

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Karlo Puzak

Zagreb, 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentori:

Prof. dr. sc. Nedeljko Štefanić, mag.ing.

Student:

Karlo Puzak

Zagreb, 2020.

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

Zahvaljujem mentoru prof.dr.sc. Nedeljku Štefaniću na podršci i savjetima tijekom izrade rada i dostupnosti za sva moja pitanja te asistentu Mihaelu Gudlinu, mag.ing.mech na pruženim savjetima tijekom praktičnog dijela rada.

Zahvaljujem se i svojoj obitelji na bezuvjetnoj podršci tijekom studija i pisanja ovog rada kao i svojim prijateljima i kolegama koji su mi bili oslonac u svim životnim situacijama.

Karlo Puzak



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomске ispite
Povjerenstvo za diplomске 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 / 20 - 6 / 3	
Ur. broj: 15 - 1703 - 20 -	

DIPLOMSKI ZADATAK

Student: **KARLO PUZAK** Mat. br.: **0035189647**

Naslov rada na hrvatskom jeziku: **Povećanje produktivnosti proizvodnog poduzeća primjenom dubokog učenja**

Naslov rada na engleskom jeziku: **Increasing the productivity of a manufacturing enterprise by applying deep learning**

Opis zadatka:

Pametna tvornica, najvažniji proizvod Industrije 4.0, predstavlja povezivanje virtualnog i fizičkog svijeta kroz primjenu novih digitalnih tehnologija poput Umjetne inteligencije, robotizacije, Big data, virtualne i proširene stvarnosti, aditivne proizvodnje, IoT sustava i drugih. Posebna se pažnja u pametnoj tvornici posvećuje stalnom poboljšavanju produktivnosti, pri čemu su se metode Umjetne inteligencije pokazuju kao jedno od mogućih pristupa, posebno u uvjetima malih serija proizvoda i velikog broja varijanti.

U radu je potrebno obraditi sljedeće:

1. Definirati i detaljno objasniti pojmove: Industrija 4.0, Pametno poduzeće, Digitalne tehnologije.
2. Detaljno objasniti pojam Umjetne inteligencije te navesti metode koje koristi.
3. Definirati pojam produktivnosti proizvodnog poduzeća te sistematizirati glavne utjecajne parametre (zaposlenik, proces, kvaliteta, materijal...).
4. Razraditi sustav praćenja produktivnosti u proizvodnom poduzeću koristeći metodu dubokog učenja.
5. Za proizvoljno odabrano proizvodno poduzeće i proizvoljno odabrani proces primijeniti razvijeni sustav praćenja produktivnosti te komentirati dobivene rezultate.
6. Razviti opći pristup primjene Umjetne inteligencije u proizvodnom poduzeću.

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

Zadatak zadan:
24. rujna 2020.

Rok predaje rada:
26. studenog 2020.

Predviđeni datum obrane:
30. studenog do 4. prosinca 2020.

Zadatak zadao:

prof. dr. sc. Nedeljko Štefanić

Predsjednica Povjerenstva:

