiperparametara*. U razvoju finalnih modela optimalni hiperparametri su traženi manualno na temelju poznavanja utjecaja hiperparametara na pristranost i varijancu modela. Ovaj pristup dao je dobar uvid u određena teorijska svojstva modela, međutim u kontekstu razvoja sustava koji se sposoban samostalno prilagođavati novim podatcima to je ograničavajući faktor. Krajnji cilj područja umjetne inteligencije je sustav koji ne zahtjeva ljudsku intervenciju u realnoj primjeni. Kako bi se razvijeni modeli u okviru disertacije mogli prilagođavati novim podatcima kod primjene u realnim uvjetima bit će potrebno istražiti na koji način je moguće priključiti automatizirane pristupe traženja optimalnih hiperparametara u cjelokupni tok razvoja modela te koja je vrsta automatiziranog traženja hiperparametara optimalna.
- *Ograničenje 3*: *Potpuno nadzirani pristup učenju modela.* Glavninu pristupa iz literature u domeni istraživanog problema sačinjavaju modeli razvijeni nadziranim pristupom učenja. Ovaj pristup osigurava da model ima jasan signal koji ga usmjerava u procesu učenja. Ali, kao što je i sam autor spoznao, proces kreiranja skupa podataka

za učenje nadziranim pristupima je izrazito mukotrpan i vremenski neefikasan. Moguće rješenje su pristupi polu-nadziranog i nenadziranog učenja, koji zahtijevaju manju količinu označenih podataka. Stoga je potrebno istražiti koji su to pristupi polu-nadziranog i nenadziranog učenja primjenjivi u realnim uvjetima. Konkretno, potrebno je ustanoviti odnos između točnosti razvijenih modela gore spomenutim pristupima te smanjenja opterećenja kod označavanja podataka.

- *Ograničenje 4*: *Prikupljeni skup podataka.* Korisnost modela strojnog učenja ograničena je skupom podataka na kojem je razvijen. Analiza modela iz disertacije ukazala je na to da su greške modela često povezane uz prijelaze između aktivnosti i pojavu pozadinskih aktivnosti. Ova pojava je razumljiva iz razloga što se u skupu podataka nalazi relativno mala količina označenih podataka s navedenom klasama, pri čemu se njihove vizualne karakteristike značajno razlikuju. Učinkovitost nepristranog modela moguće je povećati korištenjem većeg obujma podataka. U budućnosti je potrebno istražiti da li se većim skupom reprezentativnih podataka može povećati točnost modela iz disertacije.
- *Ograničenje 5: Procedura usporedbe modela*. U istraživanju je razvijen pokazatelj učinkovitosti modela koji je agregirana vrijednost četiri različite metrike učinkovitosti. Svrha razvijenog pokazatelja bila je pojednostavniti problem višekriterijske optimizacije i omogućiti usporedbu i evaluaciju različitih modela. Međutim, u suštini i dalje se radi o problemu višekriterijske optimizacije za čije rješenje je potrebno primijeniti odgovarajuće metode. Iz navedenih razloga jedan od budućih ciljeva je unapređenje procedure izbora modela zasnovano na metodama višekriterijske optimizacije.

Kroz otklanjanje opisanih ograničenja nastojat će se razviti sustav čijom će se implementacijom u realnim uvjetima olakšati posao analize produktivnosti ljudskog faktora u proizvodnim procesima, a osim toga očekuje se i doprinos području planiranja proizvodnje što će u krajnjoj liniji utjecati na učinkovitost cijelog proizvodnog i poslovnog sustava.

7. LITERATURA

[1] Romero D, Bernus P, Noran O, Stahre J, Fast-Berglund Å. The Operator 4.0: Human Cyber-Physical Systems & Adaptive Automation Towards Human-Automation Symbiosis Work Systems. In: Nääs I, Vendrametto O, Mendes Reis J, Gonçalves RF, Silva MT, von Cieminski G, et al., editors. Advances in Production Management Systems. Initiatives for a Sustainable World, vol. 488, Cham: Springer International Publishing; 2016, p. 677–86. https://doi.org/10.1007/978-3-319-51133-7_80.

[2] Xu LD, Duan L. Big data for cyber physical systems in industry 4.0: a survey. Enterprise Information Systems 2019;13:148–69. https://doi.org/10.1080/17517575.2018.1442934.

[3] Pfeiffer S. Robots, Industry 4.0 and Humans, or Why Assembly Work Is More than Routine Work. Societies 2016;6:16. https://doi.org/10.3390/soc6020016.

[4] Posada J, Zorrilla M, Dominguez A, Simoes B, Eisert P, Stricker D, et al. Graphics and Media Technologies for Operators in Industry 4.0. IEEE Computer Graphics and Applications 2018;38:119–32. https://doi.org/10.1109/MCG.2018.053491736.

[5] Abdullah R, Abdul Rahman MdN, Salleh MohdR. A systematic approach to model human system in cellular manufacturing. Journal of Advanced Mechanical Design, Systems, and Manufacturing 2019;13:JAMDSM0001–JAMDSM0001. https://doi.org/10.1299/jamdsm.2019jamdsm0001.

