

Prepoznavanje ljudskih aktivnosti u proizvodnim sustavima upotrebom umjetne inteligencije

Leljak, Mario

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:650299>

Rights / Prava: [Attribution 3.0 Unported/Imenovanje 3.0](#)

Download date / Datum preuzimanja: **2024-07-12**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mario Lejak

Zagreb, 2021. godina.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:

doc. dr. sc. Miro Hegedić, mag. ing.

Student:

Mario Leljak

Zagreb, 2021. godina.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru doc. dr. sc. Miri Hegediću na pruženoj pomoći i savjetima prilikom pisanja ovog rada. Također, zahvaljujem se dr. sc. Mihaelu Gudlinu na velikoj pomoći prilikom izrade praktičnog dijela rada.

Posebna zahvala mojim roditeljima, sestri i baki na potpori tijekom čitavog studiranja.

Zahvaljujem se i svima koji su na bilo koji način bili uz mene tijekom ovog putovanja.

Mario Leļjak



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske radove studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment,
inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa:	602-04/21-6/1
Ur. broj:	15-1703-21

DIPLOMSKI ZADATAK

Student: **MARIO LELJAK** Mat. br.: 0035200827

Naslov rada na hrvatskom jeziku: **Prepoznavanje ljudskih aktivnosti u proizvodnim sustavima upotrebom umjetne inteligencije**

Naslov rada na engleskom jeziku: **Recognition of human activities in production systems using artificial intelligence**

Opis zadatka:

Analiza ljudskih aktivnosti u proizvodnim sustavima predstavlja aktualnu, ali i složenu temu istraživanja u području industrijskog inženjerstva. Iako će sve veća automatizacija i digitalizacija proizvodnih sustava utjecati na smanjenje potrebe za direktnim ljudskim radom u proizvodnim procesima, čovjek će i dalje imati značajnu ulogu u njima. Stoga će i dalje biti potrebno pratiti učinkovitost ljudskog faktora, a što će biti moguće upotrebom video zapisa iz procesa. Umjetna inteligencija se sve više spominje u zadnje vrijeme, ali i pronalazi primjenu u različitim dijelovima ljudskih života. Jedna od primjena je i analiza ljudskih aktivnosti iz video zapisa koja spada u domenu računalnog vida.

U radu je potrebno:

1. Definirati pojam umjetne inteligencije i objasniti njezin razvoj kroz povijest te utjecaj Industrije 4.0 na nju te objasniti osnovne pojmove i alate
2. Prikazati primjere primjene umjetne inteligencije u različitim industrijama i što ona donosi tim industrijama s posebnim naglaskom na proizvodnju
3. Objasniti primjenu umjetne inteligencije u proizvodnim sustavima s fokusom na prepoznavanje ljudskih aktivnosti u procesu
4. Testirati jedan ili više postojećih modela prepoznavanja ljudskih aktivnosti u proizvodnom procesu koristeći se realnim podacima iz proizvodnje
5. Na temelju prethodnih koraka usporediti rezultate i donijeti zaključke.

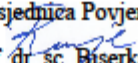
U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:
30. rujna 2021.

Rok predaje rada:
2. prosinca 2021.

Predviđeni datum obrane:
13. prosinca do 17. prosinca 2021.

Zadatak zadao:
doc. dr. sc. Miro Hegedić

Predsjednica Povjerenstva:

prof. dr. sc. Biserka Runje

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	III
POPIS TABLICA.....	IV
SAŽETAK.....	V
SUMMARY	VI
1. UVOD.....	1
2. UMJETNA INTELIGENCIJA	2
2.1. Definiranje umjetne inteligencije.....	2
2.2. Razvoj umjetne inteligencije.....	2
2.3. Umjetna inteligencija u Industriji 4.0	4
2.4. Srodna područja umjetne inteligencije.....	5
2.4.1. Strojno učenje	6
2.4.1.1. Klasifikator najbližih susjeda.....	9
2.4.2. Duboko učenje	11
2.4.2.1. Umjetne neuronske mreže.....	11
2.4.2.2. Konvolucijska neuronska mreža	12
2.4.3. Razlika između strojnog učenja i dubokog učenja	13
2.4.4. Znanost o podacima i robotika.....	15
3. PRIMJENA UMJETNE INTELIGENCIJE	16
3.1. Umjetna inteligencija u životu čovjeka.....	16
3.1.1. Autonomni automobili	16
3.1.2. Preporuka sadržaja	17
3.1.3. Obrada slika i videozapisa	17
3.2. Primjena umjetne inteligencije u industriji	18
3.2.1. Marketing	19
3.2.2. Maloprodaja	19
3.2.3. Lanac opskrbe	19
3.2.4. Kibernetička sigurnost	19
3.2.5. Prediktivno održavanje	19
3.2.6. Prednosti i nedostaci implementacije umjetne inteligencije u poduzeće.....	20
3.3. Umjetna inteligencija u proizvodnji.....	23
3.3.1. Primjena umjetne inteligencije	24
3.3.1.1. Pametno održavanje	24
3.3.1.2. Pобоljšanje kvalitete	25
3.3.1.3. Prilagođavanje tržištu.....	25
3.3.2. Primjeri proizvodnih poduzeća koja su implementirala umjetnu inteligenciju ..	26
3.3.2.1. Siemens	26
3.3.2.2. General Electric	26
3.3.2.3. Fanuc.....	26
3.3.3. Budućnost umjetne inteligencije u proizvodnom sektoru.....	28
4. PREPOZNAVANJE LJUDSKIH AKTIVNOSTI.....	30

4.1. Primjena i aktivnosti prepoznavanja ljudskih aktivnosti	30
4.1.1. Područje primjene	30
4.1.2. Osnovne vrste aktivnosti	30
4.2. Obrada slika i videozapisa	31
4.2.1. Pristup temeljen na dubokom učenju	32
4.2.2. Izazovi u obradi videozapisa	33
4.3. Slučajevi za primjenu prepoznavanja ljudskih aktivnosti u proizvodnji	34
4.3.1. Primjeri prepoznavanja ljudskih aktivnosti u proizvodnji	35
5. MODEL PREPOZNAVANJA LJUDSKIH AKTIVNOSTI U PROIZVODNOM PROCESU	37
5.1. Prikupljanje podataka	37
5.1.1. Izrezivanje videozapisa	38
5.1.2. Označavanje vrsta aktivnosti	39
5.2. Definiranje skupa podataka	39
5.3. Priprema video sličica	41
5.4. Spremanje sličica u Tensorflow Record format	43
5.5. Izvlačenje prostornih značajki	45
5.6. Spremanje značajki u Tensorflow Record format	47
5.6.1. Konkatenacija značajki	49
5.7. Učenje modela	51
5.8. Evaluacija modela	56
5.8.1. Analiza modela	58
6. ZAKLJUČAK	59
LITERATURA	61
PRILOZI	64

POPIS SLIKA

Slika 1 Industrija 4.0 [9].....	4
Slika 2 Dijelovi umjetne inteligencije [12]	6
Slika 3 Tipovi strojnog učenja [11].....	8
Slika 4 Podaci o korisnicima [1]	10
Slika 5 Podaci o Travisu [1].....	10
Slika 6 Usporedba strojnog učenja i dubokog učenja [12].....	14
Slika 7 Autonomni automobil [1].....	16
Slika 8 Umjetna inteligencija u industriji [6]	18
Slika 9 Prednosti umjetne inteligencije [7]	21
Slika 10 Nedostaci umjetne inteligencije [7]	22
Slika 11 Vještine poduzeća za implementaciju umjetne inteligencije [8]	23
Slika 12 Pametno održavanje [4]	24
Slika 13 Umjetna inteligencija u Fanucu [9].....	27
Slika 14 Primjena umjetne inteligencije u budućnosti [27]	28
Slika 15 Otvaranje/zatvaranje vrata [20].....	32
Slika 16 Lokacija prikupljanja podataka [21]	37
Slika 17 Operater O1 montira proizvod T1 [21]	38
Slika 18 Kreiranje oznaka	41
Slika 19 Funkcije za pripremu sličica	42
Slika 20 Skripta za pripremu sličica.....	43
Slika 21 TFRecord kod - dio 1	44
Slika 22 TFRecord kod - dio 2	45
Slika 23 Izvlačenje prostornih značajki	46
Slika 24 TFRecord spremanje značajki – dio 1.....	48
Slika 25 TFRecord spremanje značajki - dio 2	49
Slika 26 Konkatenacija - dio 1	50
Slika 27 Konkatenacija - dio 2	51
Slika 28 Učenje modela.....	53
Slika 29 Priprema modela za učenje	54
Slika 30 Graf učenja modela	56

POPIS TABLICA

Tablica 1 Podjela uzoraka u 3 skupa podataka	40
Tablica 2 Proces učenja modela	55
Tablica 3 Rezultati metrike	57

SAŽETAK

Tema ovog diplomskog rada je primjena umjetne inteligencije u proizvodnim sustavima. Industrija 4.0 dovela je revoluciju u industriju proizvodnje, a umjetna inteligencija ima potencijal u budućnosti transformirati postupak proizvodnje. Poseban naglasak stavljen je na prepoznavanje ljudskih aktivnosti u proizvodnim procesima primjenom strojnog učenja i dubokog učenja. Takav postupak detaljno je objašnjen u praktičnom dijelu rada gdje se testira postojeći model korištenjem realnih podataka iz proizvodnje.

Ključne riječi: umjetna inteligencija, industrija 4.0, strojno učenje, duboko učenje, prepoznavanje ljudskih aktivnosti

SUMMARY

The topic of this master thesis is the application of artificial intelligence in production systems. Industry 4.0 has brought a revolution to the manufacturing industry, and artificial intelligence has the potential to transform the manufacturing process in the future. A special focus is on recognizing human activities in production processes using machine learning and deep learning tools. Such a procedure is explained in detail in the practical part of the thesis where the existing model is tested using real data from production.

Key words: artificial intelligence, industry 4.0, machine learning, deep learning, human activity recognition

1. UVOD

Kada bi se skupini ljudi različite dobi, spola i životnih interesa postavilo pitanje jesu li ikad čuli za pojam umjetna inteligencija, odgovor većine njih bio bi potvrđan. Umjetna inteligencija svakodnevno je tema razgovora u raznim medijima, od televizije do Interneta pa je tako postala poznata široj javnosti. No, kada bi se od iste skupine ljudi tražilo da objasne što je zapravo umjetna inteligencija, rijetko tko bi znao dati detaljan i točan odgovor. Za neke umjetna inteligencija podrazumijeva umjetne oblike života koji mogu nadmašiti ljudsku inteligenciju, a drugi pak smatraju da se umjetnom inteligencijom može nazvati gotovo svaka vrsta tehnologije za obradu podataka. Svoju popularnost u medijima umjetna inteligencija djelomično duguje činjenici da su ljudi taj naziv počeli upotrebljavati za stvari koje su se nekada nazivale drugačije. Danas se umjetnom inteligencijom naziva gotovo sve, od statistike i poslovne analitike do ručno kodiranih pravila zaključivanja u obliku iskaza „ako... onda”. Zašto je tome tako? Jedan od razloga je taj što nema jasne definicije za umjetnu inteligenciju. Točnije, ona se stalno mijenja jer se određene teme prestaju svrstavati u područje umjetne inteligencije, a nove teme se pojavljuju.

Kako umjetna inteligencija sve više pronalazi primjenu u različitim dijelovima ljudskog života, u ovom radu nastojati će se pobliže objasniti njeno značenje kroz dostupne definicije, razvoj kroz povijest te primjenu u industriji s posebnim naglaskom na industriju proizvodnje. Velika pažnja u ovom radu staviti će se na sve veći utjecaj umjetne inteligencije na ljudske aktivnosti u proizvodnim procesima gdje će se na jednom stvarnom primjeru iz proizvodnje prikazati na koji način se pomoću umjetne inteligencije mogu prepoznavati ljudske aktivnosti.

2. UMJETNA INTELIGENCIJA

2.1. Definiranje umjetne inteligencije

„Elementi Umjetne Inteligencije“ (engl. *Elements of AI*) naziv je za niz besplatnih online tečajeva na temu umjetne inteligencije Reaktora i Sveučilišta u Helsinkiju u kojima se umjetna inteligencija definira kao „strojevi koji oponašaju inteligentno ljudsko ponašanje“, odnosno „prilagodljivi autonomni sustavi“ [1]. Spajanjem tih dviju definicija može se doći do nove koja bi za umjetnu inteligenciju kazala kako se bavi metodama i tehnologijama koje omogućuju računalu izvršavanje onih zadataka koji bi zahtijevali inteligenciju da ih obavljaju ljudska bića. Ona se često koristi za opisivanje računalnih sustava koji izvršavaju zadatke nakon što su obučeni velikom količinom podataka i koji nakon toga, često zajedno s drugim metodama, donose odluke izvedene iz podataka koji su im već poznati. Ovisno o kvaliteti i kvantiteti podataka za učenje, sustav umjetne inteligencije može izvršiti ono što smatra „ispravnom“ radnjom. Uz pomoć algoritma za učenje, sustavi umjetne inteligencije mogu nastaviti učenje tijekom tekućih operacija, putem kojih se naučeni modeli optimiziraju i proširuju baze podataka i znanja [2].

Razlog sve veće popularizacije umjetne inteligencije leži u njezinoj temeljnoj tehnološkoj funkciji, odnosno da se u načelu može koristiti za provedbu svake formalizacije ljudskog i racionalnog mišljenja i djelovanja, što znači učenje, planiranje i rješavanje problema. U to su uključeni postupci kao što su prepoznavanje slike i glasa, stjecanje znanja, strojno učenje, planiranje i provedba procesa industrijske automatizacije. Mnogi od prethodno navedenih pojmova biti će detaljnije opisani u nastavku ovog rada.

Trenutno, umjetna inteligencija je najviše orijentirana na čovjeka i usredotočena je na korištenje računala za podršku ljudskih aktivnosti putem sustava umjetne inteligencije [2].

2.2. Razvoj umjetne inteligencije

Iako je vrlo teško odrediti začetak primjene umjetne inteligencije, svakako treba istaknuti nekoliko povijesnih veličina koje su imale značajan utjecaj na njen razvoj. Prvi od njih je američki pisac Isaac Asimov koji 1942. godine izbacuje kratku priču „*Runaround*“ koja sadrži prvu eksplicitnu pojavu Tri zakona robotike. Asimovov rad inspirirao je generacije znanstvenika u području robotike, umjetne inteligencije i računalnih znanosti, a između

ostalih i slavnog američkog kognitivnog znanstvenika Marvinu Minsky-a koji je kasnije postao suosnivačem laboratorija umjetne inteligencije na prestižnom fakultetu MIT. Otprilike u isto vrijeme britanski matematičar Alan Turing razvio je stroj koji je danas poznat kao Turingov stroj, a koji je služio za dešifriranje njemačke Enigme za vrijeme drugog svjetskog rata. Radilo se o poduhvatu koji nije uspio poći za rukom ni najboljim ljudskim matematičarima pa se stoga Turingov stroj smatra prvim elektromehaničkim računalom. Turing je u svojim radovima opisao kako stvoriti inteligentne strojeve, a posebno kako testirati njihovu inteligenciju. Takav Turingov test se i danas smatra mjerom za identifikaciju inteligencije umjetnog sustava. Naime, ako čovjek komunicira s drugim čovjekom i sa strojem, i ne može razlikovati čovjeka od stroja, tada se kaže da je stroj inteligentan.

No, vjerojatno najvažnija godina u povijesti umjetne inteligencija bila je 1956. kada su, već spomenuti, Marvin Minsky i John McCarthy organizirali radionicu sa ciljem da okupe istraživače iz različitih područja kako bi stvorili novo istraživačko područje usmjereno na izgradnju strojeva sposobnih simulirati ljudsku inteligenciju [5]. McCarthy je na toj radionici iznio ključnu tvrdnju, a koja glasi: „Istraživanje će se temeljiti na pretpostavci da se svaki aspekt učenja ili bilo koje drugo svojstvo inteligencije u načelu može opisati toliko precizno da bi ih i stroj mogao simulirati“ [1]. Drugim riječima, svaki element inteligencije može se raščlaniti na manje korake tako da je svaki od tih koraka sam po sebi jednostavan i „mehanički“ da se može zapisati u obliku računalnog programa. Ta je tvrdnja bila, a i danas jest, samo pretpostavka, što znači da se ne može dokazati je li istinita. Bez obzira na to, ona je zaista ključna kad je riječ o načinu na koji se razmišlja o umjetnoj inteligenciji.

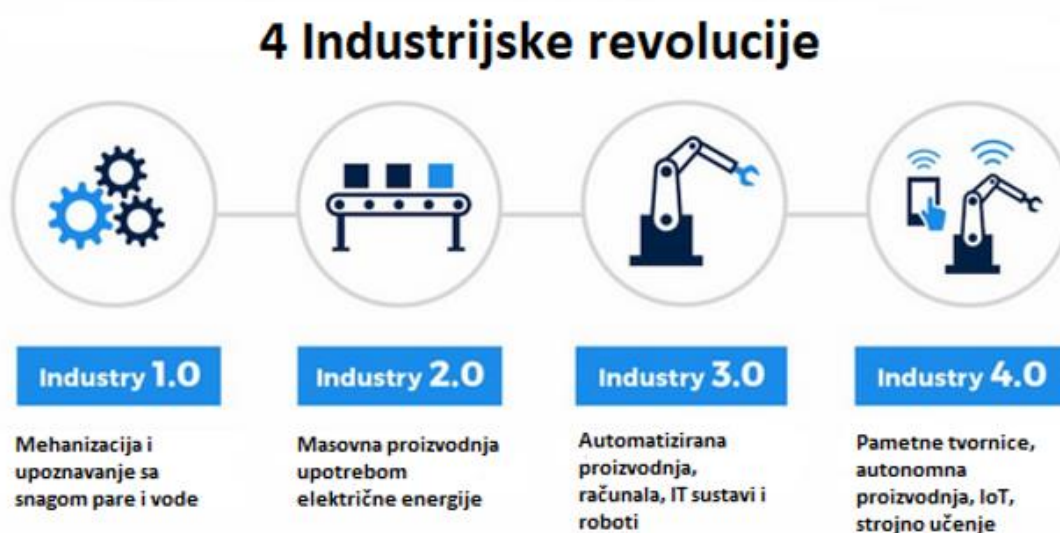
Idući bitan korak u razvoju umjetne inteligencije dogodio se 1997. godine kada je program za igranje šaha „*Deep Blue*“, razvijen od strane američke tvrtke IBM, uspio pobijediti tadašnjeg svjetskog prvaka u šahu Gary-a Kasparova. „*Deep Blue*“ je navodno mogao obraditi 200 milijuna mogućih kretanja u sekundi i odrediti sljedeći optimalni potez gledajući 20 poteza unaprijed pomoću metode koja se naziva traženje stabla.