prof. dr. sc. Biserka Runje

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	III
POPIS KRATICA	V
SAŽETAK	VI
SUMMARY	VII
1 UVOD	1
2 INDUSTRIJA 4.0	2
2.1 DIGITALNE TEHNOLOGIJE	3
2.1.1 Računarstvo u oblaku	4
2.1.2 Internet stvari	5
2.1.3 Proširena i virtualna stvarnost.....	5
2.1.4 Big Data	6
2.2 NOVE TEHNOLOGIJE I OPERACIJSKA ISTRAŽIVANJA	6
2.2.1 Heurističke metode.....	8
3 GENETSKI ALGORITMI	9
4 UMJETNA INTELIGENCIJA	10
4.1 POVIJEST I RAZVOJ UMJETNE INTELIGENCIJE	10
4.2 STROJNO UČENJE.....	13
4.2.1 Metode strojnog učenja	14
4.3 DUBOKO UČENJE	18
4.4 UMJETNA INTELIGENCIJA U HRVATSKOJ I SVIJETU	19
4.4.1 Mitsubishi Motors.....	22
4.4.2 Sandvik	23
4.4.3 Fujitsu	24
5 PRODUKTIVNOST PROIZVODNOG PODUZEĆA	25
5.1 PRODUKTIVNOST VS UČINKOVITOST	25
5.2 GRANICA MOGUĆNOSTI PROIZVODNJE	26
5.3 ZAŠTO SE MIJENJA RAZINA PRODUKTIVNOSTI?	27
5.4 POVEĆANJE PRODUKTIVNOSTI	28
5.4.1 Vitka proizvodnja (Lean) i smanjenje otpada	28
5.4.2 Strategije smanjenja otpada	29
6 DUBOKO UČENJE	31
6.1 AKTIVACIJSKE FUNKCIJE	32
6.2 HIPERPARAMETRI	34
6.2.1 Konvolucijska neuronska mreža	35
6.2.2 Autoenkoder	36
6.2.3 Ponavljajuća neuronska mreža.....	36
6.3 FUNKCIJA GUBITKA	38
6.3.1 MSE.....	38
6.3.2 Negativna log-izglednost.....	38
7 PRIMJENA DUBOKOG UČENJA U PROIZVODNJI	39
7.1 DESKRIPTIVNA ANALIZA	39
7.2 DIJAGNOSTIČKA ANALIZA	40
7.3 PREDIKTIVNA ANALIZA	40

8	PRIKAZ IMPLEMENTACIJE MODELA NEURONSKIH MREŽA	41
8.1	TRADICIONALNI PRISTUP RJEŠAVANJU PROBLEMA KLASIFIKACIJE	41
8.1.1	<i>Implementacija i analiza rezultata</i>	<i>42</i>
8.1.2	<i>Razlike između algoritama</i>	<i>47</i>
8.2	DEEP LEARNING PRISTUP U DETEKCIJI SLIKE	48
8.2.1	<i>Prikaz algoritma na realnim podacima</i>	<i>55</i>
9	ZAKLJUČAK	61
10	LITERATURA	62
11	PRILOZI.....	65

POPIS SLIKA

Slika 1.	Industrija 4.0.	3
Slika 2.	Razlike u proizvodnim linijama	4
Slika 3.	Računarstvo u oblaku	5
Slika 4.	Vizualizacija optimiranja problema	6
Slika 5.	Grafički prikaz problema NP-a	7
Slika 6.	Ilustracija neuronske mreže	12
Slika 7.	Veza između UI, strojnog i dubokog učenja	13
Slika 8.	Razlika između tradicionalnog pristupa i strojnog učenja	14
Slika 9.	Ovisnost prihoda i štednje u procjeni kreditne sposobnosti	16
Slika 10.	Odnos cijene i prijeđenih kilometara automobila	17
Slika 11.	Princip rada podržanog učenja	18
Slika 12.	Prikaz rada algoritma dubokog učenja	19
Slika 13.	Razina industrijske zrelosti u RH po segmentima	19
Slika 14.	Razina industrijske zrelosti u RH	20
Slika 15.	Robot RONNA u operacijskoj sali	21
Slika 16.	Autonomni robot Gideon Brothersa	22
Slika 17.	Mitsubishijeva mobilna platforma	23
Slika 18.	Aplikacija Prism	24
Slika 19.	Granica mogućnosti proizvodnje	27
Slika 20.	Jednostavni prikaz neuronske mreže	31
Slika 21.	Linearna funkcija	32
Slika 22.	Step funkcija	33
Slika 23.	Sigmoidna funkcija	34
Slika 24.	ReLU funkcija	34
Slika 25.	Prikaz konvolucijske mreže	36
Slika 26.	Prikaz autoenkodera	36
Slika 27.	Prikaz ponavljajuće neuronske mreže	37
Slika 28.	Razlike između neuronskih mreža	37
Slika 29.	Tablični podaci za obradu	42
Slika 30.	Učitavanje podataka u Python	43
Slika 31.	Informacije o podacima prije obrade	43
Slika 32.	Podaci poslije obrade	43
Slika 33.	Pretraživanje praznih vrijednosti	44
Slika 34.	Suma ključnih podataka za obradu	44
Slika 35.	Prosječna vremena rada	45
Slika 36.	Optimizacija podataka	46
Slika 37.	Učitavanje modela	46
Slika 38.	Grafički prikaz rada modela	46
Slika 39.	Učitavanje modela slučajnih šuma	47
Slika 40.	Prikaz rada modela RTF	47
Slika 41.	Učitavanje modela k-NN	48
Slika 42.	Usporedba točnosti modela	48