[6] Rude DJ, Adams S, Beling PA. Task recognition from joint tracking data in an operational manufacturing cell. Journal of Intelligent Manufacturing 2018;29:1203–17. https://doi.org/10.1007/s10845-015-1168-8.

[7] Jiang Q, Liu M, Wang X, Ge M, Lin L. Human motion segmentation and recognition using machine vision for mechanical assembly operation. SpringerPlus 2016;5. https://doi.org/10.1186/s40064-016-3279-x.

[8] Wang J, Ma Y, Zhang L, Gao RX, Wu D. Deep learning for smart manufacturing: Methods and applications. Journal of Manufacturing Systems 2018;48:144–56. https://doi.org/10.1016/j.jmsy.2018.01.003.

[9] Zhu F, Shao L, Xie J, Fang Y. From handcrafted to learned representations for human action recognition: A survey. Image and Vision Computing 2016;55:42–52. https://doi.org/10.1016/j.imavis.2016.06.007.

[10] Zhang S, Wei Z, Nie J, Huang L, Wang S, Li Z. A Review on Human Activity Recognition Using Vision-Based Method. Journal of Healthcare Engineering 2017;2017:1–31. https://doi.org/10.1155/2017/3090343.

[11] Saif S, Tehseen S, Kausar S. A Survey of the Techniques for The Identification and Classification of Human Actions from Visual Data. Sensors 2018;18:3979. https://doi.org/10.3390/s18113979.

[12] Chawky BS, Elons AS, Ali A, Shedeed HA. A Study of Action Recognition Problems: Dataset and Architectures Perspectives. In: Hassanien AE, Oliva DA, editors. Advances in Soft Computing and Machine Learning in Image Processing, vol. 730, Cham: Springer International Publishing; 2018, p. 409–42. https://doi.org/10.1007/978-3-319-63754-9_19.

[13] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436–44. https://doi.org/10.1038/nature14539.

[14] Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge, Massachusetts: The MIT Press; 2016.

[15] Wang L, Duan X, Zhang Q, Niu Z, Hua G, Zheng N. Segment-Tube: Spatio-Temporal Action Localization in Untrimmed Videos with Per-Frame Segmentation. Sensors 2018;18:1657. https://doi.org/10.3390/s18051657.

[16] Singh B, Marks TK, Jones M, Tuzel O, Shao M. A Multi-stream Bi-directional Recurrent Neural Network for Fine-Grained Action Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE; 2016, p. 1961–70. https://doi.org/10.1109/CVPR.2016.216.
[17] Idrees H, Zamir AR, Jiang Y-G, Gorban A, Laptev I, Sukthankar R, et al. The THUMOS challenge on action recognition for videos "in the wild." Computer Vision and Image Understanding 2017;155:1–23. https://doi.org/10.1016/j.cviu.2016.10.018.
[18] Makantasis K, Doulamis A, Doulamis N, Psychas K. Deep learning based human behavior recognition in industrial workflows. 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA: IEEE; 2016, p. 1609–13. https://doi.org/10.1109/ICIP.2016.7532630.
[19] Donahue J, Hendricks LA, Rohrbach M, Venugopalan S, Guadarrama S, Saenko K, et al. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. IEEE Transactions on Pattern Analysis and Machine Intelligence 2017;39:677–91. https://doi.org/10.1109/TPAMI.2016.2599174.
[20] Tkachenko D. Human action recognition using fusion of modern deep convolutional and recurrent neural networks. EasyChair; 2018. https://doi.org/10.29007/wj5t.
[21] Bilen H, Fernando B, Gavves E, Vedaldi A. Action Recognition with Dynamic Image Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 2018;40:2799–813. https://doi.org/10.1109/TPAMI.2017.2769085.
[22] Cai Z, Han J, Liu L, Shao L. RGB-D datasets using microsoft kinect or similar sensors: a survey. Multimedia Tools and Applications 2017;76:4313–55. https://doi.org/10.1007/s11042-016-3374-6.
[23] Li S-J, AbuFarha Y, Liu Y, Cheng M-M, Gall J. MS-TCN++: Multi-Stage Temporal Convolutional Network for Action Segmentation. IEEE Trans Pattern Anal Mach Intell 2020:1–1. https://doi.org/10.1109/TPAMI.2020.3021756.
[24] Lea C, Flynn MD, Vidal R, Reiter A, Hager GD. Temporal Convolutional Networks for Action Segmentation and Detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI: IEEE; 2017, p. 1003–12. https://doi.org/10.1109/CVPR.2017.113.
[25] Voulodimos A, Doulamis N, Doulamis A, Lalos C, Stentoumis C. Human tracking driven activity recognition in video streams. 2016 IEEE International Conference on Imaging Systems and Techniques (IST), Chania, Greece: IEEE; 2016, p. 554–9. https://doi.org/10.1109/IST.2016.7738287.
[26] Arbab-Zavar B, Carter JN, Nixon MS. On hierarchical modelling of motion for workflow analysis from overhead view. Machine Vision and Applications 2014;25:345–59. https://doi.org/10.1007/s00138-013-0528-7.
[27] Voulodimos A, Kosmopoulos D, Vasileiou G, Sardis E, Anagnostopoulos V, Lalos C, et al. A Threefold Dataset for Activity and Workflow Recognition in Complex Industrial Environments. IEEE Multimedia 2012;19:42–52. https://doi.org/10.1109/MMUL.2012.31.
[28] Taboršak D. Work Study 1994.
[29] Chan AHS, Hoffmann ER, Chung CMW. Subjective estimates of times for assembly work. International Journal of Industrial Ergonomics 2017;61:149–55. https://doi.org/10.1016/j.ergon.2017.05.017.
[30] Taylor FW. Scientific management. Routledge; 2004.
[31] Alkan B, Vera D, Ahmad M, Ahmad B, Harrison R. A Model for Complexity Assessment in Manual Assembly Operations Through Predetermined Motion Time Systems. Procedia CIRP 2016;44:429–34. https://doi.org/10.1016/j.procir.2016.02.111.