Posljednji masivan događaj u razvoju umjetne inteligencije bio je 2015. godine kada je „*AlphaGo*“, program kojeg je razvila tvrtka Google, uspio pobijediti svjetskog prvaka u društvenoj igri „*Go*“. „*Go*“ je puno složeniji od šaha te se dugo vjerovalo kako računala nikada neće moći pobijediti ljude u ovoj igri. AlphaGo je postigao svoje visoke performanse korištenjem određene vrste umjetne neuronske mreže koja se naziva duboko učenje. Danas umjetne neuronske mreže i duboko učenje čine osnovu većine aplikacija koje poznajemo pod

oznakom umjetne inteligencije. Oni su osnova algoritama za prepoznavanje slika koje koristi Facebook, algoritama za prepoznavanje govora koji pokreću pametne zvučnike i autonomnih automobila [5].

2.3. Umjetna inteligencija u Industriji 4.0

Industrija 4.0 može se definirati kao digitalna transformacija proizvodnje i srodnih industrija te procesa stvaranja vrijednosti i kao takva predstavlja novu etapu u organizaciji i kontroli industrijskog lanca vrijednosti. Ona se odnosi na inteligentno umrežavanje strojeva i procesa za industriju uz pomoć informacijske i komunikacijske tehnologije [3]. Na slici 1 prikazane su sve 4 industrijske revolucije s navedenim glavnim značajkama za svaku od njih.



Slika 1 Industrija 4.0 [9]

Sa slike je vidljivo kako u domenu Industrije 4.0 spadaju pametne tvornice, autonomna proizvodnja, IoT (engl. *Internet of Things*) i strojno učenje. No, to su samo neka od mnogih obilježja Industrije 4.0, a od ostalih svakako vrijedi spomenuti umjetnu inteligenciju, kibernetičko-fizičke sustave te kognitivno računanje koji npr. mogu omogućiti startup-ove za pametne tvornice.

S industrijskog gledišta, tehnologije umjetne inteligencije treba shvatiti kao metode i postupke

koji omogućuju tehničkim sustavima da percipiraju svoje okruženje, da obrađuju ono što su opazili, samostalno rješavaju probleme, pronalaze nove vrste rješenja, donose nove odluke, a posebno da uče iz iskustva kako bi bolje rješavali zadatke [2]. Umjetna inteligencija i strojno učenje u Industriji 4.0 utiru put ogromnoj revoluciji za proizvodne tvrtke s različitim industrijskim mogućnostima. Proizvodni procesi u pametnoj tvornici su povezani sa sučeljima, strojevima i modulima koji međusobno komuniciraju, a gdje se može prikupiti velika količina informacija za poboljšanje proizvodnog procesa. Umjetna inteligencija u proizvodnji usmjerena je na industrijsku automatizaciju u potrazi za što većom efikasnošću. Ona može olakšati komunikaciju između ljudi i procesa omogućavanjem senzora, uređaja i kontrole za međusobno dijeljenje podataka.

Pomoću tehnika strojnog učenja može se automatizirati proces proizvodnje koji povećava produktivnost, učinkovitost, optimizira troškove proizvodnje i smanjuje ručne pogreške. Mnoga poduzeća koriste strojno učenje u službama održavanja i podrške koje analiziraju pojedinačne performanse stroja. Umjetna inteligencija u industriji 4.0 smanjuje pogreške predviđanja lanca opskrbe, pruža uvid u zakazane izvedbe proizvodnje, povećava bolju dostupnost proizvoda, učinkovito upravlja zalihama i bolje kontrolira kvalitetu. Također, predviđa sljedeći kvar sustava te upozorava radnu snagu na izvršenje održavanja kako bi se spriječio kvar [4]. Analitika zasnovana na umjetnoj inteligenciji poboljšava podatke kako bi se poboljšala učinkovitost te optimizacija proizvodnih operacija, kao i kvaliteta proizvoda te sama izvedba.

2.4. Srodna područja umjetne inteligencije

Najvažniji dijelovi umjetne inteligencije koji će biti zastupljeni u ovom radu su strojno učenje i duboko učenje. Na slici 2 je prikazan odnos umjetne inteligencije, strojnog učenja i dubokog učenja.



Slika 2 Dijelovi umjetne inteligencije [12]

Iz slike se jednostavno očitava kako je strojno učenje potpodručje umjetne inteligencije (koja je potpodručje računarstva), dok je duboko učenje potpodručje strojnog učenja.

2.4.1. Strojno učenje

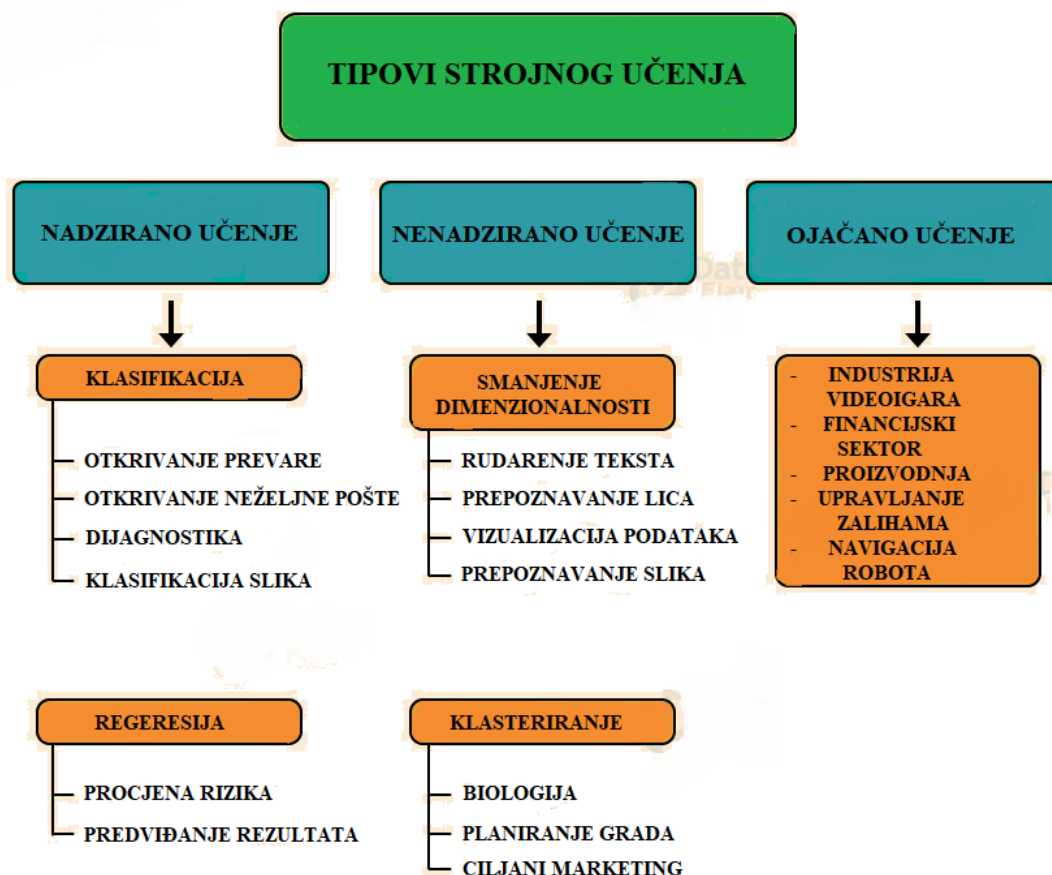
Strojno učenje je grana umjetne inteligencije koja podržava ideju da sustavi mogu učiti iz podataka, identificirati obrasce i donositi odluke uz minimalnu ljudsku intervenciju. Radi se o najpopularnijoj tehnici predviđanja budućnosti ili klasificiranja podataka koja pomaže ljudima u donošenju potrebnih odluka. Algoritmi strojnog učenja uče iz prošlih primjera podataka statističkom analizom i usklađivanjem uzoraka, a zatim na temelju naučenih podataka daju predviđene rezultate. Oni koriste obrasce sadržane u podacima o obuci za izvođenje klasifikacije i buduća predviđanja. Kad god se u model strojnog učenja uvede bilo koji novi ulaz, on primjenjuje svoje naučene obrasce na nove podatke kako bi napravio buduća predviđanja. Na temelju konačne točnosti, njihovi se modeli mogu optimizirati pomoću

različitih standardiziranih pristupa. Uz pomoć strojnog učenja mogu se razviti inteligentni sustavi sposobni za samostalno donošenje odluka [11].

Područje strojnog učenja često se dijeli u potpodručja na temelju vrsta problema koji se nastoje riješiti, a podjela je sljedeća [11]:

- **Nadzirano učenje** - cilj je na temelju dobivenog ulaza predvidjeti točan izlaz. Skup podataka na kojima se uči model je jasno označen te postoji jasno mapiranje ulaza i izlaza. Na temelju primjera ulaza, model se može obučiti u instancama. Dobar primjer nadziranog učenja je filtriranje neželjene pošte. Na temelju označenih podataka, model može utvrditi jesu li podaci neželjeni ili ne.
- **Nenadzirano učenje** - nema oznaka ni točnih izlaza, a zadatak je otkriti strukturu podataka. Algoritam identificira obrasce unutar skupa podataka i uči ih, a nakon toga grupira podatke u različite skupine na temelju njihove gustoće. Na taj način se može izvršiti vizualizacija podataka velikih dimenzija.
- **Ojačano učenje** - uglavnom se primjenjuje u situacijama u kojima određeni sustav umjetne inteligencije, kao što je autonomni automobil, mora raditi u nekom okruženju i u kojima su povratne informacije o prikladnim i neprikladnim odabirima dostupne sa zakašnjenjem.

Na slici 3 prikazana je podjela strojnog učenja.



Slika 3 Tipovi strojnog učenja [11]

U nadzirano učenje spadaju postupci klasifikacije i regresije. Primjeri klasifikacije su, već spomenuto, otkrivanje neželjene pošte te otkrivanje prijevare, dijagnostika i klasifikacija slika. S druge strane, pod regresiju su navedeni primjeri procjene rizika i predviđanja rezultata. Pod nenadziranim učenjem nalaze se tehnike smanjenje dimenzionalnosti i klasteriranje. Rudarenje teksta, prepoznavanje lica, vizualizacija velikih podataka i prepoznavanje slika su primjeri smanjenja dimenzionalnosti, dok su pod klasteriranjem navedeni biologija, planiranje grada i ciljani marketing. Posljednji tip je ojačano učenje koje se najviše primjenjuje u industriji videoigara, finansijskom sektoru, proizvodnji, upravljanju zalihama i navigaciji robota.

Prilikom primjene strojnog učenja, a da bi se izbjegle velike pogreške, skup podataka potrebno je podijeliti u dva dijela: podatke za učenje i podatke za ispitivanje [1]. Algoritam prvo uči isključivo iz podataka za učenje. Na taj način se dobiva model ili pravilo s pomoću

kojeg se na temelju ulaznih varijabli predviđaju izlazne varijable. Međutim, podacima za učenje ne smije se poslužiti kako bi se procijenilo koliko su dobivene vrijednosti predviđanja zapravo precizne. Premda određeni model može dobro predviđati na temelju podataka za učenje, to ne dokazuje da generalizacijom može proizvesti druge podatke. Za to služe podaci za ispitivanje: oni se unose u dobiveni model kako bi se dobile izlazne varijable s vrijednostima predviđanja, koje se zatim uspoređuju sa stvarnim izlaznim varijablama.

2.4.1.1. Klasifikator najbližih susjeda

U ovom dijelu biti će prikazan primjer strojnog učenja iz područja klasifikacije, a radi se o klasifikatoru najbližih susjeda. Klasifikator najbližih susjeda jedan je od najjednostavnijih klasifikatora. Pri klasifikaciji određenog elementa klasifikator pronalazi element iz podataka za učenje koji je najbliži novom elementu i generira njegovu oznaku. U slučaju umjetne inteligencije, najbliži susjedi važni su za predviđanje korisničkog ponašanja. Tipičan primjer primjene metode najbližih susjeda kod predviđanja korisničkog ponašanja su sustavi za preporuku sadržaja. Cilj je primijeniti vrlo jednostavno načelo u skladu s kojim su korisnici koji su se slično ponašali u prošlosti, skloni slično se ponašati i u budućnosti. Na sljedećem primjeru izrađen je jednostavan sustav za preporuku proizvoda za aplikaciju za internetsku kupovinu u kojoj se bilježi povijest kupnji korisnika te se na temelju nje predviđa koje će proizvode korisnik vjerojatno kupiti u budućnosti [1].

Prikupljeni su podaci za 6 korisnika koji su prikazani na slici 4. Za svakog korisnika navedena su 4 nedavno kupljena proizvoda i proizvod koji su kupili nakon kupnje tih četiriju proizvoda.

Korisnik	Povijest kupnji				Kupljeni proizvod
Sanni	boksačke rukavice	<i>Moby Dick</i> (roman)	slušalice	sunčane naočale	kava u zrnju
Jouni	majica kratkih rukava	kava u zrnju	aparati za kavu	kava u zrnju	kava u zrnju
Janina	sunčane naočale	tenisice	majica kratkih rukava	tenisice	vunene čarape
Henrik	2001.: <i>Odiseja u svemiru</i> (DVD)	slušalice	majica kratkih rukava	boksačke rukavice	japanke
Ville	majica kratkih rukava	japanke	sunčane naočale	<i>Moby Dick</i> (roman)	krema za sunčanje
Teemu	<i>Moby Dick</i> (roman)	kava u zrnju	2001.: <i>Odiseja u svemiru</i> (DVD)	slušalice	kava u zrnju

Slika 4 Podaci o korisnicima [1]

Zadatak je predvidjeti sljedeći proizvod koji će kupiti novi korisnik Travis koji je kupio proizvode prikazane na slici 5.

Korisnik	Povijest kupnji				Kupljeni proizvod
Travis	zeleni čaj	majica kratkih rukava	naočale	japanke	?

Slika 5 Podaci o Travisu [1]

Kako bi se primijenila metoda najbližih susjeda, treba se definirati što se podrazumijeva pod „najbliži“. Pojam sličnosti definira se na temelju povijesti kupnji tako da se izbroji koliko su proizvoda kupila oba korisnika. Npr. korisnici Ville i Henrik kupili su majicu kratkih rukava, stoga je njihova vrijednost sličnosti 1. Travis je kupio majicu kratkih rukava i japanke, kao i Ville, pa je njihova vrijednost sličnosti 2, što je najveća vrijednost od svih izračunatih. Kako je Ville za sljedeći kupljeni proizvod odabrao kremu za sunčanje, može se predvidjeti da će Travis također najvjerojatnije kupiti kremu za sunčanje.

U ovom primjeru na raspolaganju su bili podaci o samo 6 korisnika i provedeno predviđanje vjerojatno je bilo vrlo nepouzđano. Međutim, stranice internetskih trgovina često imaju milijune korisnika i generiraju goleme količine podataka. U mnogim slučajevima postoji mnoštvo korisnika čije je ponašanje u prošlosti bilo vrlo slično i na temelju čije se povijesti kupnje mogu vrlo dobro naslutiti nečiji interesi.

2.4.2. Duboko učenje

Duboko učenje dio je strojnog učenja koje uključuje korištenje umjetnih neuronskih mreža najčešće s 3 ili više slojeva. Algoritmi dubokog učenja najpopularniji su izbor u mnogim industrijama zbog sposobnosti neuronskih mreža da točnije uče iz velikih podataka i pružaju postojane rezultate korisniku. No, da bi se razumjelo što je duboko učenje, prvo se mora objasniti pojam umjetne neuronske mreže (engl. *Artificial Neural Networks*).

2.4.2.1. Umjetne neuronske mreže

Umjetne neuronske mreže su računalni modeli koji se temelje na strukturi i funkcijama bioloških neuronskih mreža. To je poput umjetnog ljudskog živčanog sustava za primanje, obradu i prijenos informacija u smislu računalnih znanosti [12]. One se sastoje od slojevitih sustava u kojima svaki sloj ima nekoliko blokova poznatih kao neuroni koji izvode određene računske zadatke. Ove neuronske mreže pokušavaju simulirati ponašanje ljudskog mozga, iako se ne podudaraju s njegovim sposobnostima, dopuštajući mu da uči iz velike količine podataka.

Razlog za odabir neuronskih mreža temelji se na pretpostavci da će inteligencija nastati bude li obrada podataka kakva se inače odvija na nižoj razini, odnosno ispod razine simbola, simulirana na razini neurona i neuronskih mreža. U usporedbi s tradicionalnim načinom rada računala neuronske se mreže odlikuju određenim posebnim značajkama.

U klasičnom se računalu informacije obrađuju u središnjem procesoru (engl. *Central Processing Unit*), koji se u određenom trenutku može usredotočiti samo na jednu zadaću. Središnja procesorska jedinica može dohvaćati podatke za obradu iz računalne memorije, a rezultat može pohraniti u memoriju. Stoga zadaću pohrane i zadaću obrade izvršavaju dvije različite komponente računala: memorija i središnja procesorska jedinica. Kod neuronskih mreža sustav sastoji od velikog broja neurona, a svaki od njih može samostalno obrađivati informacije. To znači da za razliku od središnje procesorske jedinice koja obrađuje jednu po jednu informaciju, neuroni mogu istodobno obraditi goleme količine informacija.

Druga je razlika ta da se pohrana (memorija) i obrada podataka ne odvijaju odvojeno kao u klasičnim računalima. Neuroni pohranjuju i obrađuju informacije, stoga podatke za obradu nije potrebno dohvaćati iz memorije. Podaci se mogu kratkoročno pohraniti u samim neuronima (koji se u određenom trenutku mogu, ali ne moraju aktivirati) ili se mogu pohraniti dugoročno, u vezama među neuronima, odnosno njihovim tzv. težinama.

Iako se neuronske mreže mogu u potpunosti simulirati u klasičnim računalima, kako su se dugo vremena i upotrebljavale, njihov maksimalni kapacitet postiže se samo ako se koriste posebni hardveri (računalni uređaji) koji mogu obrađivati više informacija istodobno. Taj se postupak zove paralelna obrada. Tu zadaću mogu obavljati grafički procesori (ili grafičke procesorske jedinice, engl. GPU = *Graphics Processing Units*) koji su postali isplativo rješenje za primjenu brojnih metoda dubokog učenja [1].