Slika 43.	Određivanje značajki	49
Slika 44.	Integralne slike	50
Slika 45.	Naredba za pokretanje kamere	51
Slika 46.	Definiranje kvadrata lokalizacije lica	51
Slika 47.	Detekcija lica	52
Slika 48.	Nema detekcije lica	53
Slika 49.	Lažno pozitivna slika	53
Slika 50.	Detekcija lica na isprintanoj slici.....	54
Slika 51.	Prikupljanje podataka u postrojenju	55
Slika 52.	Prikaz mogućih oštećenja površine	57
Slika 53.	Učitavanje podataka.....	57
Slika 54.	Struktura podataka.....	58
Slika 55.	Nazivlje podataka	58
Slika 56.	Definiranje modela klasifikatora	58
Slika 57.	Grafički prikaz MSE.....	59
Slika 58.	Numerički prikaz točnosti.....	59
Slika 59.	Prva nasumično odabrana slika za klasifikaciju	60
Slika 60.	Druga nasumično odabrana slika za klasifikaciju.....	60

POPIS KRATICA

Kratika	Opis
AE	<i>Artificial Intelligence</i> – umjetna inteligencija
AI	<i>Artificial Neural Network</i> – umjetna neuronska mreža
ANN	<i>Augmented Reality</i> – proširena stvarnost
AR	<i>Auto Encoder</i> – autoenkoder
CNC	<i>Comma Separated Value</i> – vrijednosti odvojene zarezom
CNN	<i>Computer Numerical Control</i> – računalno numeričko upravljanje
CRM	<i>Convolutional Neural Network</i> – konvolucijska neuronska mreža
CSV	<i>Customer Relationship Management</i> – sustav upravljanja kupcima
DL	<i>Decision Tree</i> – stablo odlučivanja
DNK	<i>Deep Learning</i> – duboko učenje
DT	Deoksiribonukleinska kiselina
FER	Fakultet elektrotehnike i računarstva
INSENT	<i>Innovative Smart Enterprise</i> – inovativno pametno poduzeće
IoT	<i>Internet of Things</i> – internet stvari
k-NN	<i>k-Nearest neighbor</i> – k najbliži susjedi
MIT	<i>Machine Learning</i> – strojno učenje
ML	<i>Massachusetts Institute of Technology</i>
MSE	<i>Mean Square Error</i> – srednja kvadratna pogreška
NERO	<i>Negative Log Likelihood</i> – vjerojatnost negativnog logaritma
NLL	<i>Neurosurgical Robot</i> – neurokirurški robot
RNN	<i>Random Tree Forest</i> – algoritam slučajne šume
RONNA	<i>Recursive Neural Network</i> – ponavljajuća neuronska mreža
RTF	<i>Robotic Neurosurgical Navigation</i> – robotska navigacija za neurokirurgiju
VR	<i>Virtual Reality</i> – virtualna stvarnost

SAŽETAK

Klasični proizvodni pogoni suočavaju se s izazovima u pronalaženju adekvatne radne snage, opstanka na tržištu i povećanjem profitabilnosti. Razvojem novih tehnologija stvaraju se nove multidisciplinarnе struke koje omogućavaju spajanje nekoliko razdvojenih struka poput primjerice psihologije i računarstva za analizu i predikciju zapošljavanja, pa se tako i kreiraju nova radna mjesta koja zahtijevaju drugačije kompetencije i znanja koja nisu zaobišla ni strojarstvo. U ovom radu stavit će se fokus na primjenu umjetne inteligencije u strojarstvu. Umjetna inteligencija razvijala se od 50-ih godina prošlog stoljeća, a razvija se i danas sve više. U strojarstvu se najviše koristi u obliku prediktivnih modela za procjenu nastanka kvarova, optimizacije ruta vozila ili vizijsko komisioniranje pomoću proširene stvarnosti.