[32] Genaidy AM, Agrawal A, Mital A. Computerized predetermined motion-time systems in manufacturing industries. Computers & Industrial Engineering 1990;18:571–84. https://doi.org/10.1016/0360-8352(90)90016-F.
[33] Maynard HB, Stegemerten GJ, Schwab JL. Methods-time measurement. 1948.
[34] Di Gironimo G, Di Martino C, Lanzotti A, Marzano A, Russo G. Improving MTM-UAS to predetermine automotive maintenance times. Int J Interact Des Manuf 2012;6:265–73. https://doi.org/10.1007/s12008-012-0158-8.
[35] Luxhoj JT, Giacomelli GA. Comparison of Labour Standards for a Greenhouse Tomato Production System: A Case Study. Int Jrnl of Op & Prod Mnagemnt 1990;10:38–49. https://doi.org/10.1108/01443579010001591.
[36] Karim ANM, Tuan ST, Emrul Kays HM. Assembly line productivity improvement as re-engineered by MOST. IJPPM 2016;65:977–94. https://doi.org/10.1108/IJPPM-11-2015-0169.
[37] Boden MA. Mind as machine 1. 1. Oxford: Clarendon Press; 2008.
[38] Changizi M. The vision revolution: How the latest research overturns everything we thought we knew about human vision. Benbella books; 2010.
[39] Moravec H. Locomotion, vision and intelligence 1984.
[40] Szeliski R. Computer vision: algorithms and applications. London: Springer; 2011.
[41] Prince SJ. Computer vision: models, learning, and inference. Cambridge University Press; 2012.
[42] Gonzales RC, Woods RE. Digital image processing. Prentice hall New Jersey; 2002.
[43] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE; 2016, p. 770–8. https://doi.org/10.1109/CVPR.2016.90.
[44] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE international conference on computer vision, 2015, p. 1026–34.
[45] Szegedy C, Ioffe S, Vanhoucke V, Alemi A. Inception-v4, inception-resnet and the impact of residual connections on learning. Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, 2017.
[46] Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning Deep Features for Discriminative Localization. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE; 2016, p. 2921–9. https://doi.org/10.1109/CVPR.2016.319.
[47] Tychsen-Smith L, Petersson L. Improving Object Localization with Fitness NMS and Bounded IoU Loss. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT: IEEE; 2018, p. 6877–85. https://doi.org/10.1109/CVPR.2018.00719.
[48] Kudo Y, Aoki Y. Dilated convolutions for image classification and object localization. 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), Nagoya, Japan: IEEE; 2017, p. 452–5. https://doi.org/10.23919/MVA.2017.7986898.
[49] Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE; 2016, p. 779–88. https://doi.org/10.1109/CVPR.2016.91.
[50] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans Pattern Anal Mach Intell 2017;39:1137–49. https://doi.org/10.1109/TPAMI.2016.2577031.
[51] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, et al. SSD: Single Shot MultiBox Detector. In: Leibe B, Matas J, Sebe N, Welling M, editors. Computer Vision