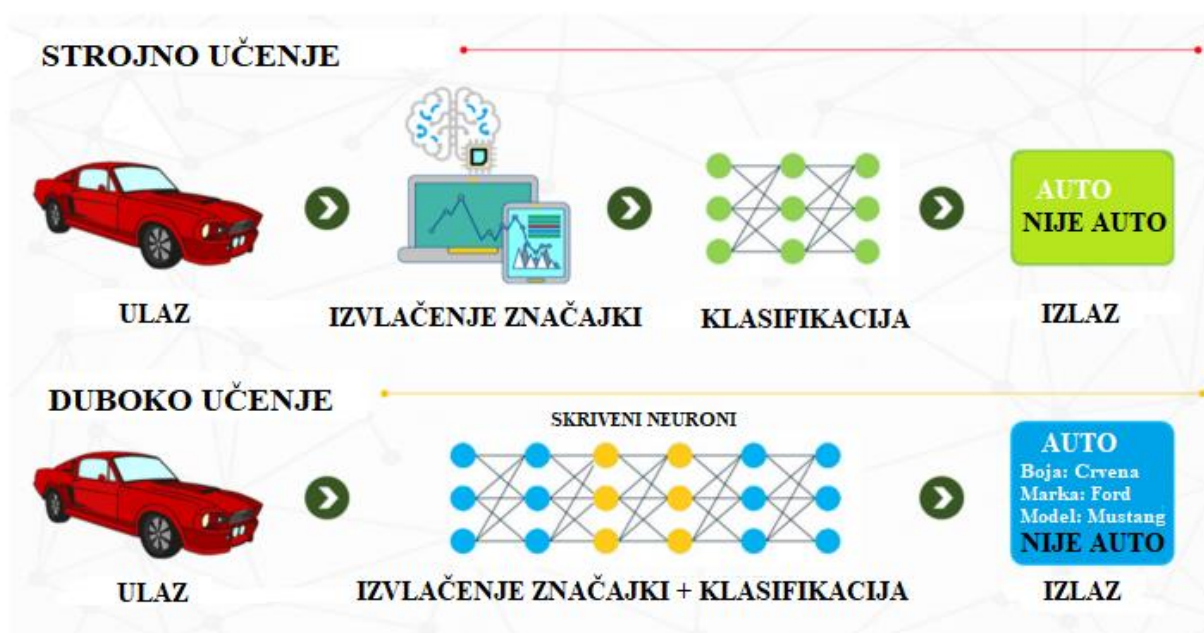
2.4.2.2. Konvolucijska neuronska mreža

Već je spomenuto kako strojno učenje teži tome da računala izvršavaju zadatke bez ljudske intervencije, ali ta računala i dalje razmišljaju i ponašaju se poput strojeva. Njihova sposobnost izvršavanja nekih složenijih zadataka poput prikupljanja podataka sa slika ili videozapisa još je uvijek daleko od onoga za što su ljudi sposobni. S druge strane, mreža s više od 2 sloja koja se naziva duboka neuronska mreža (engl. *Deep Neural Network*) može postići točnost sličnu čovjeku u zadacima kao što su klasifikacija slika, otkrivanje i klasifikacija predmeta, prepoznavanje govora, prepoznavanje rukopisa i računalni vid. Uz pravilan trening, neuronska mreža može izvršavati neke zadatke čak i točnije od čovjeka. Konvolucijska neuronska mreža (engl. *Convolutional Neural Network*) podvrsta je duboke neuronske mreže, odnosno ona predstavlja višeslojni algoritam koji nalikuje radu vidnog korteksa. Takve neuronske mreže se uglavnom koriste za obradu slike, video zapisa i jezika

[13]. Njihovo je ključno svojstvo sposobnost prepoznavanja obilježja slika kao što su svijetle ili tamne mrlje (ili mrlje određene boje), različito usmjereni rubovi, uzorci itd. Na tim se obilježjima temelji prepoznavanje apstraktnijih obilježja kao što su mačje uši, pseća njuška, ljudsko oko ili osmerokutni oblik znaka obvezno zaustavljanje. Neuronsku bi mrežu zapravo bilo teško naučiti da takva obilježja prepozna na temelju piksela ulazne slike. Naime, obilježja koja se pojavljuju na slikama mogu se razlikovati po veličini, usmjerenosti i položaju, pa će se zbog svakog pomicanja objekta ili kuta snimanja vrijednosti piksela drastično promijeniti čak i ako sam objekt izgleda potpuno jednako. Da bi mreža naučila prepoznati znak za obvezno zaustavljanje u svim tim različitim uvjetima, bile bi potrebne goleme količine podataka za učenje jer bi mreža taj znak mogla prepoznati samo u uvjetima u kojima se on pojavio u podacima za učenje. Stoga bi, na primjer, znak za obvezno zaustavljanje u gornjem desnom kutu slike bio prepoznat samo ako su podaci za učenje sadržavali sliku znaka za obvezno zaustavljanje u gornjem desnom kutu. Konvolucijske neuronske mreže mogu prepoznati objekt bilo gdje na slici bez obzira na to gdje je on bio smješten na slikama za učenje. Konvolucijski neuroni uglavnom se nalaze u donjim slojevima mreže, u kojima se obrađuju neobrađeni ulazni pikseli. Osnovni neuroni nalaze se u višim slojevima u kojima se obrađuje izlaz donjih slojeva. Donje slojeve najčešće se može učiti metodom nenadziranog učenja, ne vodeći računa o nekom konkretnom zadatku predviđanja. Njihove težine prilagoditi će se kako bi se mogla prepoznati obilježja koja se često pojavljuju u ulaznim podacima. Stoga, kad je riječ o fotografijama životinja, tipična obilježja bit će uši i njuške, dok su na slikama zgrada to arhitektonske komponente kao što su zidovi, krovovi, prozori itd. Ako se kao ulazni podaci upotrijebe mješavine različitih objekata i prizora, obilježja koja nauče donji slojevi bit će uglavnom generička. To znači da se tako dobiveni konvolucijski slojevi mogu ponovno upotrijebiti u brojnim različitim zadacima obrade slika. To je iznimno važno jer su neoznačeni podaci za učenje (odnosno slike bez oznaka) koji se mogu upotrebljavati za učenje donjih slojeva dostupni u gotovo neograničenim količinama. Gornje slojeve uvijek se uči s pomoću tehnika nadziranog strojnog učenja [1].

2.4.3. Razlika između strojnog učenja i dubokog učenja

Na slici 6 nalazi se usporedna grafika za strojno učenje i duboko učenje koja prikazuje tijek od ulaza do izlaza.



Slika 6 Usporedba strojnog učenja i dubokog učenja [12]

Na slici se nalazi primjer gdje je crveni automobil ulaz i za strojno učenje i za duboko učenje. Pomoću algoritama strojnog učenja dobije se izlaz da se radi o automobilu ili da se ne radi o automobilu (auto ili nije auto). U slučaju dubokog učenja izlaz je jednak (auto ili nije auto), ali zahvaljujući umjetnim neuronskim mrežama i obradi ogromne količine podataka, kao izlaz se ne dobije samo automobil nego i njegova boja, marka i model.

Iz prethodnog poglavlja o strojnom učenju i dubokom učenju treba zapamtiti sljedeće činjenice:

1. Duboko učenje je vrsta strojnog učenja koje je podskup umjetne inteligencije
2. Strojno učenje govori o tome da računala mogu razmišljati i djelovati uz manje ljudske intervencije, dok duboko učenje govori o tome kako računala uče razmišljati koristeći strukture po uzoru na ljudski mozak
3. Strojno učenje zahtijeva znatno manje računalne snage u odnosu na duboko učenje, dok duboko učenje zahtijeva manje ljudske intervencije
4. Duboko učenje može analizirati slike, videozapise i nestrukturirane podatke na način na koji to strojno učenje ne može učiniti
5. Svaka će industrija imati karijere koje uključuju strojno učenje i duboko učenje

2.4.4. Znanost o podacima i robotika

Znanost o podacima noviji je krovni pojam (obuhvaća više poddisciplina) koji se odnosi na strojno učenje i statistiku te određene aspekte računarstva, uključujući algoritme, pohranu podataka i izradu internetskih aplikacija. Znanost o podacima ujedno je i praktična disciplina koja zahtijeva razumijevanje područja u kojem se primjenjuje, na primjer područja poslovanja ili znanosti, odnosno njegove svrhe (značenje pojma „dodana vrijednost”), osnovnih pretpostavaka i ograničenja. Rješenja koja pruža znanost o podacima često uključuju barem neki oblik umjetne inteligencije [1].

Robotika se odnosi na izradu i programiranje robota kako bi mogli raditi u složenim situacijama iz stvarnog svijeta. Robotika je na neki način najveći izazov u području umjetne inteligencije jer zahtijeva kombinaciju praktički svih njezinih potpodručja. Na primjer [1]:

- računalni vid i prepoznavanje govora potrebni su za snalaženje u okruženju
- obrada prirodnog jezika, dohvaćanje informacija i donošenje odluka u neizvjesnim okolnostima služe za obradu uputa i predviđanje posljedica mogućih radnji
- kognitivno modeliranje i afektivno računarstvo (sustavi koji reagiraju na izražaje ljudskih osjećaja ili oponašaju osjećaje) upotrebljavaju se pri interakciji i radu s ljudima.

Mnogim problemima umjetne inteligencije koji su povezani s robotikom najbolje se može pristupiti s pomoću strojnog učenja, zbog čega je strojno učenje središnja grana umjetne inteligencije za robotiku.

3. PRIMJENA UMJETNE INTELIGENCIJE

3.1. Umjetna inteligencija u životu čovjeka

Umjetna inteligencija se pojavljuje u životu i poslu svakog pojedinca, bio on toga svjestan ili ne. Kao takva, ona postaje dio svakodnevnice te donosi jako puno prednosti u smislu olakšavanja ljudskog života, ali s druge strane također donosi i mnoge opasnosti. Definicije možda ne mogu do kraja približiti čovjeku što je to umjetna inteligencija, ali njena primjena zasigurno može.

3.1.1. Autonomni automobili

Prva primjena umjetne inteligencije su autonomni automobili (slika 7).



Slika 7 Autonomni automobil [1]

Iako još uvijek nisu uvelike rasprostranjeni po svjetskim ulicama, autonomni automobili pravi su primjer umjetne inteligencije u budućnosti.

U autonomnim je automobilima potrebno primjenjivati kombinaciju raznih tehnika umjetne inteligencije: pretraživanje i planiranje upotrebljavaju se za pronalaženje najprikladnije rute od točke A do točke B, računalni vid potreban je za utvrđivanje prepreka što je vidljivo iz slike, a sposobnost donošenja odluka u neizvjesnim okolnostima služi za snalaženje u

složenom i dinamičnom okruženju. Svaka od tih tehnika mora biti gotovo savršeno precizna kako bi se izbjegle nesreće. Iste se tehnologije upotrebljavaju i u drugim autonomnim sustavima, kao što su roboti dostavljači, bespilotne letjelice i autonomni brodovi [1]. Autonomni automobili sa sobom donose i posljedice, a one bi uglavnom trebale biti pozitivne. Najbitnija je definitivno sigurnost na cestama koja će biti povećana jer razina pouzdanosti sustava nadilazi ljudske sposobnosti. Također, povećati će se i učinkovitost logističkih lanaca pri prijevozu robe, okoliš će postati čišći, a za posljedicu se očekuju i ekonomski rast te stvaranje novih poslova.

3.1.2. Preporuka sadržaja

Vjerojatno najčešća primjena umjetne inteligencije u svakodnevnom životu je preporuka sadržaja. Svaka osoba koja se koristi Internetom nailazi na razne informacije, a mnoge od njih nisu personalizirane. To su, na primjer, sadržaji na društvenim mrežama kao što su Facebook, Instagram ili Twitter, internetski oglasi te preporuke filmova na uslugama kao što su Netflix ili HBO. Ponudeni sadržaj personaliziraju i mnogi internetski izdavači kao što su novinski portali i stranice radiotelevizijskih kuća, te internetski pretraživači kao što je Google. Tako dok je naslovnica tiskanih novina ista za sve čitatelje, naslovnica portala za svakog je pojedinog korisnika drugačija. Algoritmi koji određuju koji će sadržaj osoba vidjeti temelji se upravo na umjetnoj inteligenciji [1].

Preporuka sadržaja donosi mnoge negativne posljedice kao što su filter-mjehurići (engl. *filter bubbles*), komore jeke (engl. *echo chambers*), tvornice trolova (engl. *troll factories*), lažne vijesti (engl. *fake news*) i novi oblici propagande.

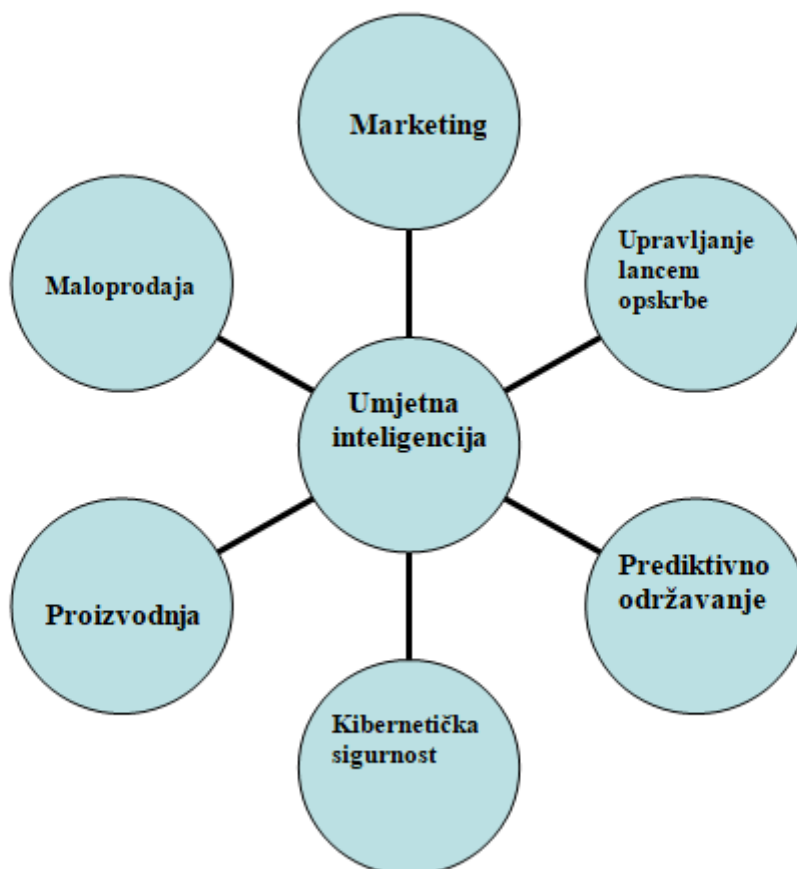
3.1.3. Obrada slika i videozapisa

Posljednji primjer umjetne inteligencije u ovom odlomku je obrada slika i videozapisa. Sustav za prepoznavanje lica već ima brojne korisničke, poslovne i administrativne primjene, na primjer za grupiranje fotografija prema osobama koje su na njima prikazane, automatsko označavanje na društvenim mrežama i provjeru putovnica. Slične tehnike mogu se upotrebljavati i za prepoznavanje drugih vozila i prepreka oko autonomnog automobila [1]. I u ovom primjeru može doći do negativnih posljedica jer će tehnike s vremenom postati sve

naprednije te će moći lako izraditi lažne videozapise događaja koji izgledaju prirodno i koje će biti nemoguće razlikovati od stvarnih snimaka.

3.2. Primjena umjetne inteligencije u industriji

Umjetna inteligencija može se primijeniti u gotovo svim sferama jednog poduzeća, što je prikazano na slici 8.



Slika 8 Umjetna inteligencija u industriji [6]

Na slici je navedena primjena umjetne inteligencije u marketingu, maloprodaji, lancu opskrbe, kibernetičkom održavanju, prediktivnom održavanju te proizvodnji. U nastavku slijedi kratak opis svih primjena [6].

3.2.1. Marketing

U marketingu se korištenjem podataka u stvarnom vremenu, nestrukturiranih i web podataka poboljšava postrojenje u smislu dostave informacija i preporuka na pravo mjesto u pravo vrijeme. Umjetna inteligencija može analizirati podatke sa društvenih mreža kako bi se povećao personalizirani marketing koji dovodi do više ciljanih interakcija i marketinga.

3.2.2. Maloprodaja

Umjetna inteligencija u maloprodaji transformira način na koji ljudi kupuju i pomaže organizaciji da optimizira zalihe kroz poboljšanje logistike i isporuke. Ona se proteže na proširenu stvarnost, robote i virtualne pomoćnike koji sugeriraju proizvode i usluge kupcima što pomaže organizaciji da se usredotoči na kupce visoke vrijednosti. U maloprodaji industrijski „botovi” i digitalni asistenti koji se bave automatiziranim razgovorima transformiraju način na koji se odvijaju funkcije održavanja.

3.2.3. Lanac opskrbe

Lanac opskrbe integrira krajnji proces proizvoda i isporučuje ih potrošačima. Slučajevi upotrebe umjetne inteligencije u logistici i optimiziranju opskrbnog lanca se sve više povećava. Strojno učenje otkriva obrasce u lancu opskrbe koji revolucioniraju svako poslovanje. Algoritmi strojnog učenja analiziraju velike skupove podataka za donošenje pametnijih odluka o planiranju, niži inventar i operativne troškove, poboljšanu izvedbu isporuke i smanjeni rizik dobavljača.

3.2.4. Kibernetička sigurnost

Što se tiče primjene u svrhu kibernetičke sigurnosti, umjetna inteligencija štiti poduzeće otkrivanjem neobičnih aktivnosti i obrazaca koji sadrže podatke kretanja paketa. Analizira ogromnu količinu podataka i odgovara na različite situacije.