Ključne riječi:

Umjetna inteligencija, strojno učenje, duboko učenje, industrija 4.0, pametna tvornica, IoT

SUMMARY

Traditional manufacturing is facing hardships with regards to staying competitive on the market, employing the best people, and increasing the revenue. With the improvements the technology, new multidisciplinary professions are emerging. In every field of work, people are required to have new knowledge and the same is for the manufacturing industry. This thesis focuses on explaining the effects of machine learning and artificial intelligence on mechanical and manufacturing processes. Mechanical engineering is mostly using predictive analytics to detect defects in equipment, vehicle route optimization or visionary commissioning in combination with augmented reality.

Key words:

Artificial intelligence, machine learning, deep learning, industry 4.0, smart factory, IoT

1 Uvod

Svijet se mijenja. Proizvodnja se mijenja. No mijenja li se dovoljno brzo i učinkovito da odgovori na sve potrebe tržišta? Prva industrijska revolucija donijela je otkriće i upotrebu parnog pogona, druga nam je donijela masovnu industrijalizaciju, a treća automatizaciju. Sada se nalazimo u četvrtoj industrijskoj revoluciji koja primarno donosi pametnu automatizaciju i upravljanje podacima. Nove generacije inženjera odrastaju s novim tehnologijama i naučeni su ih koristiti od djetinjstva. U mnogim slučajevima tvornice nisu digitalizirale poslovanje i manje su povezane u odnosu na nove tvrtke koje se stvaraju. Te nove tvrtke koriste tehnologije budućnosti i nazivamo ih pametnim tvornicama jer imaju potencijal za revoluciju u globalnim proizvodnim operacijama. Inženjeri bez iskustva s novim tehnologijama imaju znanje i iskustvo koje novi, mlađi kolege nemaju. Kombinacijom te dvije ekspertize stvaraju se stručnjaci koji su sposobni odgovoriti na izazove u poslovanju. Korištenje simulacija, prototipa i omogućavanje testiranja otvara nove mogućnosti za donošenje informiranih odluka. Tvrtke koje integriraju umjetnu inteligenciju moći će klasificirati i organizirati svoje podatke za praćenje toka materijala ili praćenje transakcija. Omogućeno je i predviđanje ponašanja korisnika, prodaje i optimizacija procesa.