– ECCV 2016, vol. 9905, Cham: Springer International Publishing; 2016, p. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2.
[52] Shelhamer E, Long J, Darrell T. Fully Convolutional Networks for Semantic Segmentation. IEEE Trans Pattern Anal Mach Intell 2017;39:640–51. https://doi.org/10.1109/TPAMI.2016.2572683.
[53] Jegou S, Drozdzal M, Vazquez D, Romero A, Bengio Y. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA: IEEE; 2017, p. 1175–83. https://doi.org/10.1109/CVPRW.2017.156.
[54] Yang X, Li X, Ye Y, Zhang X, Zhang H, Huang X, et al. Road Detection via Deep Residual Dense U-Net. 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary: IEEE; 2019, p. 1–7. https://doi.org/10.1109/IJCNN.2019.8851728.
[55] Toshev A, Szegedy C. DeepPose: Human Pose Estimation via Deep Neural Networks. 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA: IEEE; 2014, p. 1653–60. https://doi.org/10.1109/CVPR.2014.214.
[56] Insafutdinov E, Pishchulin L, Andres B, Andriluka M, Schiele B. DeeperCut: A Deeper, Stronger, and Faster Multi-person Pose Estimation Model. In: Leibe B, Matas J, Sebe N, Welling M, editors. Computer Vision – ECCV 2016, vol. 9910, Cham: Springer International Publishing; 2016, p. 34–50. https://doi.org/10.1007/978-3-319-46466-4_3.
[57] Mehta D, Sridhar S, Sotnychenko O, Rhodin H, Shafiei M, Seidel H-P, et al. VNect: real-time 3D human pose estimation with a single RGB camera. ACM Trans Graph 2017;36:1–14. https://doi.org/10.1145/3072959.3073596.
[58] Chen L, Zhang H, Xiao J, Nie L, Shao J, Liu W, et al. SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI: IEEE; 2017, p. 6298–306. https://doi.org/10.1109/CVPR.2017.667.
[59] Anderson P, He X, Buehler C, Teney D, Johnson M, Gould S, et al. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT: IEEE; 2018, p. 6077–86. https://doi.org/10.1109/CVPR.2018.00636.
[60] Doulamis N. Adaptable deep learning structures for object labeling/tracking under dynamic visual environments. Multimedia Tools and Applications 2018;77:9651–89. https://doi.org/10.1007/s11042-017-5349-7.
[61] Liu L, Liu Y, Zhang J. Learning-Based Hand Motion Capture and Understanding in Assembly Process. IEEE Transactions on Industrial Electronics 2019;66:9703–12. https://doi.org/10.1109/TIE.2018.2884206.
[62] Ristani E, Solera F, Zou R, Cucchiara R, Tomasi C. Performance Measures and a Data Set for Multi-target, Multi-camera Tracking. In: Hua G, Jégou H, editors. Computer Vision – ECCV 2016 Workshops, vol. 9914, Cham: Springer International Publishing; 2016, p. 17–35. https://doi.org/10.1007/978-3-319-48881-3_2.
[63] Fan K, Joung C, Baek S. Sequence-to-Sequence Video Prediction by Learning Hierarchical Representations. Applied Sciences 2020;10:8288. https://doi.org/10.3390/app10228288.
[64] Xu J, Ni B, Yang X. Progressive Multi-granularity Analysis for Video Prediction. Int J Comput Vis 2020. https://doi.org/10.1007/s11263-020-01389-w.
[65] Zhao Y, Dou Y. Pose-Forecasting Aided Human Video Prediction With Graph Convolutional Networks. IEEE Access 2020;8:147256–64. https://doi.org/10.1109/ACCESS.2020.2995383.

[66] Lee D, Oh YJ, Lee I-K. Future-Frame Prediction for Fast-Moving Objects with Motion Blur. Sensors 2020;20:4394. https://doi.org/10.3390/s20164394.
[67] Costante G, Mancini M, Valigi P, Ciarfuglia TA. Exploring Representation Learning With CNNs for Frame-to-Frame Ego-Motion Estimation. IEEE Robot Autom Lett 2016;1:18–25. https://doi.org/10.1109/LRA.2015.2505717.
[68] Zhu Y, Newsam S. DenseNet for dense flow. 2017 IEEE International Conference on Image Processing (ICIP), Beijing: IEEE; 2017, p. 790–4. https://doi.org/10.1109/ICIP.2017.8296389.
[69] Herath S, Harandi M, Porikli F. Going deeper into action recognition: A survey. Image and Vision Computing 2017;60:4–21. https://doi.org/10.1016/j.imavis.2017.01.010.
[70] Farha YA, Gall J. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, p. 3575–84.
[71] Buch S, Escorcia V, Shen C, Ghanem B, Niebles JC. SST: Single-Stream Temporal Action Proposals. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI: IEEE; 2017, p. 6373–82. https://doi.org/10.1109/CVPR.2017.675.
[72] Xu H, Das A, Saenko K. Two-Stream Region Convolutional 3D Network for Temporal Activity Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 2019;41:2319–32. https://doi.org/10.1109/TPAMI.2019.2921539.
[73] O'Mahony N, Campbell S, Carvalho A, Harapanahalli S, Hernandez GV, Krpalkova L, et al. Deep Learning vs. Traditional Computer Vision. In: Arai K, Kapoor S, editors. Advances in Computer Vision, vol. 943, Cham: Springer International Publishing; 2020, p. 128–44. https://doi.org/10.1007/978-3-030-17795-9_10.
[74] Lowe DG. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 2004;60:91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.
[75] Bay H, Tuytelaars T, Van Gool L. SURF: Speeded Up Robust Features. In: Leonardis A, Bischof H, Pinz A, editors. Computer Vision – ECCV 2006, vol. 3951, Berlin, Heidelberg: Springer Berlin Heidelberg; 2006, p. 404–17. https://doi.org/10.1007/11744023_32.
[76] Dalal N, Triggs B. Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, San Diego, CA, USA: IEEE; 2005, p. 886–93. https://doi.org/10.1109/CVPR.2005.177.
[77] Planche B, Andres E. Hands-on computer vision with TensorFlow 2 leverage deep learning to create powerful image processing apps with TensorFlow 2.0 and Keras. 2019.
[78] Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS. Deep learning for visual understanding: A review. Neurocomputing 2016;187:27–48. https://doi.org/10.1016/j.neucom.2015.09.116.
[79] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM 2017;60:84–90. https://doi.org/10.1145/3065386.
[80] Deng J, Dong W, Socher R, Li L-J, Kai Li, Li Fei-Fei. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL: IEEE; 2009, p. 248–55. https://doi.org/10.1109/CVPR.2009.5206848.
[81] Aggarwal CC. Neural Networks and Deep Learning: A Textbook. Cham: Springer International Publishing; 2018. https://doi.org/10.1007/978-3-319-94463-0.