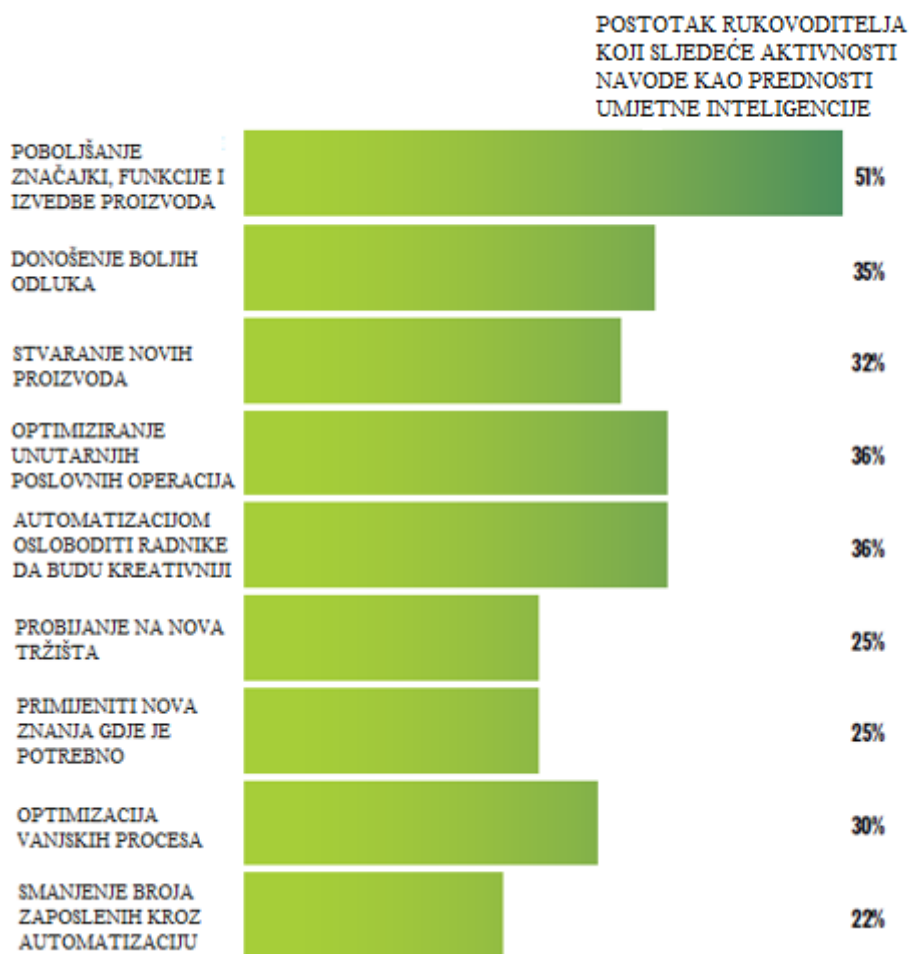
3.2.5. Prediktivno održavanje

Prediktivno održavanje učinkovito analizira rad i postupak lanca opskrbe, korištenje imovine i gubitak prihoda zbog zastoja. Prediktivna analitika bilježi podatke o opremi u stvarnom vremenu i procjenjuje povijesne podatke za procjenu životnog ciklusa opreme. Koristi

algoritam strojnog učenja za otkrivanje osnovnog uzroka kvarova. To je napredna analitika za optimiziranje planiranja rada, nadzor podataka u stvarnom vremenu, prepoznavanja uzoraka i modeliranje podataka. Prediktivno održavanje smanjuje neplanirano isključivanje predviđanjem kada je potrebno održavanje i optimizacija rada kako bi se postigli uvjeti za maksimizaciju vijeka trajanja opreme.

3.2.6. Prednosti i nedostaci implementacije umjetne inteligencije u poduzeće

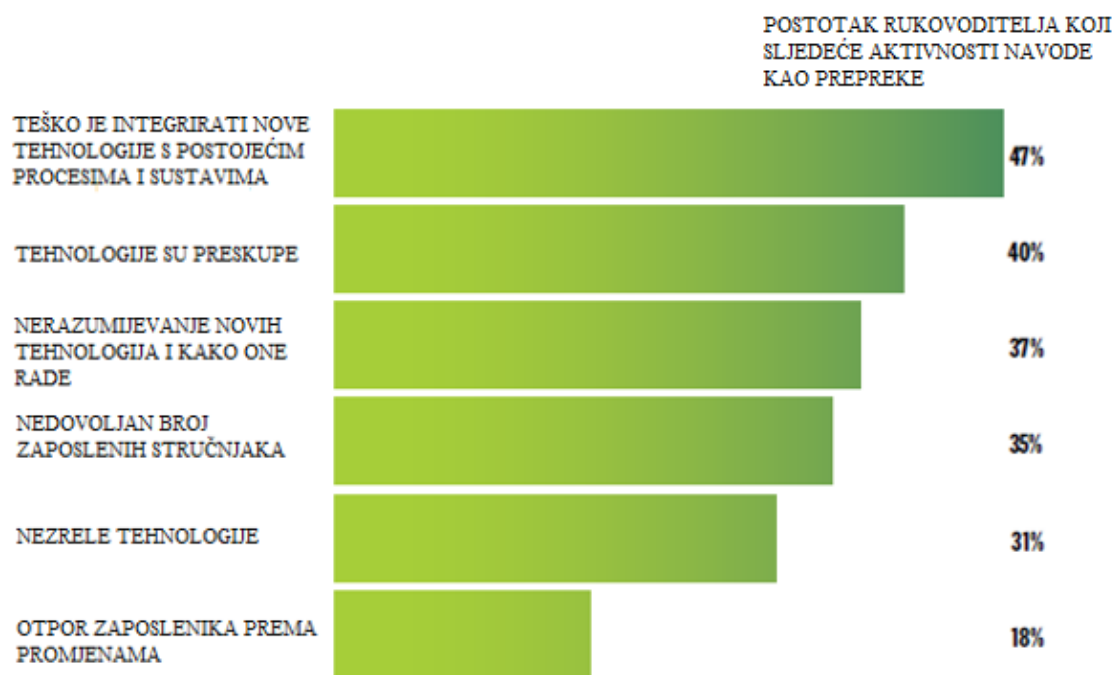
Pitanje koje si postavljaju mnogi svjetski rukovoditelji jest zašto bi oni uveli neku od tehnologija umjetne inteligencije u svoje poduzeće. Unatoč mnogim prednostima koje umjetna inteligencija donosi u industriji, ona još uvijek nije pretjerano zastupljena u poduzećima. Zašto je tome tako biti će objašnjeno pomoću rezultata ankete koja je provedena na 250 rukovoditelja u čijim poduzećima je implementiran neki oblik umjetne inteligencije [7]. Na slici 9 je prikazano koji je bio prvotni cilj poduzeća prilikom implementacije umjetne inteligencije.



Slika 9 Prednosti umjetne inteligencije [7]

Kao što je vidljivo sa slike, glavni cilj većine rukovoditelja bio je postizanje kvalitetnijeg proizvoda kroz poboljšanje značajki, funkcije i izvedbe. Zanimljivo, smanjenje broja zaposlenih kroz automatizaciju spomenulo je tek 22% ispitanika što sugerira na činjenicu kako je čovjek i dalje nezamjenjiv član poduzeća. Od ostalih razloga za uvođenje umjetne inteligencije najzastupljeniji su donošenje boljih odluka, stvaranje novih proizvoda, optimiziranje unutarnjih poslovnih operacija, rasterećivanje radnika uvođenjem automatizacije, probijanje na nova tržišta, primjena novih znanja gdje je potrebno te optimizacija vanjskih procesa.

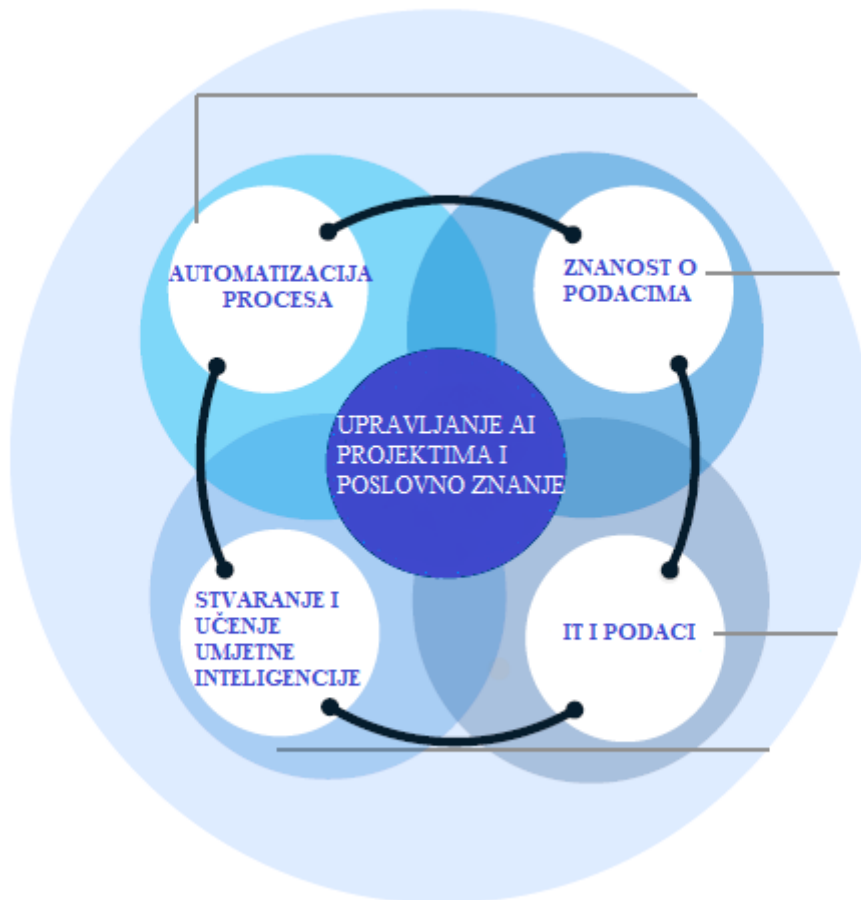
S druge strane, na slici 10 prikazani su glavni problemi s kojim su se poduzeća susretala prilikom implementacije umjetne inteligencije, koji se onda mogu shvatiti kao razlog njene manje zastupljenosti.



Slika 10 Nedostaci umjetne inteligencije [7]

Glavni problem koji se ističe je integracija nove tehnologije s već postojećim procesima i sustavima. Od ostalih prepreka izdvajaju se cijene tehnologija koje su izrazito visoke, njihovo nerazumijevanje i kako one rade, nedovoljan broj zaposlenih stručnjaka koji bi mogli upravljati tehnologijama umjetne inteligencije te nezrele tehnologije. Još jedan zanimljivi rezultat je taj da otpor zaposlenika prema promjenama iznosi tek 18% na temelju čega se može zaključiti da ljudi još uvijek nemaju strah od sve većeg razvoja umjetne inteligencije i eventualnog gubitka radnog mjesta u budućnosti.

Vidljivo je kako veliki otpor implementiranju umjetne inteligencije pružaju neznanje i nerazumijevanje. Na slici 11 prikazane su vještine koje bi svako poduzeće trebalo imati ako želi, u ovom slučaju, implementirati tkz. „in-house“ umjetnu inteligenciju.



Slika 11 Vještine poduzeća za implementaciju umjetne inteligencije [8]

U središtu slike se, kao najbitnija stavka, nalazi upravljanje projektima i znanjima. Također, potrebno je imati čitav niz onoga što se na engleskom jeziku naziva „*know-how*“. Radi se o potrebnom znanju i iskustvu o procesnim podacima i automatizaciji, naprednoj analitici, IT infrastrukturi te učenju modela umjetne inteligencije.

3.3. Umjetna inteligencija u proizvodnji

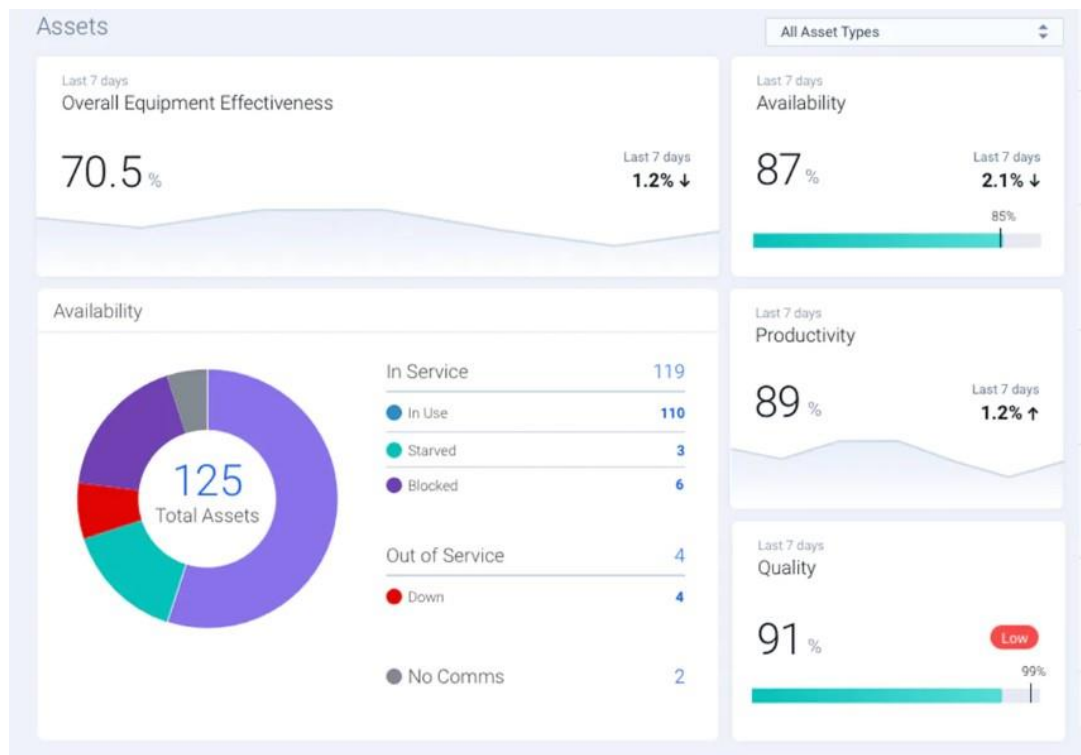
Umjetna inteligencija se u proizvodnji može koristiti u različite svrhe, a u ovom radu objašnjena je njena primjena u nekoliko područja proizvodnog poduzeća.

3.3.1. Primjena umjetne inteligencije

3.3.1.1. Pametno održavanje

Budući da je vrlo važan dio svakog proizvodnog postupka, održavanje opreme jedan je od najvećih troškova u proizvodnoj industriji. Neplanirani zastoji koji se pojavljuju u tvornicama sa sobom donose gubitke u milijardama dolara od čega je otprilike 40% [9] zbog gubitka imovine. Zbog tog razloga je predviđanje održavanja postalo vrlo bitno rješenje koje može pomoći uštedjeti poduzećima ogromnu količinu novca. Složeni algoritmi umjetne inteligencije poput neuronskih mreža i strojnog učenja stvaraju pouzdana predviđanja u pogledu statusa imovine i strojeva. Preostali vijek trajanja opreme (engl. *Remaining Useful Life*) postaje znatno duži. Sve to omogućuje tehničarima da mogu unaprijed znati ako nešto treba popraviti ili zamijeniti, pa čak i koje metode koristiti za rješavanje problema.

Na slici 12. prikazan je jedan primjer kako može izgledati ekran s povratnim informacijama koje se dobiju primjenom umjetne inteligencije u održavanju [4].



Slika 12 Pametno održavanje [4]

Na slici je prikazana ukupna iskoristivost imovine u prethodnih 7 dana koja iznosi 70,5% i predstavlja pad od 1,2% u odnosu na tjedan prije. Dostupnost je 87%, produktivnost 89%, dok kvaliteta iznosi 91%. Dostupnost je dodatno pojašnjena gdje se lako može vidjeti koliko je imovine u službi, a koliko nije. Isto tako, koliko imovine je trenutno u upotrebi, a koliko je u kvaru itd.

3.3.1.2. Poboljšanje kvalitete

U suvremenom svijetu kratkih rokova i povećane razine složenosti proizvoda postaje još teže ispuniti najviše standarde i propise u pogledu kvalitete. Kupci, naravno, uvijek očekuju besprijekorne proizvode. Također, nedostaci proizvoda mogu uzrokovati opoziv, što uvelike šteti ugledu tvrtke i njezinom brendu. Umjetna inteligencija može upozoriti tvrtke na probleme u proizvodnoj liniji koji mogu rezultirati nedovoljno dobrom kvalitetom. Ove greške mogu biti velike ili male, ali sve utječu na ukupnu razinu proizvodnje i mogle bi se otkloniti u ranim fazama. Računalni vid rješenje je umjetne inteligencije koje koristi kamere visoke rezolucije za praćenje defekata mnogo bolje nego što to može čovjek. Može se kombinirati s okvirom za obradu podataka zasnovanim na oblaku (engl. *cloud*) koji generira automatski odgovor. Također, proizvođači mogu dobiti podatke o performansama svojih proizvoda kada se pojave na tržištu kako bi u budućnosti donosili bolje strateške odluke.

3.3.1.3. Prilagođavanje tržištu

Umjetna inteligencija može poboljšati opskrbne lance poduzeća čineći ih interaktivnim s promjenama na tržištu unaprijed. Tako menadžeri mogu poboljšati svoju stratešku viziju oslanjajući se na prijedloge umjetne inteligencije. Ona generira procjene na temelju povezivanja brojnih čimbenika, poput političkih situacija, vremena, ponašanja potrošača i statusa gospodarstva. Osoblje, inventar i zalihe materijala također bi se mogli izračunati prema predviđanjima [9].

Postoji još nekoliko zanimljivih primjera upotrebe umjetne inteligencije, a jedna od njih su Digitalni blizanci. Umjetna inteligencija ima mogućnost potpune vizualizacije infrastrukture, proizvoda ili usluge. Proces virtualnog predstavljanja neke proizvodne jedinice naziva se digitalni blizanac. Korištenjem alata za prikupljanje podataka poput senzora i kamera, fizički prikaz proizvodnog okruženja može biti u potpunosti vizualiziran. Da bi korisnik bio siguran da digitalni blizanci rade ispravno, treba integrirati sve pametne komponente poput senzora

koji prikupljaju podatke s opreme. Korištenjem veze u oblaku, podaci generirani pametnim komponentama se prikupljaju, a zatim pohranjuju i obrađuju [4].

3.3.2. Primjeri proizvodnih poduzeća koja su implementirala umjetnu inteligenciju

Najveće tvrtke širom svijeta već koriste umjetnu inteligenciju i strojno učenje u proizvodnji i ulažu milijune u njezin razvoj, a ističu se svjetski velikani Siemens, General Electric i Fanuc.

3.3.2.1. Siemens

Siemens je 2016. godine predstavio svoj pametni oblak koji proizvođačima omogućuje nadzor nad strojnim parkovima diljem svijeta. Svrha ovog rješenja je prikupiti sve parametre u proizvodnom procesu od razvoja do isporuke i pronaći probleme i načine za njihovo rješavanje. Također, Siemens koristi umjetnu inteligenciju u neuronskim mrežama u svojim plinskim turbinama. Preko 500 senzora prati različite parametre, a sustav uči i donosi odluke o podešavanju vrijednosti goriva za najučinkovitije performanse.