2 Industrija 4.0

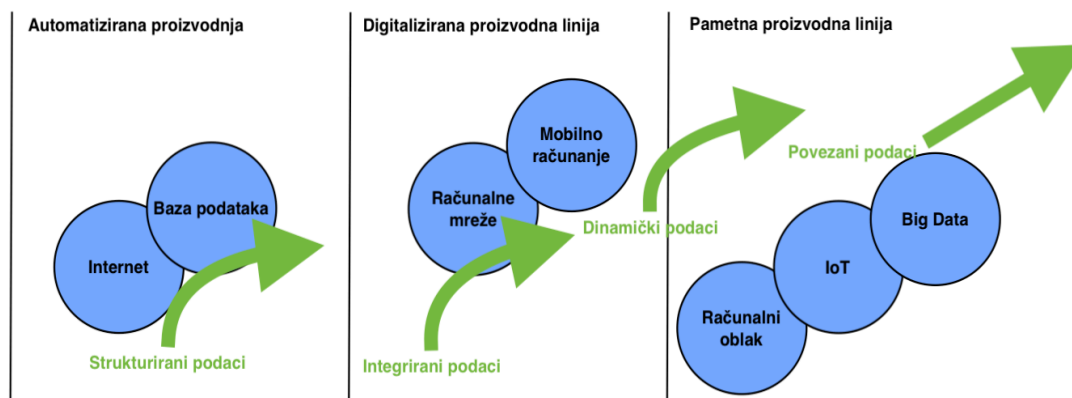
Početak 19. stoljeća došlo je do prve industrijske revolucije koja je za posljedicu imala proizvodnju transportnih sredstava s pogonom na vodu i vodenu paru – parobrod i parna lokomotiva. Druga industrijska revolucija uslijedila je početkom 20. stoljeća i pojavom električne energije što je omogućilo Henry-ju Fordu da pokrene proizvodnju automobila pomoću proizvodne trake. Početkom 1970. započela je treća industrijska revolucija koja je doživjela procvat u proizvodnji elektronike i robotike što je omogućilo automatizaciju operacija u proizvodnom procesu. U zadnjih nekoliko godina osjeti se i integracija informacijsko komunikacijskih tehnologija u proizvodne procese čime je osiguran razvoj novih poslovnih modela i tržišta. Integracija interneta stvari (eng. Internet-of-Things, IoT), računarstva u oblaku (eng. Cloud computing), aditivnih tehnologija, računalnog vida, velikih baza podataka itd. su srž nove industrijske revolucije – Industrije 4.0. Najveća snaga i prednosti Industrije 4.0 je upravo umreženost svih strojeva i procesa s čovjekom koji može donijeti strateški bitne odluke samo ako ima sve raspoložive informacije. Računala služe kao potpora u odlučivanju čime se omogućava optimizacija proizvodnih linija, skladišta, odnosa s kupcima itd. Razni senzori u opremi omogućavaju pravovremeno preventivno održavanje, a aditivne tehnologije kao sastavni dio nove industrijske revolucije stvaraju manje troškove za zamjenu dijelove i omogućuju izradu zamjenskih dijelova. Slika 1. prikazuje različite koncepte koji čine sastavke dijelove Industrije 4.0 [1].



Slika 1. Industrija 4.0.

2.1 Digitalne tehnologije

Sustavi i programi koji se koriste godinama, i koji su isprobani, su jedan od glavnih razloga zašto poduzeća teško prihvaćaju promjene. Ono što svako poduzeće treba uvidjeti je da su takvi sustavi često izvor troška radi održavanja, nedovoljnih funkcionalnosti što uzrokuje korištenje dodatnih alata i da nisu skalabilni u pogledu omogućavanja širenja poduzeća. Potrebno je napraviti *cost-benefit* studiju gdje se na temelju informacija može donijeti odluka o zadržavanju sustava, potpunom uklanjanju ili modernizaciji. Faktori koji utječu na konačnu odluku mogu biti smanjenje troškova, povećanje efikasnosti ili usklađivanje s novim regulativama i zakonima. Prikupljanje raznih podataka znači imati i sustav za njihovo skladištenje što se nerijetko odvija po raznim bazama koje nisu dio jedne cjeline i gdje je teško raditi analitike. Objedinjavanje podataka je prvi korak prema informiranijem donošenju odluka ključnih za razvoj poduzeća, a to je omogućeno korištenjem digitalnih tehnologija. Na slici Slika 2. prikazane su razlike između 3 razine automatizacije proizvodnog poduzeća [2].



Slika 2. Razlike u proizvodnim linijama

2.1.1 Računarstvo u oblaku

Računarstvo u oblaku (eng. Cloud Computing) omogućuje infrastrukturu za korištenje aplikacija ili prostora za pohranu podataka putem podatkovne mrežne veze. Oblaci mogu dostupni široj populaciji – javni ili mogu biti limitirani za korištenje u samo jednom poduzeću – privatni.

Značajke računarstva u oblaku su [3]:

- dostupnost podataka – podacima se može pristupiti bilo kada i bilo gdje ako je osigurana internet veza
- olakšano održavanje – nema troškova instalacije programa na svako računalo pa su i troškovi održavanja niži
- povećanje produktivnosti – s obzirom na to da su podaci spremljeni na jedno mjesto, moguće je da više korisnika koristi iste podatke odjednom za analizu
- agilnost i skalabilnost – ovisno o pružatelju usluga moguće je imati gotovo pa neograničeni prostor za pohranu podataka što podržava rast i razvoj poduzeća.