[82] Deisenroth MP, Faisal AA, Ong CS. Mathematics for machine learning. Cambridge University Press; 2020.
[83] Bishop CM. Pattern recognition and machine learning. New York: Springer; 2006.
[84] Friedman J, Hastie T, Tibshirani R, others. The elements of statistical learning. vol. 1. Springer series in statistics New York; 2001.
[85] Domingos P. A few useful things to know about machine learning. Commun ACM 2012;55:78–87. https://doi.org/10.1145/2347736.2347755.
[86] McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 1943;5:115–33. https://doi.org/10.1007/BF02478259.
[87] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review 1958;65:386–408. https://doi.org/10.1037/h0042519.
[88] Widrow B, Hoff ME. Adaptive switching circuits. Stanford Univ Ca Stanford Electronics Labs; 1960.
[89] McClelland JL, Rumelhart DE, Group PR, others. Parallel distributed processing. vol. 2. MIT press Cambridge, MA; 1986.
[90] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature 1986;323:533–6.
[91] Cybenko G. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems 1989;2:303–14.
[92] Choromanska A, Henaff M, Mathieu M, Arous GB, LeCun Y. The loss surfaces of multilayer networks. Artificial intelligence and statistics, PMLR; 2015, p. 192–204.
[93] Bellman R. Dynamic programming. Science 1966;153:34–7.
[94] Bengio Y, Lee D-H, Bornschein J, Mesnard T, Lin Z. Towards biologically plausible deep learning. ArXiv Preprint ArXiv:150204156 2015.
[95] Lee D-H, Zhang S, Fischer A, Bengio Y. Difference target propagation. Joint european conference on machine learning and knowledge discovery in databases, Springer; 2015, p. 498–515.
[96] Choromanska A, Cowen B, Kumaravel S, Luss R, Rigotti M, Rish I, et al. Beyond backprop: Online alternating minimization with auxiliary variables. International Conference on Machine Learning, PMLR; 2019, p. 1193–202.
[97] Jaderberg M, Czarnecki WM, Osindero S, Vinyals O, Graves A, Silver D, et al. Decoupled neural interfaces using synthetic gradients. International Conference on Machine Learning, PMLR; 2017, p. 1627–35.
[98] Sutskever I, Martens J, Dahl G, Hinton G. On the importance of initialization and momentum in deep learning. International conference on machine learning, PMLR; 2013, p. 1139–47.
[99] Kingma DP, Ba J. Adam: A method for stochastic optimization. ArXiv Preprint ArXiv:14126980 2014.
[100] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. Proceedings of the fourteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings; 2011, p. 315–23.
[101] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. International conference on machine learning, PMLR; 2015, p. 448–56.
[102] He K, Zhang X, Ren S, Sun J. Identity Mappings in Deep Residual Networks. In: Leibe B, Matas J, Sebe N, Welling M, editors. Computer Vision – ECCV 2016, vol. 9908, Cham: Springer International Publishing; 2016, p. 630–45. https://doi.org/10.1007/978-3-319-46493-0_38.