3.3.2.2. General Electric

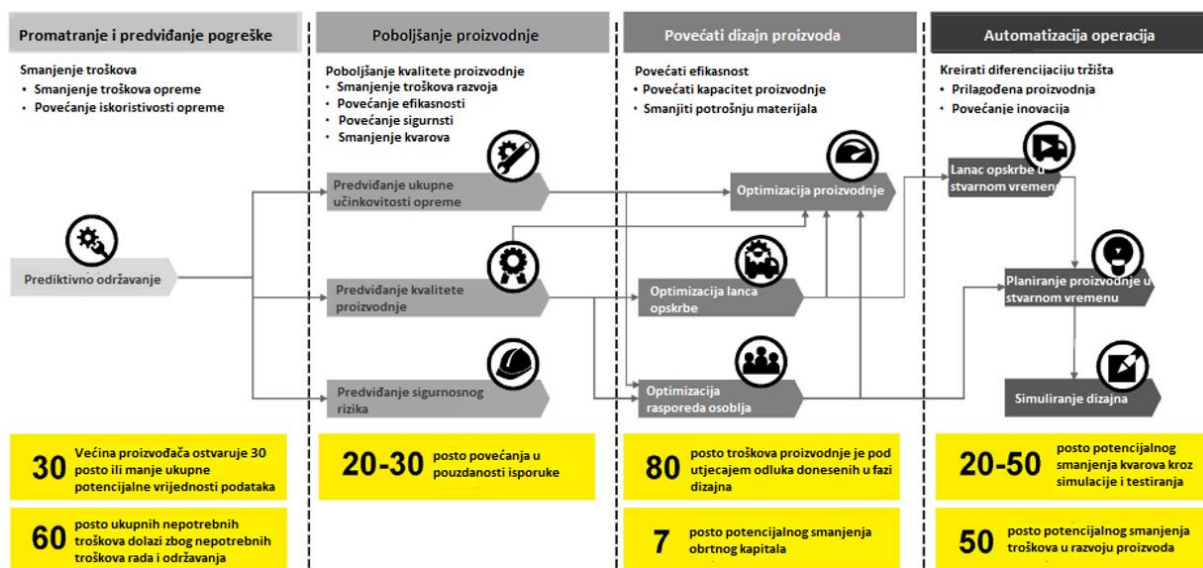
General Electric razvilo je rješenje koje nastoji pratiti i obrađivati sve informacije u svakom aspektu proizvodnje kako bi pronašao sve moguće probleme i kvarove. Primjerice, njihova prva tvornica koja se nalazi u Indiji dobila je 200 milijuna dolara ulaganja i povećala je stopu učinkovitosti postrojenja za 18%. Cilj je povezati sve elemente proizvodnje, poput dizajna, inženjeringa ili distribucije, u jedan globalni pametni sustav koji je skalabilan. To rješenje čak ima i svoju industrijsku IoT platformu koja koristi senzore za praćenje svih aspekata proizvodnog procesa i performansi sofisticirane opreme. Ona ima mogućnosti dubokog učenja koje mogu obraditi sve te informacije i doći do korisnih uvida.

3.3.2.3. Fanuc

Fanuc implementira umjetnu inteligenciju u svoje kompanije kako bi robote učinili pametnijima. Oni su, zapravo, lider u industrijskoj robotici integrirajući duboko učenje u robote. Fanuc, kao i General Electric, ima razvijenu vlastitu IoT platformu za proizvodnu industriju. Korištenje pojačanog dubokog učenja dovelo je do sposobnosti nekih industrijskih robota da se sami obučavaju, a cilj je omogućiti da više robota uči istovremeno jer ako roboti mogu učiti zajedno, to će biti znatno brže za svakog od njih ponaosob. U budućnosti će roboti

zasigurno moći međusobno dijeliti svoje vještine i tako štedjeti vrijeme u proizvodnim procesima u Pametnoj tvornici [9].

Na slici 13 prikazan je lanac primjene umjetne inteligencije u poduzeću Fanuc.



Slika 13 Umjetna inteligencija u Fanucu [9]

U prvoj fazi promatra se proizvodni proces i predviđaju se pogreške. Smanjuju se troškovi opreme, a povećava se njena iskoristivost. Za to je zaslužno prediktivno održavanje. Smatra se kako većina proizvođača ostvaruje 30% ili manje ukupne potencijalne vrijednosti podataka. Također, od ukupnih nepotrebnih troškova, njih 60% dolazi zbog nepotrebnih troškova rada i održavanja.

U drugoj fazi nastupa poboljšanje kvalitete proizvodnje. Ono se postiže smanjenjem troškova razvoja, povećanjem efikasnosti, povećanjem sigurnosti i smanjenjem kvarova. Takvi rezultati mogući su zahvaljujući metodama kao što su predviđanje ukupne učinkovitosti opreme, predviđanje kvalitete proizvodnje i predviđanje sigurnosnog rizika. Fanuc u toj fazi ostvaruje povećanje u pouzdanosti opreme (20-30)%.

Treća faza je faza povećanja dizajna proizvoda. Efikasnost raste povećanjem kapaciteta proizvodnje i smanjenjem potrošnje materijala. Optimizacija proizvodnje, lanca opskrbe i rasporeda osoblja ključna je za povećanje efikasnosti. Ova faza je bitna jer je 80% troškova proizvodnje pod utjecajem odluka donesenih u fazi dizajna.

Automatizacija operacija predstavlja posljednju fazu. Proizvodnja se prilagođava, a inovacije se povećavaju. Automatizacija je moguća zahvaljujući lancu opskrbe u stvarnom vremenu, planiranju proizvodnje u stvarnom vremenu i simuliranju dizajna. Kroz simulaciju i testiranja može doći do potencijalnog smanjenja kvarova (20-50)%. Smanjenje troškova u razvoju proizvoda može iznositi i do 50%.

3.3.3. Budućnost umjetne inteligencije u proizvodnom sektoru

Na slici 14 prikazano je izvješće o tome kako se očekuje da će tehnologije umjetne inteligencije u proizvodnom sektoru rasti u idućih pet godina.



Slika 14 Primjena umjetne inteligencije u budućnosti [27]

Danas su u proizvodnoj industriji najzastupljenije tehnologije koje pružaju optimizaciju resursa s omogućenim podacima, a tako bi trebalo i ostati u narednih 5 godina kada se očekuje da će biti korištena od strane čak 77% poduzeća. Najveći napredak predviđa se u korištenju tehnologija prediktivnog održavanja, optimizacije kvalitete i procesa vođenih velikim podacima, automatizacije procesa, povezane tvornice te integriranog planiranja. Za

sve navedene primjene se očekuje da će u narednih 5 godina biti zastupljene u više od 60% poduzeća. Danas se umjetna inteligencija u proizvodnom sektoru najmanje koristi u vidu postizanja potpuno autonomnosti digitalne kompanije (svega 5%), a to se u budućih 5 godina vjerojatno neće puno promijeniti pošto se očekuje rast od svega 6%. Iz toga se može zaključiti kako će do potpunog postizanja autonomnosti poduzeća u cjelini ipak morati proći nešto više vremena.

4. PREPOZNAVANJE LJUDSKIH AKTIVNOSTI

Prepoznavanje ljudskih aktivnosti (engl. *Human Activity Recognition*) je grana znanosti koja spada u domenu računalnog vida (engl. *Computer Vision*), a definira se kao proces zaključivanja o aktivnosti 3D objekata sa video sekvenci koje karakterizira visoka stopa preklapanja objekata sa samim sobom ili drugim objektima i velik broj spojnih točaka [14]. U posljednjih nekoliko godina postaje sve zastupljenije i popularnije, zahvaljujući znatnom razvoju hardvera i softvera koji omogućuju njenu širu i dublju primjenu. Glavni cilj prepoznavanja ljudskih aktivnosti je izgradnja inteligentnog stroja koji može točno protumačiti ljudsko ponašanje i postupke iz videa [15].

4.1. Primjena i aktivnosti prepoznavanja ljudskih aktivnosti

4.1.1. Područje primjene

Postoje 3 glavna područja primjene prepoznavanja ljudskih aktivnosti [16]:

1. Nadzor (engl. *surveillance*) – videom se prati jedna ili više osoba kako bi se ustanovilo izvodi li itko od njih sumnjive aktivnosti. Primjeri nadzora su nadgledanje javnih površina i nadzor aerodroma.
2. Kontrola (engl. *control*) – videom se snimaju pokreti jedne osobe koji se interpretiraju kao naredbe ili komande odvojenom, vanjskom sustavu. Primjer kontrole su sustavi virtualne stvarnosti.
3. Analiza (engl. *analysis*) – jedna ili više osobe se prate kroz duži vremenski period te se prepoznavanje pokreta vrši ili u realnom vremenu ili naknadno, nakon pohranjivanja snimljenog materijala. Detaljno se analiziraju snimljeni pokreti te je najznačajniji doprinos ovakve primjene rezultat analize pokreta. Primjer analize su aplikacije koje analiziraju neki sportski događaj te daju statistiku (npr. tehnički dobro ili loše izvedenih udaraca) i zdravlje gdje se može nadgledati oporavak pacijenta ili sprječavati nastupanje bolesti kroz preventivni nadzor i analizu.

4.1.2. Osnovne vrste aktivnosti

Osnovne ljudske aktivnosti koje ulaz u proces prepoznavanja su [19]:

1. Gesta - gesta ili nedjeljiva akcija je pokret koji se ne može dalje raslojavati, odnosno pokret na razini ekstremiteta. Primjer geste je podizanje nogu ili ispruživanje ruke.
2. Akcija – akcija je niz gesta jedne osobe koje zajedno (gledane kao cjelina) definiraju pokret cijelog tijela ili samo jednog njegovog dijela. Primjer akcije je mahanje rukom, šetnja ili udarac šakom.
3. Aktivnost – aktivnost predstavlja niz akcija koje se vremenski mogu odvijati jedna iza druge ili istovremeno , a koje analizirane zajedno (kao skup) otkrivaju o kojoj vrsti aktivnosti na videu se radi. Aktivnost fizičkog napada se sastoji od akcije hodanja (do mjesta napada) plus akcije udaranja šakom.
4. Interakcija – prethodno su navedeni pokreti na razini jedne osobe , od osnovnih do najkompleksnijih. Međutim, postoje i definicije radnji ili pokreta koji definiraju radnje na nivou više osoba ili osobe i okoliša, a nazivaju se interakcije. Dakle, radi se o aktivnostima koje podrazumijevaju aktivnosti najmanje dvije osobe ili aktivnost jedne osobe i objekta iz okoliša.
5. Grupna aktivnost – grupna aktivnost predstavlja skup interakcija , dakle više osoba ili osoba i predmeta koji istovremeno obavljaju interakciju. Primjer grupne aktivnosti više osoba je poslovni sastanak, a osoba i predmeta utrka automobila.

4.2. Obrada slika i videozapisa

Prepoznavanje ljudskih aktivnosti iz videozapisa vrlo je izazovan zadatak. Analiza ljudskog djelovanja nije samo pitanje predstavljanja obrazaca kretanja različitih dijelova tijela, već je to i opis namjere, emocija i misli neke osobe. Stoga je postao ključna komponenta u analizi i razumijevanju ljudskog ponašanja, koja je bitna u različitim područjima kao što su sigurnosni nadzor, robotika, zdravstvena zaštita itd.

Sustav za prepoznavanje kretnji obično analizira određene video sekvence ili okvire kako bi naučio obrasce određene ljudske kretnje u procesu obuke i upotrijebio naučeno znanje za klasifikaciju sličnih kretnji tijekom faze testiranja [17].

Primarni izazov video analize je razumijevanje prostorno-vremenskih elemenata unutar video isječka. Dok se slika sastoji isključivo od prostornih elemenata (različitih piksela i boja koji zauzimaju prostor unutar slike), video analiza predstavlja izazov za razumijevanje i analizu vremenskih aspekata videozapisa. Vremenski elementi u videozapisu su odnos vremena s

obzirom na prostorne aspekte slike. Na primjer slika 15 prikazuje čovjeka i kvaku na vratima te je iz nje moguće izvući uvide poput „Vrata“, „Čovjek“ i „Čovjek u interakciji s vratima“ putem algoritama za analizu slike. Međutim, dodavanjem i analizom naknadnih okvira i uključivanjem kretanja vrata mogu se iskoristiti vremenski elementi da se naznači je li radnja koja se izvodi „zatvaranje vrata“ ili „otvaranje vrata“ [20].



Slika 15 Otvaranje/zatvaranje vrata [20]

4.2.1. Pristup temeljen na dubokom učenju

Tijekom posljednjih nekoliko godina, pristupi temeljeni na dubokom učenju postali su vrlo popularni u području istraživanja prepoznavanja ljudskih aktivnosti iz videozapisa jer imaju mogućnost hijerarhijski učiti značajke iz više slojeva i automatski stvarati prikaz sirovih unosa na visokoj razini. Stoga je, za razliku od tradicionalnih pristupa, proces izdvajanja značajki potpuno automatiziran. Sustav dubokog učenja koristi metode kao što su lokalna percepcija, dijeljenje težine, jezgra s više konvolucija, spuštanje u niz itd. za učenje lokalnih značajki iz dijela slike umjesto iz cijele slike [17]. Konačni izlaz prepoznavanja određen je rezultatom više konvolucijskih slojeva.

Jedan od popularnih pristupa dubokog učenja koji se koristi za slike/okvire je konvolucijska neuronska mreža (CNN). Njena arhitektura primijenjena je za generiranje više kanala informacija i izvođenje konvolucije i poduzorkovanja u svakom kanalu iz susjednih video okvira. CNN može prepoznati emocije, geste, govor i rukopis, otkriti i klasificirati predmete te otkriti i prepoznati kretnje u videozapisu. Međutim, postoje mnogi izazovi u korištenju CNN-a za prepoznavanje i otkrivanje kretnji u videozapisu od kojih se izdvajaju sljedeće: pronalaženje ravnoteže između točnosti i performansi, priprema skupa podataka sa sličnim

parametrima onom s kojim je mreža trebala raditi, odabir relevantnog CNN-a za zadatak i njegovo konfiguriranje.

Glavna prednost pristupa dubokog učenja u usporedbi s tradicionalnim pristupima je njihova sposobnost prepoznavanja aktivnosti na visokoj razini sa složenim strukturama. Stoga istraživači radije koriste metode dubokog učenja za uključivanje predstavljanja značajki, kao što su interesne točke u vremenskom prostoru, frekvencija, lokalni deskriptori i modeliranje tijela iz videa, dubinskih video zapisa ili RGB (engl. *Red Green Blue*) video podataka. Obećavajuće performanse, robusnost u izdvajanju značajki i mogućnosti generalizacije pristupa dubokog učenja glavni su razlog njihove sve veće upotrebe.

Međutim, ti pristupi također imaju nekoliko nedostataka kao što su potreba za generiranjem velikih skupova podataka i ovisnost performansi o razmjeru težina mreže, dok je podešavanje hiperparametara beznačajno [17].

4.2.2. Izazovi u obradi videozapisa

U procesu klasifikacije videozapisa postoje 3 osnovna izazova [21]: anotirani skupovi podataka, računaska snaga i učinkoviti modeli strojnog učenja.

Imati čist, točan i označen skup podataka temeljni je aspekt bilo kojeg nadziranog učenja. U posljednjih nekoliko godina postoji nekoliko skupova podataka koji se izdvajaju kao što su UFC-101, HMDB-51, Sports1M i Kinetics. Ti su skupovi videozapisa standardna mjerila za obuku video klasifikacije i algoritama prepoznavanja aktivnosti u ovom području. Svaka od njih predstavlja određenu klasu kretnji s različitom kvalitetom podataka.

Sljedeći izazov u razvoju najsuvremenijih algoritama video analize ostaju skupi računski resursi potrebni za obuku i zaključivanje takvih modela. Iako analiza slika na GPU-ima može potrajati nekoliko dana, trening algoritma za klasifikaciju videozapisa može potrajati tjednima i mjesecima s obzirom na količinu vremenskih podataka (video okvira) za svaki video isječak. Stoga se povećavaju troškovi i vrijeme potrebno za izvođenje značajnih eksperimenata i modeliranja na arhitekturi neuronskih mreža. Stalnim napretkom u GPU tehnologiji i čipovima za zaključivanje specifični za računalni vid poput TensorRT-a tvrtke Nvidia, prepreka za obuku i primjenu ovih modela smanjuje se.

Modeliranje učinkovitih i djelotvornih arhitektura neuronskih mreža može se smatrati najvažnijim dijelom video analize. Uvođenjem konvolucijskih neuronskih mreža napravljen je

veliki skok u području analize i klasifikacije slika. Trenutne arhitekture neuronske mreže za video analizu uvelike se oslanjaju na konvolucijske neuronske mreže zbog sličnosti prostornih aspekata videozapisa sa statičkim slikama. Najnovije istraživanje na ovom polju uključuje vremenske aspekte video zapisa pomoću tehnika kao što su optički vektori protoka, fuzija vremenskih informacija, LSTM-ovi i rekurentne neuronske mreže (RNN). Ovi pristupi zajedno sa konvolucijskim neuronskim mrežama postigli su ogroman napredak. Nedavno provedena istraživanja otvorila su put razvoju nove generacije prostorno-vremenskih algoritama. Novi algoritmi učenja, poput malobrojnog učenja i meta-učenja, također su pokazali znakove optimizma u rješavanju problema nedostatka podataka i učinkovitosti modela.

4.3. Slučajevi za primjenu prepoznavanja ljudskih aktivnosti u proizvodnji

Slučajevi upotrebe prepoznavanja ljudskih aktivnosti podijeljeni su u skupine zadataka koje ljudi izvode u industrijskim okruženjima. Rezultat je sljedeća kategorizacija slučajeva upotrebe koji se odnose na zadatke [18]:

- Zadaci rukovanjem - uobičajeni zadaci radnika u proizvodnji obuhvaćaju upotrebu alata, rad sa strojevima ili rukovanje izrascima (npr. opskrba materijala strojevima). Za zadatke ove skupine karakteristično je to što sadrže mali broj jednostavnih koraka koji se često ponavljaju. Nadalje, informacije o procesu često se mogu dobiti iz informacija o unutarnjem stanju stroja. U slučaju uporabe alata (kao što su pištolji za zavarivanje, pištolji za zakovice, pištolji za ljepilo ili vodilice za matice), položaj alata pruža korisne naznake i za status zadatka.
- Zadatak montaže ili održavanja - zadaci koji pripadaju ovoj kategoriji obično su složeniji od zadataka rukovanjem. Tijekom montažnog zadatka, proizvod se sastavlja od obratka na logički slijedan način. Slično ovome, održavanje se vrši postupno prema planu održavanja. Budući da su moderna radna okruženja visoko strukturirana, za zadatke ove kategorije karakteristično je da se izvode kao niz unaprijed definiranih koraka.
- Nedeterministički stručni zadatak - kategorija neodređenih stručnih zadataka obuhvaća zadatke koji se temelje na nepredvidivom događaju i koji se izvršavaju na način koji ovisi o događaju koji se odvio. Primjeri za to su zadaci popravka (koji se bave

neidentificiranim problemima) ili nadzor procesa. Budući da su ti zadaci izrazito nedeterministički, određeno izvršavanje zadatka nije unaprijed poznato. Stoga su zadaci ove vrste relativno složeni i obično predstavljaju nerješive probleme za prepoznavanje aktivnosti povezanih sa zadacima.

4.3.1. Primjeri prepoznavanja ljudskih aktivnosti u proizvodnji

U prethodnom odlomku objašnjeno je za koje zadatke u proizvodnji se može koristiti postupak prepoznavanja ljudskih aktivnosti, a u nastavku su prikazani konkretni scenariji [18]:

- Scenarij točkastog zavarivanja - reprezentativni scenarij za zadatak rukovanja u automobilskoj industriji je scenarij točkastog zavarivanja. U scenariju se koristi takozvani pištolj za ručno zavarivanje, koji je industrijski alat za ručno otporno zavarivanje. Zadatak radnika u ovom scenariju obuhvaća aktivnosti kao što su postavljanje vrha pištolja za zavarivanje poravnavanjem na određeni položaj ili izvođenje procesa zavarivanja. Scenarij je dio demonstracije u kojoj ljudski radnik preuzima zadatak robota za zavarivanje nakon što dođe do kvara u robotu.
- Scenarij montaže osobnog računala - scenarij montaže osobnog računala reprezentativan je za radničke zadatke u elektroindustriji kategorije sastavljanja ili održavanja. Budući da su elektronički uređaji industrijski proizvodi koji se obično proizvode ručnim sastavljanjem, sastavljanje računala predstavlja tipičan zadatak ove kategorije. Scenarij se sastoji od koraka zadatka u kojima su komponente računala (npr. matična ploča) ugrađene u kućište računala. Ovi koraci zadatka redom sadrže aktivnosti relevantne za izvršavanje zadatka, kao što su podizanje određenih vijaka, postavljanje i pričvršćivanje vijka na ispravnom položaju itd. Sastavljanje računala radnički je zadatak veće složenosti od zadatka točkastog zavarivanja.
- Prepoznavanje gesta rukama - prepoznavanje gesta rukama pruža radniku ugodnu interakciju kako bi dobio pomoć pri radu ili informacije u vezi sa zadacima iz nadređenog okvira. U ovom scenariju treba prepoznati ograničen broj unaprijed definiranih gesta rukama. Zbog činjenice da se interakcija može dogoditi i tijekom izvršavanja zadatka, ovaj se scenarij ne mora nužno tretirati u samostalnoj

demonstraciji, već se također može integrirati u jedan od prethodno spomenutih scenarija prepoznavanja zadatka.

5. MODEL PREPOZNAVANJA LJUDSKIH AKTIVNOSTI U PROIZVODNOM PROCESU

Cilj u ovom dijelu je testirati postojeći model prepoznavanja ljudskih aktivnosti u proizvodnom procesu, iz doktorske disertacije [21], koristeći se realnim podacima iz proizvodnje. Čitav proces detaljno je objašnjen u narednim potpoglavljima. Postupak prikupljanja podataka kao i sam skup podataka preuzeti su iz doktorske disertacije [21] dr. sc. Mihaela Gudlina, dok su svi ostali postupci samostalno izrađeni od strane autora ovog rada.

5.1. Prikupljanje podataka

Prema [21] podaci su prikupljeni u poduzeću Klimaoprema d.d., a prikupljeni skup podataka odnosi se na proces ručne montaže metalnih rešetki HVAC sustava. Podaci su prikupljeni tijekom sedam radnih dana, a snimanje je provedeno u realnim proizvodnim uvjetima. Na slici 16 prikazana je lokacija prikupljanja podataka.



Slika 16 Lokacija prikupljanja podataka [21]

Skup podataka sadrži interakcije između izvoditelja aktivnosti, radnog objekta i alata. Prilikom prikupljanja podataka promatrale su se radne aktivnosti 4 operatera. Operateri su u radu [21] označeni velikim početnim slovom „O“ i brojem, na primjer O1, O2, O3 i O4, pa će se taj princip označavanja koristiti i u nastavku ovog rada. Radne aktivnosti montaže provedene su na tri različita tipa metalnih rešetki koje se razlikuju s obzirom na dimenzije proizvoda. Tipovi proizvoda obilježavani su oznakama T1, T2 i T3 [21]. 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). Za potrebe ovog rada primjenjivati će se kadrovi HE i Fokus.

Na slici 17 prikazan je primjer gdje operater O1 montira proizvod T1, a sniman je kadrom Fokus.



Slika 17 Operater O1 montira proizvod T1 [21]

Prema radu [21], snimanje je provedeno 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 kadra Fokus 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 c) poduzorkovanje je opravdano jer postoji značajna redundantnost između susjednih sličica u video zapisu.

5.1.1. Izrezivanje videozapisa

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 ukupnu veličinu uzorka [21].

5.1.2. Označavanje vrsta aktivnosti

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:

- Formiranje kuta
- Umetanje i učvršćivanje kopče – lijeve gornje
- Umetanje lamela
- Postavljanje poprečne stanice
- Postavljanje uzdužne stanice
- Odlaganje gotovog proizvoda
- Umetanje i učvršćivanje kopče – desne gornje
- Umetanje i učvršćivanje kopče – desne donje
- Umetanje i učvršćivanje kopče – lijeve donje

5.2. Definiranje skupa podataka

Definiranje skupa podataka, kao i svi budući koraci izrađeni su u programskom jeziku Python. Skup podataka definirao se na način da se vršila podjela uzoraka u skup za učenje, validaciju i testiranje. Za ovaj rad definirano je da će se u skupu koji je namijenjen za testiranje nalaziti jedino uzorci operatera O3. Prema tome, u skupu za učenje nalaze se 142 opažanja. Od preostalih 478 opažanja, svojevrijedno je 400 opažanja smješteno u skup za učenje, a 78 u skup za validaciju. Provedena je i stratificirana podjela skupa za učenje i validacijskog skupa s preostalim podacima što je prikazano u tablici 1.1.

Tablica 1 Podjela uzoraka u 3 skupa podataka

Operater_Tip proizvoda	Uzorak [%]	Učenje [%]	Validacija [%]	Testiranje [%]
O1_T1	8.06	10.50	10.26	0
O1_T2	11.45	14.75	15.38	0
O1_T3	10.81	14.00	14.10	0
O2_T1	8.06	10.50	10.26	0
O2_T2	7.90	10.25	10.26	0
O2_T3	8.06	10.50	10.26	0
O3_T1	7.74	0	0	33.80
O3_T2	7.90	0	0	34.50
O3_T3	7.26	0	0	31.70
O4_T1	8.06	10.50	10.26	0
O4_T2	7.74	10.00	10.26	0
O4_T3	6.94	9.00	8.97	0

Iz tablice je vidljivo da je uspješno provedena ideja da se u skupu za testiranje nalaze samo opažanja operatera O3 (postotak učenja i validacije za O3 iznosi 0), kao i stratifikacija pošto su udjeli pojedinih slojeva otprilike jednaki u skupu za učenje i skupu za validaciju.

Idući korak bio je kreiranje oznaka za svaku pojedinu sličicu što je napravljeno na dva načina koja su prikazana na slici 18 na primjeru operatera O1 koji montira proizvod T1.

1. mijenja se dimenzija ulaznih sličica u 224x224 te one ostaju u uint8 (cijeli brojevi od 0 do 255) formatu
2. mijenjaju se dimenzije sličica te se provodi centriranje po kanalima i zamjena redoslijeda kanala usred čega su sličice u float32 formatu (format koji zauzima 32 bita u memoriji računala)

U ovom radu primijenio se prvi način pripreme sličica koji ima manji trag na disku i omogućava korištenje Colab virtualnih instanca.

Slika 19 prikazuje koje funkcije su se koristile u postupku pripreme sličica, a na slici 20 prikazan je čitav kod.

```
##Biblioteke
import cv2
from tensorflow.keras.applications.resnet50 import preprocess_input

##Funkcija za promjenu veličine slike
def resize_img(image_path, img_size=(224, 224)):
    '''Čitanje i promjena veličine slike. Vraća uint8 sliku.'''
    #Čitanje slike
    img = cv2.imread(image_path)
    #Promjena iz BGR u RGB redoslijed kanala
    img = img[:, :, ::-1]
    #Promjena rezolucije u 224x224, zahtjev ResNet50 ulaza
    img = cv2.resize(img, img_size)

    return img

##Funkcija za pripremu slike obradom ResNet50 mrežom
def preprocess_img_for_ResNet50(image_path):
    '''Čitanje i priprema slike za daljnju obradu ResNet50 modelom.
    Vraća float32 ndarray.'''
    #Čitanje slike
    img = cv2.imread(image_path)
    #Promjena iz BGR u RGB redoslijed kanala
    img = img[:, :, ::-1]
    #Promjena rezolucije u 224x224, zahtjev ResNet50 ulaza
    img = cv2.resize(img, (224, 224))
    #ResNet50 predprocesuiranje slike
    img_array = preprocess_input(img)

    return img_array
```

Slika 19 Funkcije za pripremu sličica

Korištene su dvije funkcije: funkcija „resize_img“ za promjenu veličine slike te funkcija „preprocess_img_for_ResNet50“ za pripremu slike obradom ResNet50 mrežom.

```
##Definiranje ulaznih i izlaznih direktorija

#Ulazni direktorij sa slikama
ulazni_dir= config.INPUT_DATASET

#Izlazni direktorij za RESIZED slike
izlazni_dir = config.IMAGES_RESIZED

#Generiranje liste putanja do svih slika
img_paths = glob.glob(os.path.join(ulazni_dir, "**", "*.jpeg"), recursive=True)

#Petlja po putanjama do slika
for img_path in tqdm(img_paths):

    #Procesuiranje slike u np.array za obradu ResNet50 modelom
    img_array = resize_img(img_path)

    #Struktura direktorija za pohranu obrađene slike - reži na separatoru direktorija
    *_ , kadar, podjela_podataka, video_id, _ = img_path.split(os.sep)

    #Odredišni direktorij
    dest_dir = os.path.join(izlazni_dir, kadar, podjela_podataka, video_id)

    #Provjeri da li postoji odredišni direktorij, ako ne kreiraj ga
    if not os.path.exists(dest_dir):
        os.makedirs(dest_dir)

    #Naziv datoteke, izbaci .jpeg ekstenziju
    img_name = os.path.basename(img_path).split(".")[0]

    #Puna putanje za spremanje
    array_path = os.path.join(dest_dir, img_name)

    #Spremanje
    np.save(array_path, img_array)

100%|████████████████████████████████████████| 349147/349147 [4:53:30<00:00, 19.83it/s]
```

Slika 20 Skripta za pripremu sličica

Kao što je vidljivo sa slike, na početku se odredi ulazni direktorij sa slikama te izlazni direktorij u kojem će se nalaziti sličice novih dimenzija. Zatim se odredi petlja po putanjama do slika te se provede spremanje. U posljednjem redu kao izlaz je prikazana obrada da je obrađeno 100% sličica, odnosno njih 349147 za kadrove HE i Fokus.

5.4. Spremanje sličica u Tensorflow Record format

Idući korak je spremanje sličica pripremljenih za obradu ResNet50 modelom u TFRecord (engl. *Tensorflow Record*) format. TFRecord format jednostavan je format za spremanje niza binarnih zapisa. Takvo spremanje nekompatibilno je s drugim čitačima datoteka, ali ima nekoliko osnovnih prednosti [21]:

- zauzima manje prostora na disku, što znači da je svaka operacija koju je potrebno izvesti, bilo čitanja ili pisanja, brža

- potrebno im je manje vremena za kopiranje i mogu se mnogo učinkovitije očitavati s diska
- TFRecord datoteke su optimizirane za rukovanje sastavnim dijelovima većeg skupa podataka
- TFRecord datoteke su optimizirane za pohranjene sekvencirane podatke

Dakle, ako se radi sa velikim skupom podataka, upotreba binarnog formata datoteke za pohranu podataka može imati značajan utjecaj na performanse i vrijeme obuke modela [25].

Dio koda za spremanje sličica u TFRecord format prikazan je na slikama 21 i 22.

```
#Petlja po kadrovima
for kadar in kadrovi:

    #Inicijalizacija brojača sličica u kadru
    num_images = 0
    #Inicijalizacija brojača uzoraka u kadru
    num_samples = 0

    #Petlja po podjelama podataka
    for data_split in data_splits:

        #Trenutne putanje do sličica
        image_path = os.path.join(base_image_dir, kadar, data_split)

        #Petlja po video_id-u
        for video_id in os.listdir(image_path):

            #Putanja do trenutnog videa
            video_path = os.path.join(image_path, video_id)

            #Popis sličica u videu, potrebno ga je SORTIRATI, tako da se poklapa sa oznakama
            image_names = sorted(os.listdir(video_path))

            #Lista oznaka trenutnog videa
            label_path = os.path.join(base_label_dir, data_split, video_id)
            labels = read_labels(label_path)

            #Kreiranje direktorija za izlazne TFRecord
            dest_dir = os.path.join(base_tfrecord_dir, kadar, data_split)
            if not os.path.exists(dest_dir):
                os.makedirs(dest_dir)
```

Slika 21 TFRecord kod - dio 1

Najprije se definiraju bazni direktoriji s lokacijom ulaznih slika i oznaka te odredištem TFRecorda. Nakon toga se odrede kadrovi i način podjele podataka te se raspišu petlje po kadrovima, podjeli podataka i video_id-u.

```
#Definiranje izlazne TF record datoteke
dest_file = os.path.join(dest_dir, video_id)

#Otvaranje TF pisača, sve sličice i oznake jednog videa idu u isti TFRecord
with tf.io.TFRecordWriter(dest_file + '.tfrecord') as writer:

    #Petlja po sličicama i njihovim oznakama
    for image_name, label in zip(image_names, labels):

        #Elementi Example-a
        #Enkodiraj u byte image_id string
        image_id = os.path.join(kadar, data_split, video_id, image_name.split(".")[0]).encode()
        image = np.load(os.path.join(video_path, image_name))

        #Kreiranje serijaliziranog Example-a
        example = image2example(image_id, image, label)
        #Zapiši ga
        writer.write(example)

        #Ažuriraj brojač sličica
        num_images += 1

    #Ažuriraj brojač opažanja
    num_samples += 1

#Ispiši informativnu poruku
print(f"[INFO] obrađen je kadar {kadar}, koji sadrži {num_samples} video zapisa i {num_images} sličica.")
```

[INFO] obrađen je kadar HE, koji sadrži 620 video zapisa i 174573 sličica.
[INFO] obrađen je kadar Fokus, koji sadrži 620 video zapisa i 174573 sličica.

Slika 22 TFRecord kod - dio 2

U nastavku koda je prikazano definiranje izlazne TFRecord datoteke, otvaranje TF pisača te petlja po sličicama i njihovim oznakama. Na kraju koda je zadana naredba da se ispiše poruka kako bi bili sigurni da su obrađeni svi podaci koji su nam potrebni. Vraćena poruka govori kako je obrađen kadar HE koji sadrži 620 video zapisa i 174573 sličica te kadar Fokus sa istim brojem video zapisa i sličica, što znači da su naši zahtjevi ispunjeni. Izlaz su pripremljene sličice spremljene u TFRecord formatu.

5.5. Izvlačenje prostornih značajki

Izvlačenje značajki (*engl. feature extraction*) je proces smanjenja dimenzionalnosti kojim se početni skup neobrađenih podataka svodi na grupe kojima je lakše upravljati za obradu. Karakteristika ovih velikih skupova podataka je veliki broj varijabli koje zahtijevaju puno računalnih resursa za obradu. Izvlačenje značajki ujedno je naziv za metode koje odabiru i/ili kombiniraju varijable u značajke, učinkovito smanjujući količinu podataka koji se moraju obraditi, dok još uvijek točno i potpuno opisuju izvorni skup podataka.

Proces izdvajanja značajki koristan je kada treba smanjiti broj resursa potrebnih za obradu bez gubitka važnih ili relevantnih informacija. Ono također može smanjiti količinu suvišnih podataka za datu analizu. Isto tako, smanjenje podataka i napora stroja u izgradnji varijabilnih

kombinacija (značajki) olakšava brzinu učenja i korake generalizacije u procesu strojnog učenja.

Izdvajanje prostornih značajki često svoju primjenu pronalazi u obradi slika gdje se algoritmi koriste za otkrivanje značajki poput oblika, rubova ili kretnji u digitalnoj slici ili videu [25].

U doktorskoj disertaciji [21] napravljena je iduća podjela koja se odnosi na način pripreme prostornih značajki:

- engl. *feature extraction* – čisti izračun značajki predtreniranom ResNet50 mrežom
- engl. *transfer learning base model* – izračun značajki s fino podešenom ResNet50 mrežom
- engl. *train base model* – izračun značajki s modelom temeljenim na rezidualnim blokovima potpuno naučenim na vlastitim podacima

U ovom radu provodi se izvlačenje prostornih značajki primjenom predtrenirane ResNet50 mreže što je prikazano na slici 23.

```
#Petlja po id-ovima slika te slikama, oznake nas ne zanimaju kod izvlačenja značajki
for image_ids, images, _ in tqdm(feats_dataset):
    #Izvlačenje značajki za grupu slika
    batch_features = feature_extraction_model(images)
    #Osiguranje da su dimenzije izvučenih značajki (br.uzoraka x 2048)
    batch_features = tf.reshape(batch_features, (batch_features.shape[0], -1))
    #Petlja po id-ovima slika i izvučenim značajkama
    for image_id, features in zip(image_ids.numpy(), batch_features.numpy()):
        #Izvuci video_id iz image_id-a, ostalo nas ne zanima (image_id => kadar/data_split/video_id/image_name)
        _, _, video_id, _ = image_id.decode().split(os.sep)
        #U slučaju da se promijenio video iz čijih slika izvlačimo značajke,
        #trebamo zapisati sve značajke iz spremnika current_video_features
        if video_id != current_video_id and current_video_id is not None:
            #Potrebne komponente iz current_image_id-a
            kadar, data_split, _, _ = current_image_id.decode().split(os.sep)
            #Struktura izlaznog direktorija
            output_dir = os.path.join(base_feats_dir, kadar, data_split)
            if not os.path.exists(output_dir):
                os.makedirs(output_dir)
            #Izlazna datoteka za video prije promjene id-a
            output_file = os.path.join(output_dir, current_video_id + ".npy")
            np.save(output_file, current_video_features)
            #Bitan korak je nakon što zapišemo podatke, da ponovno inicijaliziramo spremnik
            current_video_features = []
        #Zapiši koji je video trenutno u obradi i sličica, te dodaj njegove značajke u spremnik
        current_video_id = video_id
        current_image_id = image_id
        current_video_features.append(features)
    #Zapiši i zadnji video
    kadar, data_split, _, _ = current_image_id.decode().split(os.sep)
    output_dir = os.path.join(base_feats_dir, kadar, data_split)
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)
    output_file = os.path.join(output_dir, current_video_id + ".npy")
    np.save(output_file, current_video_features)
```

21822it [54:51:58, 9.05s/it]

Slika 23 Izvlačenje prostornih značajki

Početni koraci u postupku izvlačenja prostornih značajki slični su kao i u prethodnim postupcima. Najprije se definira model sa završnim slojem sažimanja srednjih vrijednosti (engl. *average pooling*) koji se koristi s ciljem progresivnog smanjenja veličine slike, samim time i broja karakteristika, što dovodi do smanjenja kompleksnosti računanja i brže treniranje mreže. Zatim se odredi ulazni direktorij sa sličicama promijenjenih veličina te bazni direktorij sa određenom lokacijom značajki kako bi se napisala petlja za izvlačenja značajki.