[103] Hubel DH, Wiesel TN. Receptive fields of single neurones in the cat's striate cortex. The Journal of Physiology 1959;148:574–91. https://doi.org/10.1113/jphysiol.1959.sp006308.
[104] Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. International conference on machine learning, PMLR; 2013, p. 1310–8.
[105] Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation 1997;9:1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.
[106] Lipton ZC, Berkowitz J, Elkan C. A critical review of recurrent neural networks for sequence learning. ArXiv Preprint ArXiv:150600019 2015.
[107] Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar: Association for Computational Linguistics; 2014, p. 1724–34. https://doi.org/10.3115/v1/D14-1179.
[108] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. California Univ San Diego La Jolla Inst for Cognitive Science; 1985.
[109] Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Trans Signal Process 1997;45:2673–81. https://doi.org/10.1109/78.650093.
[110] Clevert D-A, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (elus). ArXiv Preprint ArXiv:151107289 2015.
[111] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings; 2010, p. 249–56.
[112] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 2014;15:1929–58.
[113] Smith LN. A disciplined approach to neural network hyper-parameters: Part 1– learning rate, batch size, momentum, and weight decay. ArXiv Preprint ArXiv:180309820 2018.
[114] Swersky K, Snoek J, Adams RP. Freeze-thaw bayesian optimization. ArXiv Preprint ArXiv:14063896 2014.
[115] Dosovitskiy A, Fischer P, Ilg E, Hausser P, Hazirbas C, Golkov V, et al. Flownet: Learning optical flow with convolutional networks. Proceedings of the IEEE international conference on computer vision, 2015, p. 2758–66.
[116] Horn BKP, Schunck BG. Determining optical flow. Artificial Intelligence 1981;17:185–203. https://doi.org/10.1016/0004-3702(81)90024-2.
[117] Ma S, Sigal L, Sclaroff S. Learning Activity Progression in LSTMs for Activity Detection and Early Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE; 2016, p. 1942–50. https://doi.org/10.1109/CVPR.2016.214.
[118] Montes A, Salvador A, Giró-i-Nieto X. Temporal Activity Detection in Untrimmed Videos with Recurrent Neural Networks 2016.
[119] Ding L, Xu C. TricorNet: A Hybrid Temporal Convolutional and Recurrent Network for Video Action Segmentation. 2017.
[120] Yuan Z, Stroud JC, Lu T, Deng J. Temporal Action Localization by Structured Maximal Sums. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI: IEEE; 2017, p. 3215–23. https://doi.org/10.1109/CVPR.2017.342.

[121] Shou Z, Chan J, Zareian A, Miyazawa K, Chang S-F. CDC: Convolutional-De-Convolutional Networks for Precise Temporal Action Localization in Untrimmed Videos 2017.
[122] Jin Y, Dou Q, Chen H, Yu L, Qin J, Fu C-W, et al. SV-RCNet: Workflow Recognition From Surgical Videos Using Recurrent Convolutional Network. IEEE Transactions on Medical Imaging 2018;37:1114–26. https://doi.org/10.1109/TMI.2017.2787657.
[123] Bai R, Zhao Q, Zhou S, Li Y, Zhao X, Wang J. Continuous Action Recognition and Segmentation in Untrimmed Videos. 2018 24th International Conference on Pattern Recognition (ICPR), Beijing: IEEE; 2018, p. 2534–9. https://doi.org/10.1109/ICPR.2018.8546019.
[124] Yang H, He X, Porikli F. Instance-Aware Detailed Action Labeling in Videos. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV: IEEE; 2018, p. 1577–86. https://doi.org/10.1109/WACV.2018.00175.
[125] Lei P, Todorovic S. Temporal Deformable Residual Networks for Action Segmentation in Videos. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT: IEEE; 2018, p. 6742–51. https://doi.org/10.1109/CVPR.2018.00705.
[126] Bodenstedt S, Rivoir D, Jenke A, Wagner M, Breucha M, Müller-Stich B, et al. Active learning using deep Bayesian networks for surgical workflow analysis. International Journal of Computer Assisted Radiology and Surgery 2019;14:1079–87. https://doi.org/10.1007/s11548-019-01963-9.
[127] Yang K, Shen X, Qiao P, Li S, Li D, Dou Y. Exploring frame segmentation networks for temporal action localization. Journal of Visual Communication and Image Representation 2019;61:296–302. https://doi.org/10.1016/j.jvcir.2019.02.003.
[128] Shou Z, Wang D, Chang S-F. Action Temporal Localization in Untrimmed Videos via Multi-stage CNNs 2016.
[129] Escorcia V, Caba Heilbron F, Niebles JC, Ghanem B. DAPs: Deep Action Proposals for Action Understanding. In: Leibe B, Matas J, Sebe N, Welling M, editors. Computer Vision – ECCV 2016, vol. 9907, Cham: Springer International Publishing; 2016, p. 768–84. https://doi.org/10.1007/978-3-319-46487-9_47.
[130] Dai X, Singh B, Zhang G, Davis LS, Chen YQ. Temporal Context Network for Activity Localization in Videos. 2017 IEEE International Conference on Computer Vision (ICCV), Venice: IEEE; 2017, p. 5727–36. https://doi.org/10.1109/ICCV.2017.610.
[131] Buch S, Escorcia V, Ghanem B, Niebles JC. End-to-End, Single-Stream Temporal Action Detection in Untrimmed Videos. Procedings of the British Machine Vision Conference 2017, London, UK: British Machine Vision Association; 2017. https://doi.org/10.5244/C.31.93.
[132] Gao J, Yang Z, Sun C, Chen K, Nevatia R. TURN TAP: Temporal Unit Regression Network for Temporal Action Proposals. 2017 IEEE International Conference on Computer Vision (ICCV), Venice: IEEE; 2017, p. 3648–56. https://doi.org/10.1109/ICCV.2017.392.
[133] Gao J, Yang Z, Nevatia R. Cascaded Boundary Regression for Temporal Action Detection. Procedings of the British Machine Vision Conference 2017, London, UK: British Machine Vision Association; 2017. https://doi.org/10.5244/C.31.52.
[134] Lin T, Zhao X, Shou Z. Single Shot Temporal Action Detection. Proceedings of the 2017 ACM on Multimedia Conference - MM '17, Mountain View, California, USA: ACM Press; 2017, p. 988–96. https://doi.org/10.1145/3123266.3123343.

[135] Xu H, Das A, Saenko K. R-C3D: Region Convolutional 3D Network for Temporal Activity Detection. 2017 IEEE International Conference on Computer Vision (ICCV), Venice: IEEE; 2017, p. 5794–803. https://doi.org/10.1109/ICCV.2017.617.
[136] Zhao Y, Xiong Y, Wang L, Wu Z, Tang X, Lin D. Temporal Action Detection with Structured Segment Networks. 2017 IEEE International Conference on Computer Vision (ICCV), Venice: IEEE; 2017, p. 2933–42. https://doi.org/10.1109/ICCV.2017.317.
[137] Yao G, Lei T, Liu X, Jiang P. Temporal Action Detection in Untrimmed Videos from Fine to Coarse Granularity. Applied Sciences 2018;8:1924. https://doi.org/10.3390/app8101924.
[138] Chao Y-W, Vijayanarasimhan S, Seybold B, Ross DA, Deng J, Sukthankar R. Rethinking the Faster R-CNN Architecture for Temporal Action Localization. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT: IEEE; 2018, p. 1130–9. https://doi.org/10.1109/CVPR.2018.00124.
[139] Huang Y, Dai Q, Lu Y. Decoupling Localization and Classification in Single Shot Temporal Action Detection. 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China: IEEE; 2019, p. 1288–93. https://doi.org/10.1109/ICME.2019.00224.
[140] Liu Z, Wang Z, Zhao Y, Tian Y. SMC: Single-Stage Multi-location Convolutional Network for Temporal Action Detection. In: Jawahar CV, Li H, Mori G, Schindler K, editors. Computer Vision – ACCV 2018, vol. 11362, Cham: Springer International Publishing; 2019, p. 179–95. https://doi.org/10.1007/978-3-030-20890-5_12.
[141] Wang Z, Chen K, Zhang M, He P, Wang Y, Zhu P, et al. Multi-scale aggregation network for temporal action proposals. Pattern Recognition Letters 2019;122:60–5. https://doi.org/10.1016/j.patrec.2019.02.007.
[142] Veres G, Grabner H, Middleton L, Van Gool L. Automatic Workflow Monitoring in Industrial Environments. In: Kimmel R, Klette R, Sugimoto A, editors. Computer Vision – ACCV 2010, vol. 6492, Berlin, Heidelberg: Springer Berlin Heidelberg; 2011, p. 200–13. https://doi.org/10.1007/978-3-642-19315-6_16.
[143] Jaeger H. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. Science 2004;304:78–80. https://doi.org/10.1126/science.1091277.
[144] Voulodimos A, Kosmopoulos D, Veres G, Grabner H, Van Gool L, Varvarigou T. Online classification of visual tasks for industrial workflow monitoring. Neural Networks 2011;24:852–60. https://doi.org/10.1016/j.neunet.2011.06.001.
[145] Voulodimos AS, Kosmopoulos DI, Doulamis ND, Varvarigou TA. A top-down event-driven approach for concurrent activity recognition. Multimedia Tools and Applications 2014;69:293–311. https://doi.org/10.1007/s11042-012-0993-4.
[146] Kosmopoulos D, Chatzis S. Robust Visual Behavior Recognition. IEEE Signal Process Mag 2010;27:34–45. https://doi.org/10.1109/MSP.2010.937392.
[147] Davis JW. Hierarchical motion history images for recognizing human motion. Proceedings IEEE Workshop on Detection and Recognition of Events in Video, Vancouver, BC, Canada: IEEE Comput. Soc; 2001, p. 39–46. https://doi.org/10.1109/EVENT.2001.938864.
[148] Heilbron FC, Escorcia V, Ghanem B, Niebles JC. ActivityNet: A large-scale video benchmark for human activity understanding. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA: IEEE; 2015, p. 961–70. https://doi.org/10.1109/CVPR.2015.7298698.
[149] Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L. Large-Scale Video Classification with Convolutional Neural Networks. 2014 IEEE Conference on