Svi ostali potrebni koraci za izvlačenje značajki prikazani su na slici iznad, a započinju petljom po id-ovima slika i slikama (oznake nisu važne u postupku izvlačenja značajki). Izlaz su prostorne značajke dimenzija 2048 u ndarray formatu tipa float32.

5.6. Spremanje značajki u Tensorflow Record format

TFRecord format objašnjen je u poglavlju 5.4., a nakon spremanja sličica u taj format, isto je potrebno učiniti i sa izvučenim značajkama.

Postupak spremanja značajki prikazan je na slikama 24 i 25.

Postupak započinje određivanjem baznog direktorija s lokacijom ulaznih video značajki i oznaka te odredištem TFRecord-a, baznog direktorija s lokacijama oznaka te baznog direktorija s odredištem TFRecord-a. Određuje se i maksimalan broj opažanja za train, val i test skupove u TFRecord-ima. Maksimalan broj opažanja za train iznosi 60, a za val i test 70. Brojke su temeljene na preporukama o veličini TFRecord-a.

```

kadrovi = ["HE", "Fokus"]
data_splits = ["train", "val", "test"]

#Petlja po kadrovima
for kadar in kadrovi:
    #Petlja po podjelama podataka
    for data_split in data_splits:
        #Trenutna putanja
        current_path = os.path.join(base_video_dir, kadar, data_split)
        #Generiraj listu videa u odgovarajućem kadru i podjeli podataka
        video_list = os.listdir(current_path)
        #Maksimalan broj opažanja u jednom TFRecord-u
        max_examples = max_examples_per_split[data_split]
        #Izračunaj broj TFRecorda za data split
        num_records = int(np.ceil(len(video_list) / max_examples))
        #Generiraj grupe videa koje idu u isti TFRecord
        video_batches = [video_list[(idx * max_examples):(idx + 1) * max_examples] for idx in range(num_records)]
        #Petlja po grupama videa
        for tfr_id, video_batch in enumerate(video_batches, start=1):
            #Kreiranje direktorija za izlazne TFRecord-e
            dest_dir = os.path.join(base_tfr_dir, kadar, data_split)
            if not os.path.exists(dest_dir):
                os.makedirs(dest_dir)
            #Kreiranje naziva datoteke izlaznog TFRecorda
            dest_file = os.path.join(dest_dir, f"{tfr_id}_grupa.tfrecord")
            #Otvoravanje TF pisaača, više videa i oznaka ide u isti TFRecord
            with tf.io.TFRecordWriter(dest_file) as writer:
                #Petlja po video zapisima u grupi videa
                for video in video_batch:
                    #Izvuci video id
                    video_id = video.split(".")[0]
                    #Putanja do trenutnog videa
                    video_path = os.path.join(current_path, video)
                    #Putanja do oznaka trenutnog videa
                    labels_path = os.path.join(base_label_dir, data_split, video_id)
                    #Učitaj video i oznake
                    video_images = np.load(video_path)
                    labels = read_labels(labels_path)

```

Slika 24 TFRecord spremanje značajki – dio 1

Na slici je prikazano koji kadrovi se koriste te kakva podjela podataka, nakon čega je raspisana petlja po tom kadru i podjeli podataka. Također, raspiše se petlja po grupama videa te se kreiraju direktoriji i nazivi datoteka za izlazne TFRecord-e.

```

#Izvuoci informacije o broju slika i oznaka i potvrdi da su jednake
num_images = video_images.shape[0]
num_labels = len(labels)
assert num_images == num_labels, "Broj sličica u video zapisu i oznaka je jednak"
#Kreiraj serijalizirani SequenceExample
example = video2example(video_id.encode(), video_images, labels, num_images, num_labels)
#Zapiši ga
writer.write(example)

#Ispiši informativnu poruku
print(f"[INFO] Zapisana je {tfr_id}. grupa za {kadar}/{data_split} podatke.")

```

```

[INFO] Zapisana je 1. grupa za HE/train podatke.
[INFO] Zapisana je 2. grupa za HE/train podatke.
[INFO] Zapisana je 3. grupa za HE/train podatke.
[INFO] Zapisana je 4. grupa za HE/train podatke.
[INFO] Zapisana je 5. grupa za HE/train podatke.
[INFO] Zapisana je 6. grupa za HE/train podatke.
[INFO] Zapisana je 7. grupa za HE/train podatke.
[INFO] Zapisana je 1. grupa za HE/val podatke.
[INFO] Zapisana je 2. grupa za HE/val podatke.
[INFO] Zapisana je 1. grupa za HE/test podatke.
[INFO] Zapisana je 2. grupa za HE/test podatke.
[INFO] Zapisana je 3. grupa za HE/test podatke.
[INFO] Zapisana je 1. grupa za Fokus/train podatke.
[INFO] Zapisana je 2. grupa za Fokus/train podatke.
[INFO] Zapisana je 3. grupa za Fokus/train podatke.
[INFO] Zapisana je 4. grupa za Fokus/train podatke.
[INFO] Zapisana je 5. grupa za Fokus/train podatke.
[INFO] Zapisana je 6. grupa za Fokus/train podatke.
[INFO] Zapisana je 7. grupa za Fokus/train podatke.
[INFO] Zapisana je 1. grupa za Fokus/val podatke.
[INFO] Zapisana je 2. grupa za Fokus/val podatke.
[INFO] Zapisana je 1. grupa za Fokus/test podatke.
[INFO] Zapisana je 2. grupa za Fokus/test podatke.
[INFO] Zapisana je 3. grupa za Fokus/test podatke.

```

Slika 25 TFRecord spremanje značajki - dio 2

Na ovoj slici raspisana je petlja po videozapisima u grupi videa. Postavljen je uvjet da broj sličica i oznaka mora biti jednak te je zadan izlaz u kojem se mora ispisati koja je TFRecord grupa zapisana, za koji kadar i u kojem skupu podataka.

Vidljivo je kako je, za kadrove HE i Fokus, zapisano sedam grupa u train skupu podataka, dvije grupe u val skupu podataka i tri grupe u test skupu podataka. Izlaz su značajke dimenzija 2048 izračunate za kadrove HE i Fokus.

5.6.1. Konkatencija značajki

Konkatencija se može objasniti kao združivanje sličica videozapisa s istim id-om za kadrove HE i Fokus. Primjerice, spajanje dva vektora u jedan može se smatrati konkatencijom. Vektor $x = (1,2,3)$ i vektor $y = (4,5,6)$ bi konkatencijom kreirali vektor $z = (1,2,3,4,5,6)$. U ovom slučaju se provodi postupak konkatencije kako bi model koristio informacije iz oba kadra „istovremeno“ tj. da koristi fino fokusirane informacije (Fokus) te širu sliku (HE). Točnije, ideja je da kadar Fokus daje fine informacije o aktivnostima koje se izvode, a kadar HE informacije o interakciji s cijelom radnom okolinom.

Postupak konkatencije prikazan je na slikama 26 i 27.

U početnom koraku se ponovno odrede svi bazni direktoriji s lokacijom video ulaznih značajki te se postavi maksimalan broj opažanja u pojedinom skupu. Vrijednosti su, u odnosu na prethodni slučaj, prepolovljene te sada maksimalan broj opažanja za train skup iznosi 30, dok za val i test skup iznos 35.

```

kadrovi = ["HE", "Fokus"]
data_splits = ["train", "val", "test"]
#Petlja po podjelama podataka
for data_split in data_splits:
    #Trenutna putanja
    current_path_HE = os.path.join(base_video_dir, "HE", data_split)
    current_path_Fokus = os.path.join(base_video_dir, "Fokus", data_split)
    #Generiraj listu videa u odgovarajućoj podjeli podataka (ista je lista, neovisno od kadra)
    video_list = os.listdir(current_path_HE)
    #Maksimalan broj opažanja u jednom TFRecord-u
    max_examples = max_examples_per_split[data_split]
    #Izračunaj broj TFRecords za data split
    num_records = int(np.ceil(len(video_list) / max_examples))
    #Generiraj grupe videa koje idu u isti TFRecord
    video_batches = [video_list[(idx * max_examples):(idx + 1) * max_examples] for idx in range(num_records)]
    #Petlja po grupama videa
    for tfr_id, video_batch in enumerate(video_batches, start=1):
        #Kreiranje direktorija za izlazne TFRecord-e
        dest_dir = os.path.join(base_tfreord_dir, "concat", data_split)
        if not os.path.exists(dest_dir):
            os.makedirs(dest_dir)
        #Kreiranje naziva datoteke izlaznog TFRecorda
        dest_file = os.path.join(dest_dir, f"{tfr_id}_grupa.tfreord")
        #Otvoravanje TF pisaača, više videa i oznaka ide u isti TFRecord
        with tf.io.TFRecordWriter(dest_file) as writer:
            #Petlja po video zapisima u grupi videa
            for video in video_batch:
                #Izvuci video_id
                video_id = video.split(".")[0]
                #Putanja do trenutnog videa iz oba kadra
                video_path_HE = os.path.join(current_path_HE, video)
                video_path_Fokus = os.path.join(current_path_Fokus, video)
                #Putanja do oznaka trenutnog videa
                labels_path = os.path.join(base_label_dir, data_split, video_id)
                #Učitaj videoe za oba kadra i oznake
                video_images_HE = np.load(video_path_HE)
                video_images_Fokus = np.load(video_path_Fokus)
                labels = read_labels(labels_path)

```

Slika 26 Konkatencija - dio 1

Nakon što se odrede kadrovi i podjela podataka, generira se lista videozapisa u odgovarajućoj podjeli podataka. Lista je ista, neovisno od kadra, pa je dovoljno provesti generiranje samo za jedan kadar. Zatim se napišu naredbe za maksimalan broj opažanja te izračun broja TFRecord-a, nakon čega se raspiše petlja po grupama videozapisa. Unutar te petlje se kreiraju direktoriji za izlazne TFRecord-e (concat) te nazivi datoteka izlaznih TFRecorda.

```

#Konkateniraj značajke videa po vremenskoj osi - svaki vremenski korak sada ima dim 4096
concat_video_images = np.concatenate((video_images_HE, video_images_Fokus), axis=1)
#Izvući informacije o broju slika i oznaka i potvrdi da su jednake
num_images = video_images_HE.shape[0]
num_labels = len(labels)
assert num_images == num_labels, "Broj sličica u video zapisu i oznaka mora biti jednak!"
#Kreiraj serijalizirani SequenceExample
example = video2example(video_id.encode(), concat_video_images, labels, num_images, num_labels)
#Zapiši ga
writer.write(example)
#Ispiši informativnu poruku
print(f"[INFO] Zapisana je {tfr_id}. grupa za {data_split} podatke.")

```

```

[INFO] Zapisana je 1. grupa za train podatke.
[INFO] Zapisana je 2. grupa za train podatke.
[INFO] Zapisana je 3. grupa za train podatke.
[INFO] Zapisana je 4. grupa za train podatke.
[INFO] Zapisana je 5. grupa za train podatke.
[INFO] Zapisana je 6. grupa za train podatke.
[INFO] Zapisana je 7. grupa za train podatke.
[INFO] Zapisana je 8. grupa za train podatke.
[INFO] Zapisana je 9. grupa za train podatke.
[INFO] Zapisana je 10. grupa za train podatke.
[INFO] Zapisana je 11. grupa za train podatke.
[INFO] Zapisana je 12. grupa za train podatke.
[INFO] Zapisana je 13. grupa za train podatke.
[INFO] Zapisana je 14. grupa za train podatke.
[INFO] Zapisana je 1. grupa za val podatke.
[INFO] Zapisana je 2. grupa za val podatke.
[INFO] Zapisana je 3. grupa za val podatke.
[INFO] Zapisana je 1. grupa za test podatke.
[INFO] Zapisana je 2. grupa za test podatke.
[INFO] Zapisana je 3. grupa za test podatke.
[INFO] Zapisana je 4. grupa za test podatke.
[INFO] Zapisana je 5. grupa za test podatke.

```

Slika 27 Konkatenacija - dio 2

Iz ove slike je vidljiva naredba za konkateniranje značajki te ponovni uvjet da broj sličica i oznaka u zapisu mora biti jednak. Kao ispis prikazan je koliko grupa je zapisano za pojedini skup podataka. Tako je zapisano 14 grupa za train skup podataka, 3 grupe za val skup podataka te 5 grupa za test skup podataka. Dakle, u ovom slučaju, izlaz su značajke koje su konkatenacija značajki kadra HE i Fokus dimenzija 4096.

5.7. Učenje modela

Prema radu [21] najzastupljenije arhitekture za učenje modela su:

- LSTM (engl. *Long Short-Term Memory*) model – umjetna rekurentna arhitektura neuronske mreže s povratnim vezama koja se koristi za duboko učenje
- Dvosmjerni LSTM (engl. *Bidirectional Long Short-Term Memory*) model – model obrade sekvence koji se sastoj od dva LSTM-a: jedan uzima ulaz u smjeru unaprijed, a drugi unatrag
- Konvolucijski više etapni (engl. *Multi-Stage Convolutional Neural Network*) model – konvolucijski više etapni model pruža mogućnost konstruiranja više etapa modela

neuronske mreže. Primjerice, ako se radi o modelu s dvije etape, u prvoj etapi neuronske mreže se mogu koristiti za generiranje nekog skupa podataka, dok onda u drugoj etapi model neuronske mreže provodi radnje na temelju rezultata generiranja iz prve etape.

Za potrebe ovog rada odabran je konvolucijski više etapni model.

Postupak učenja modela započinje određivanjem tipa modela (CONV), tipa značajki (feat extraction) i kadra (Fokus). Zatim se odrede putanje za pohranu modela, rezultata metrike i vizualizacije krivulja. Kreira se direktorij za pohranu te se odrede hiperparametri učenja gdje se postavlja broj epoha za učenje na 80. Broj epoha je hiperparametar koji definira koliko će puta algoritam učenja raditi kroz cijeli skup podataka za učenje (train). Jedna epoha znači da je svaki uzorak u skupu podataka za učenje imao priliku ažurirati interne parametre modela [26].

Također, potrebno je odrediti lokaciju ulaznih podataka (učenje se provodi za oba kadra HE i Fokus) te generirati train i val skup podataka. Treba obratiti pažnju da je za konkatenciju kadrova dimenzija značajki 4096, dok je za ostale kadrove 2048. Ostatak koda za učenje prikazan je na slici 28.

```

###Definiranje modela za učenje
print("[INFO] priprema modela")
tf.keras.backend.clear_session()
#Komponente modela
if MODEL_TYPE == "CONV":
    model = build_ms_tcn_model(input_shape=(None, feature_dim[KADAR]), num_layers_PG=5, R_stages=1,
                              num_layers_R=5, filters=64, training=True, dropout_rate=0.2, shared_R=False)
else:
    model = build_conv_model(input_shape=(None, feature_dim[KADAR]), mask_value=0., bidirect=False,
                             num_lstm_layers=1,
                             lstm_units=256, lstm_dropout=0.5, lstm_recurrent_dropout=0.,
                             num_fc_layers=1, fc_reg=0.005, fc_units=256, fc_dropout=0.5,
                             num_classes=10, seed=42)
###Kompajliranje modela za fino podešavanja
optimizer = tf.keras.optimizers.SGD(learning_rate=INIT_LR, momentum=0.9)#decay=5e-2
loss = tf.keras.losses.SparseCategoricalCrossentropy()
#loss = SegmentationLoss()
model.compile(optimizer=optimizer, loss=loss, metrics=["accuracy"])
###Traženje optimalne stope učenja
lr_finder = LearningRateFinder(model, stop_factor=1.5, smoothing_factor=0.9)
lr_finder.find_lr(train_dataset, y=None, num_samples=400,
                 batch_size=BATCH_SIZE, epochs=5, min_lr=1e-5, max_lr=1)
###Definiranje alata za pohranu modela i nadzor učenja
#Checkpointer modela
cm = EpochCheckpoint(output_path=MODEL_PATH, every=40, start_at_epoch=START_EPOCH)
#Trening monitor
#MultiOut trening monitor
tm = MultiOutTrainingMonitor(plot_path=PLOT_PATH, history_path=HISTORY_PATH, start_at_epoch=START_EPOCH)
#Raspored stope učenja - ciklička stop učenja
schedule = CyclicLR(base_lr=1e-4, max_lr=1e-2, step_size=60)
###Učenje modela
print("[INFO] model je u procesu učenja")
model.fit(train_dataset, epochs=EPOCHS, callbacks=[cm, tm,schedule], validation data=val_dataset)

```

Slika 28 Učenje modela

Na slici je prikazano definiranje modela za učenje. Postavljene su komponente modela, potražene su optimalne stope učenja te su definirani alati za pohranu modela i nadzor učenja. Broj slojeva modela postavljen je na 5, a za postupak optimizacije koristi se SGD (engl. *Stochastic Gradient Descent*) metoda. Radi se o iterativnoj metodi za optimizaciju ciljne funkcije s prikladnim svojstvima gustoće. Može se smatrati stohastičkom aproksimacijom optimizacije gradijentnog spuštanja, jer stvarni gradijent zamjenjuje njegovom procjenom [29]. Optimalna granica učenja određena je pomoću LRF (engl. *Learning Rate Finder*) metode. Takva metoda se najjednostavnije može opisati kao parametar podešavanja u algoritmu optimizacije koji određuje veličinu koraka pri svakoj iteraciji dok se kreće prema minimalnoj funkciji gubitka [30].

Na slici 29 prikazan je dobiveni izlaz za postupak učenja modela.


```
[INFO] priprema ulaznog pipeline-a
[INFO] priprema modela
[INFO] tražim optimalne granice stope učenja...
Epoch 1/5
13/13 [=====] - 102s 2s/step - loss: 8.9722 -
prediction_generation_loss: 7.2705 - refinement_loss: 1.7017 -
prediction_generation_accuracy: 0.1170 - refinement_accuracy: 0.1180
Epoch 2/5
13/13 [=====] - 44s 1s/step - loss: 6.2799 -
prediction_generation_loss: 4.6069 - refinement_loss: 1.6729 -
prediction_generation_accuracy: 0.1703 - refinement_accuracy: 0.1357
Epoch 3/5
13/13 [=====] - 30s 1s/step - loss: 3.3933 -
prediction_generation_loss: 1.8318 - refinement_loss: 1.5615 -
prediction_generation_accuracy: 0.4046 - refinement_accuracy: 0.2097
Epoch 4/5
1/13 [=>.....] - ETA: 2:17 - loss: 6.9502 -
prediction_generation_loss: 5.4522 - refinement_loss: 1.4980 -
prediction_generation_accuracy: 0.1872 - refinement_accuracy: 0.1665
[INFO] gubitak je veći od izračunate gornje granice, prekidam učenje
13/13 [=====] - 11s 666us/step - loss: 6.9502
prediction_generation_loss: 5.4522 - refinement_loss: 1.4980 -
prediction_generation_accuracy: 0.1872 - refinement_accuracy: 0.1665
[INFO] postupak traženja optimalnih granica stope učenja završen
```

Slika 29 Priprema modela za učenje

Unutar određivanja optimalne granice učenja, broj uzoraka postavljen je na 400 što odgovara broju uzoraka u skupu za učenje, dok je broj epoha 5. Tijekom prve tri epohe gubitak je u konstantnom padu, dok u četvrtoj epohi raste na 6.9502 te se učenje prekida jer je gubitak veći od izračunate gornje granice.

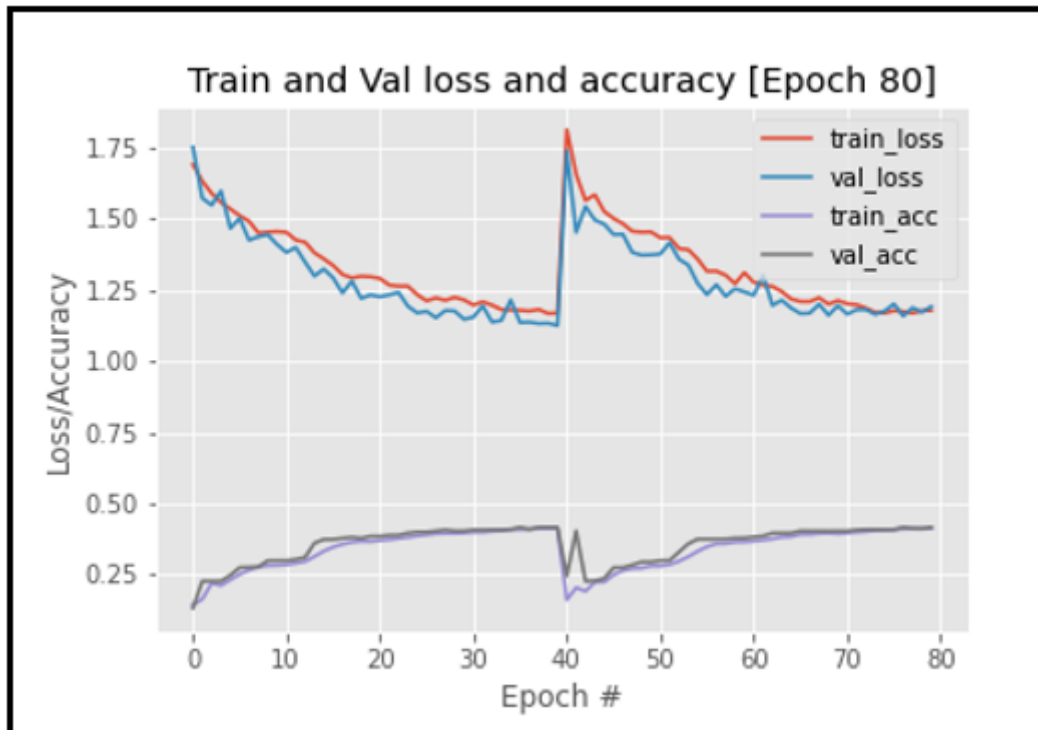
U tablici 2 prikazan je proces učenja modela s rezultatima koji su bitni za ovaj rad. U prvom stupcu nalazi se broj epoha, a zatim redom slijede gubitak i točnost za skup train te gubitak i točnost za skup val. Kako je broj epoha postavljen na 80, u tablici su prikazani rezultati za posljednjih 10 epoha što je dovoljno za primjećivanje trenda rasta ili pada.

Tablica 2 Proces učenja modela

BROJ EPOHA	TRAIN GUBITAK	TRAIN TOČNOST	VAL GUBITAK	VAL TOČNOST
71	1.2022	0.3966	1.1656	0.4035
72	1.1987	0.3996	1.1811	0.4071
73	1.1869	0.4017	1.1819	0.4086
74	1.1720	0.4058	1.1648	0.4088
75	1.1695	0.4069	1.1763	0.4085
76	1.1775	0.4082	1.2012	0.4086
77	1.1750	0.4112	1.1588	0.4161
78	1.1693	0.4117	1.1880	0.4124
79	1.1774	0.4114	1.1729	0.4123
80	1.1786	0.4129	1.1919	0.4159

Iz tablice je vidljivo kako se povećanjem broja epoha, odnosno povećanjem broja prolazaka algoritma kroz skup za učenje, smanjuje gubitak, a povećava točnost. Isto vrijedi i u slučaju validacije. Nakon 80 epoha gubitak za train skup iznosi 1.1786, dok je točnost 0.4129, odnosno 41.29%. U slučaju val skupa gubitak iznosi 1.1919, a točnost 0.4159, odnosno 41.59%. Točnost na validacijskom skupu je nešto veća nego što je to na trening skupu, a razlog tome je što se u procesu učenja koriste regularizacijske tehnike poput „dropout“ sloja.

Na slici 30 prikazan je graf gubitka i točnosti iz kojeg se može vidjeti zašto je odabrano da broj epoha iznosi 80.



Slika 30 Graf učenja modela

Slika prikazuje kako se kreću vrijednosti točnosti i gubitka za validacijski i trening skup. Jasno je vidljivo kako se za oba skupa, najniža vrijednost točnosti, a najveća vrijednost gubitka postiže kada je broj epoha 40. Suprotno tome, najveća vrijednost točnosti, a najmanja vrijednost gubitka ostvaruje se kada je broj epoha iznosio 80. Nakon te vrijednosti stagnira rast točnosti i pad gubitka te se iz tog razloga za konačan broj epoha odabire 80.

5.8. Evaluacija modela

U ovom dijelu rada biti će prikazani rezultati evaluacije modela na skupu podataka za testiranje. Provođi se postupak izračuna četiri tipa metrike za model na definiranoj podjeli podataka (test, train, val) koje služe za ocjenjivanje modela. Izračunate metrike su sljedeće [21]:

- točnost po pojedinačnoj sličici (engl. *framewise accuracy*) – mjeri udio točno klasificiranih sličica u video zapisu u odnosu na ukupan broj sličica

- srednje apsolutno odstupanje (engl. *Mean Absolute Deviation MAD*) vremena trajanja aktivnosti – uprosječno po vrsti aktivnosti i broju opažanja
- segmentacijski F1 rezultat uz definiran prag preklapanja (10%, 25%, 50%) (engl. *F1@Intersection over Union IoU*) – prikladna metrika za evaluaciju modela jer kažnjava greške prekomjerne segmentiranosti, ne kažnjava manje pomake u vremenu koji mogu biti posljedica varijabilnosti i subjektivnosti kod označavanja te daje rezultate koji ovise o broju aktivnosti, a ne njihovom trajanju [28]
- srednja prosječna preciznost uz definiran prag preklapanja (10%, 25%, 50%) (engl. *mean average precision mAP@IoU*) [28]

Prilikom pisanja koda za evaluaciju modela ulaz je putanja do modela te podjela podataka za koju želimo rezultate. Iz putanje modela izvlače se informacije o ulaznim značajkama koje se prosljeđuju modelu za daljnju obradu. Struktura modela je „tip modela/tip značajke/kadar“. To u ovom slučaju znači da je konvolucijski model (CONV) naučen na značajkama izvučenim iz ResNet 50 mreže bez finog podešavanja (feature extraction) na kadru Fokus te će to biti ulazne značajke za evaluaciju modela. Izlaz je izračunata metrika u txt formatu pohranjena u određenoj lokaciji koja se ispisiuje po izračunu metrike.

Dobivena metrika za sva 3 skupa prikazana je u tablici 3.

Tablica 3 Rezultati metrike

Metrika/skup podataka	train	val	test
„Frame-wise“ točnost [%]	42.188	41.603	34.239
MAD vremena trajanja aktivnosti [s]	7.097	7.140	11.185
Segmentacijski F1 rezultat F1@10	51.612	50.881	41.229
Segmentacijski F1 rezultat F1@25	45.714	43.595	33.514
Segmentacijski F1 rezultat F1@50	27.309	26.086	20.132
Srednja prosječna preciznost mAP@10	49.443	48.745	40.971
Srednja prosječna preciznost mAP@25	40.888	39.622	31.341
Srednja prosječna preciznost mAP@50	23.579	21.803	14.86

Iz tablice je jasno vidljivo kako rezultati metrike test skupa znatno odskoču od rezultata metrike skupova train i val. Točnost od 34.239% znatno je niža od približno 42% koliko iznosi točnost skupova train i val. Vrijeme trajanja aktivnosti je znatno duže (11.18) u odnosu na preostala dva skupa (7). Segmentacijski F1 rezultati, kao i srednja prosječna vrijednost, s preklapanjima od 10, 25 i 50% također su u značajnijem padu u odnosu na train i val skup. Očekivano je da su rezultati test skupa niži s obzirom da je model testiran na operateru koji nije bio ni u train ni u val skupu (operater O3). Jedna od glavnih ideja je bila ispitati koliko model može generalizirati van distribucije, a rezultati ukazuju na to da je ipak potrebno koristiti podatke o svim operaterima kako bi se dobio praktično koristan model.

5.8.1. Analiza modela

Točnost modela (CONV) naučenog na značajkama izvučenim iz ResNet 50 mreže bez finog podešavanja (feature extraction) na kadru Fokus iznosi 41,6% uz F1@50 od 26%. Manja vrijednost točnosti se također može objasniti primjenom izvlačenja značajki bez finog podešavanja. Takav postupak omogućava brzo izvlačenje značajki slabije kvalitete, u smislu da je finalni model naučen na njima manje učinkovit u odnosu na druge modele koji koriste značajke dobivene nekim od preostala dva pristupa. S druge strane očekivano je da će model za izvlačenje značajki razvijen na vlastitim podacima dati najbolje značajke, ali će proces učenja i izvlačenja značajki biti znatno duži u odnosu na pristup korišten u ovom radu [21].

Model temeljen na LSTM ili dvosmjernim LSTM slojevima vjerojatno bi ostvario veću točnost, ali bi bio teži i sporiji za učenje od konvolucijskog modela u ovom radu. Shodno tome, očekivano je da je konvolucijski model brži u učenju, ali i potencijalno neprecizniji. Iz tog razloga je često 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.

6. ZAKLJUČAK

Umjetna inteligencija u sklopu industrije 4.0 ima sposobnost u budućnosti u potpunosti transformirati, ne samo industriju proizvodnje kao takve, već čitavo okruženje u kojem čovjek djeluje. U ovom radu nastojala se prikazati primjena umjetne inteligencije u svim sferama života, s posebnim naglaskom na proizvodnju, a s ciljem da se približe njene snage i mogućnosti, kao i eventualne mane. S gledišta proizvodnje nastojalo se objasniti zašto je dobro implementirati aplikacije umjetne inteligencije u poduzeće, ali isto tako i s kojim problemima se prilikom tog postupka susreću rukovoditelji mnogih poduzeća.

Detaljno su objašnjeni najbitniji dijelovi i alati umjetne inteligencije kao što su strojno i duboko učenje, umjetne neuronske mreže i konvolucijske neuronske mreže kojim se nastojalo pružiti uvod u primjenu umjetne inteligencije za postupke otkrivanja i prepoznavanja kretnji u videozapisu.

Fokus ovog rada bio je prikazati primjenu umjetne inteligencije u svrhu prepoznavanja ljudskih aktivnosti u proizvodnji. Postavljeni cilj bio je da se na temelju podataka prikupljenih iz realnog proizvodnog procesa testira model temeljen na računalnom vidu i dubokom strojnom učenju koji ima sposobnost prepoznavanja niza ljudskih aktivnosti, a čijom se primjenom mogu unaprijediti postupci studija vremena i analize produktivnosti ljudskog faktora. Odlika kreiranog uzorka je ta da je sačinjen od podataka prikupljenih u realnim proizvodnim uvjetima, na različitim tipovima proizvoda i od strane različitih izvođača čime su stvorene dobre pretpostavke za ispitivanje objektivne primjenjivosti modela dubokog strojnog učenja. U ovom slučaju testiran je konvolucijski model (CONV) naučen na značajkama izvučenim iz ResNet 50 mreže bez finog podešavanja (feature extraction) na kadru Fokus, a čitav postupak detaljno je objašnjen u poglavlju 5. Na temelju dobivenih rezultata može se smatrati da je postavljeni cilj ipak ispunjen. Konkretno, model je ostvario točnost od 41,6% uz F1 rezultat od 26%. Ti rezultati se u nekim daljnjim eksperimentima mogu poboljšati na način da se provedu eksperimenti s kombinacijama različitih kadrova snimanja (Fokus, HE), pristupa izvlačenju značajki (feature extraction, transfer learning model, train base model), i modela (LSTM, dvosmjerni LSTM, konvolucijski). Testiranjem više modela i njihovom međusobnom usporedbom moguće je ostvariti bolje rezultate. Ipak, uzimajući u obzir dobiveni rezultat može se reći da su stvorene pretpostavke za unaprjeđenje

postojećih pristupa studiju vremena primjenom testiranog modela.

LITERATURA

- [1] Elements of AI, <https://course.elementsofai.com/hr/> (3.10.2021.)
- [2] Platform Industrie 4.0: Technology Scenario ‘Artificial Intelligence in Industrie 4.0’. 2019 Federal Ministry for Economic; March; 2019
- [3] Industry 4.0 and the fourth industrial revolution explained, <https://www.i-scoop.eu/industry-4-0/> (4.10.2021.)
- [4] J.Mattioli; P.Perico; P.Robic: Improve Total Production Maintenance with Artificial Intelligence. 2020 Third International Conference on Artificial Intelligence for Industries (AI4I)
- [5] M.Haenlein; A.Kaplan: A Brief History Of Artificial Intelligence: On the Past, Present and Future of Artificial Intelligence. The Regents of the University of California 2019
- [6] E.Balamurugan; L.R.Flaih; D.Yuvaraj: Use Case of Artificial Intelligence in Machine Learning Manufacturing 4.0. 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE); December 11-12, 2019, Amity University Dubai, UAE
- [7] Davenport, Thomas; Ronanki, Rajeev: Artificial Intelligence for the Real World. 2018 Harvard Business Review; January-February, 2018
- [8] Charalambous, Eleftherios; Feldmann, Robert; Richter, Gerard; Schmitz, Christoph: AI in production: A game changer for manufacturers with heavy assets. 2019 McKinsey & Company; March, 2019
- [9] Chuprina, Roman: AI and Machine Learning in Manufacturing: The Complete Guide. January, 2020; <https://spd.group/machine-learning/ai-and-ml-in-manufacturing-industry/> (7.10.2021.)
- [10] Maksymenko, Serhii: AI in Manufacturing: Use Cases and Trends in 2021. January, 2021; <https://mobidev.biz/blog/ai-machine-learning-in-manufacturing> (7.10.2021.)
- [11] Machine Learning Tutorial, <https://data-flair.training/blogs/machine-learning-tutorial/> (10.10.2021.)
- [12] Understanding Machine Learning and Deep Learning, <https://dlflabs.medium.com/understanding-machine-learning-deep-learning-f5aa95264d61> (10.10.2021.)

- [13] Action Detection Using Deep Neural Networks: Problems and Solutions, <https://dzone.com/articles/action-detection-using-deep-neural-networks-proble> (10.10.2021.)
- [14] Moeslund, T.B.; Hilton, A.; Krueger, V.: A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2), 90-126, February, 2006
- [15] Rajendran, Gopika; Lee, Ojus; Gopi, Arya; Jose, Jais; Gautham, Neha: / Study on Machine Learning and Deep Learning Methods for Human Action Recognition/. 2020 Creative Commons CC BY; May, 2020
- [16] Poppe, R.: A survey on vision-based human action recognition . *Image and Vision Computing*, June 2010
- [17] Wu, Di; Sharma, Nabin; Blumenstein, Michael: Recent Advances in Video-Based Human Action Recognition using Deep Learning: A Review. 2017 School of Software, University of Technology Sydney New South Wales, Australia; 2017
- [18] Hartman, B.: Human Worker Activity Recognition in Industrial Enviroments. Karlsruhe Institut fur Technologie, 2011
- [19] Đapić, I.: Prepoznavanje ljudskih aktivnosti. Sveučilište u Splitu, Fakultet Elektrotehnike, Strojarsstva i Brodogradnje, 2015.
- [20] Action Recognition in Video Analysis, <https://www.optima.ai/action-recognition-in-video-analysis/> (13.10.2021.)
- [21] Gudlin, M.: Novi model prepoznavanja ljudskih aktivnosti u proizvodnim procesima primjenom računalnog vida. Sveučilište u Zagrebu, Fakultet Strojarsstva i Brodogradnje, Zagreb 2021.
- [22] What is Lean Managment? Definition and Benefits, <https://kanbanize.com/lean-management/what-is-lean-management> (20.10.2021.)
- [23] Understanding and Coding a ResNet in Keras, <https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33> (21.10.2021.)
- [24] How to Create to a TFRecord File for Computer Vision and Object Detection, <https://blog.roboflow.com/create-tfrecord/> (22.10.2021.)
- [25] Feature Extraction, <https://deeptai.org/machine-learning-glossary-and-terms/feature-extraction> (22.10.2021.)

-
- [26] Difference Between a Batch and an Epoch in a Neural Network, <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/> (4.11.2021.)
- [27] Geissbauer R.; Schrauf S.; Berttram P.; Cheraghi F.: Digital Factories 2020: Shaping the future of manufacturing. PricewaterhouseCoopers GmbH Wirtschaftsprüfungsgesellschaft, 2017.
- [28] 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.
- [29] An overview of gradient descent optimization algorithms, <https://runder.io/optimizing-gradient-descent/> (30.11.2021.)
- [30] Keras Learning Rate Finder, <https://www.pyimagesearch.com/2019/08/05/keras-learning-rate-finder/> (30.11.2021.)

PRILOZI

- I. CD-R disc
- II. Tehnička dokumentacija