Computer Vision and Pattern Recognition, Columbus, OH, USA: IEEE; 2014, p. 1725–32. https://doi.org/10.1109/CVPR.2014.223.
[150] Womack JP, Jones DT. Lean Thinking—Banish Waste and Create Wealth in your Corporation. Journal of the Operational Research Society 1997;48:1148–1148. https://doi.org/10.1057/palgrave.jors.2600967.
[151] Shapiro SS, Wilk MB. An analysis of variance test for normality (complete samples). Biometrika 1965;52:591–611. https://doi.org/10.1093/biomet/52.3-4.591.
[152] Kruskal WH, Wallis WA. Use of Ranks in One-Criterion Variance Analysis. Journal of the American Statistical Association 1952;47:583–621. https://doi.org/10.1080/01621459.1952.10483441.
[153] Mann HB, Whitney DR. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. Ann Math Statist 1947;18:50–60. https://doi.org/10.1214/aoms/1177730491.
[154] Dunn OJ. Multiple Comparisons among Means. Journal of the American Statistical Association 1961;56:52–64. https://doi.org/10.1080/01621459.1961.10482090.
[155] Tran D, Bourdev L, Fergus R, Torresani L, Paluri M. Learning Spatiotemporal Features with 3D Convolutional Networks. 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile: IEEE; 2015, p. 4489–97. https://doi.org/10.1109/ICCV.2015.510.
[156] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. Tensorflow: A system for large-scale machine learning. 12th ${$USENIX$}$ symposium on operating systems design and implementation (${$OSDI$}$ 16), 2016, p. 265–83.
[157] Oquab M, Bottou L, Laptev I, Sivic J. Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA: IEEE; 2014, p. 1717–24. https://doi.org/10.1109/CVPR.2014.222.
[158] Razavian AS, Azizpour H, Sullivan J, Carlsson S. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA: IEEE; 2014, p. 512–9. https://doi.org/10.1109/CVPRW.2014.131.

ŽIVOTOPIS

Mihael Gudlin rođen je 31. listopada 1987. u Koprivnici. Osnovnu školu završavao je u Delovima i Novigradu Podravskom, nakon koje polazi srednju *Obrtničku školu*, zanimanje elektrotehničar, u Koprivnici. 2006. godine upisuje studij strojarstva na *Fakultetu strojarstva i brodogradnje*. Diplomirao je 2011. na smjeru *Industrijsko inženjerstvo i menadžment*.

2012. zapošljava se u *Zagrebačkom holdingu d.o.o., Podružnica Čistoća* gdje radi na poslovima tehnologa u *Sektoru održavanja*. Sudjeluje na projektima optimizacije poslovnih procesa održavanja te izrade procedura i postupaka prema ISO standardima. Vodio je projekt optimizacije i organizacije aktivnosti servisa vozila *Podružnice*. U akademskoj godini 2013./2014. upisuje poslijediplomski studij strojarstva na *Fakultetu strojarstva i brodogradnje,* smjer *Industrijsko inženjerstvo i menadžment*. Od veljače 2015. godine zaposlen je kao asistent na *Katedri za upravljanje proizvodnjom, Zavoda za industrijsko inženjerstvo*.

Sudjeluje u izvođenju nastave iz kolegija na *Katedri za upravljanje proizvodnjom*. Uz nastavne aktivnosti, sudjelovao je na organizaciji znanstvenih i stručnih konferencija (GALP, LSS). Radio je kao suradnik na istraživačko-razvojnim projektima i projektima stručnog savjetovanja. Do sada je kao autor ili koautor objavio 8 znanstvenih radova u časopisima te zbornicima radova u zemlji i inozemstvu. Područja kojima se bavi su: optimizacija procesa, strojno učenje, operacijska istraživanja te primjena računalnog vida u industrijskom inženjerstvu.

Aktivno se služi engleskim jezikom, a pasivno njemačkim.

Oženjen je i otac dvoje djece.

SHORT BIOGRAPHY

Mihael Gudlin was born on October 31, 1987 in Koprivnica. He completed elementary school education in Delovi and Novigrad Podravski, after which he attended high school *Obrtnička škola* in Koprivnica. In 2006 he enrolled at the *Faculty of Mechanical Engineering and Naval Architecture*. He graduated in 2011 with a master's degree in *Industrial Engineering and Management*.

In 2012. he was employed by *Zagrebački holding d.o.o., Podružnica Čistoća* where he worked as a technologist in the maintenance sector. He participated in projects focused on the optimization of maintenance processes and the development of procedures according to ISO standards. He led a project with the aim of optimization and organization of vehicle service in *Podružnica Čistoća*. In the academic year 2013/2014 he enrolled in a postgraduate study in mechanical engineering at the *Faculty of Mechanical Engineering and Naval Architecture*, majoring in *Industrial Engineering and Management*. Since February 2015, he has been employed as a teaching and research assistant at the *Chair for Production Management* at the *Department of Industrial Engineering.*

He is involved in the presentation of the course lectures at the *Chair for Production Management.* In addition to his teaching activities, he was also a part of the team organizing scientific and professional conferences (GALP, LSS). He worked as a project associate on several scientific and professional projects. So far, as an author or co-author, he has published 8 scientific papers in journals and proceedings in the country and abroad. His research interests are process optimization, machine learning, operations research and application of computer vision in industrial engineering.

He speaks and writes English and is a passive user of German.

He is married and father of two children.