Na slici Slika 3 prikazana je pojednostavljena shema računarstva u oblaku gdje su umreženi računala, mobiteli, laptopi, baze podataka i ljudi preko jednog oblaka u kojem se nalaze sve potrebne informacije [4].



Slika 3. Računarstvo u oblaku

2.1.2 Internet stvari

Internet stvari (eng. Internet of Things - IoT) podrazumijeva sve proizvode i sustave koji su međusobno umreženi i mogu komunicirati putem interneta. U proizvodnim poduzećima korištenje IoT omogućuje upravljanje proizvodnom opremom i proizvodnim procesom s udaljenosti uz osiguranu internetsku vezu. IoT povezan kroz senzore u proizvodnji pomaže u prikupljanju podataka bitnih za prediktivno održavanje, statističku obradu podataka i smanjenje troškova potrošnje električne energije.

2.1.3 Proširena i virtualna stvarnost

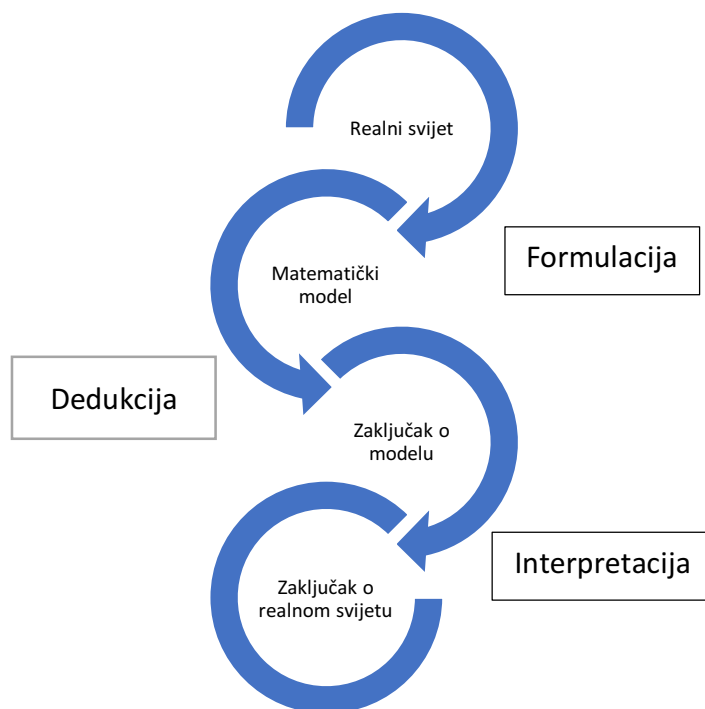
Proširena (eng. Augmented Reality - AR) i virtualna stvarnost (eng. Virtual Reality - VR) su različiti oblici dobivanja dodatnih informacija iz okoline korištenjem pametnih naočala ili kamere mobitela. Slika koja je prikazana na ekranu mobitela je obogaćena dodatnim objektima i informacijama dobivenim kroz računalo. Tako se primjerice AR koristi u skladištima za dobivanje informacija o ruti kako dohvatiti određeni proizvod dok se VR često koristi za vježbanje i testiranje znanja o procesu zavarivanja ili tijekom obuke pilota zrakoplova. VR u potpunosti zamjenjuje stvarni svijet s novim, umjetno stvorenim svijetom [5].

2.1.4 Big Data

Velike količine podataka (eng. Big Data) predstavljaju sve podatke nekog poduzeća koji se spremaju, a nisu nužno obrađeni, odnosno to su sirovi podaci (eng. Raw data) koje je potrebno obraditi korištenjem statističkih metoda. Količina podataka je još izraženija uvođenjem senzora i digitalizacije poduzeća. Zato je potrebno imati sustave koji mogu spremati, ali i obraditi velike količine podataka. Ako se ispravno koristi u suradnji s ostalim tehnologijama može pridonijeti u smanjenju troškova proizvodnje, povećanju proizvodnosti, utvrđivanju i predviđanju kvarova u strojevima i cjelokupnom stanju poduzeća [5].

2.2 Nove tehnologije i operacijska istraživanja

Operacijska istraživanja su stručno-znanstvena disciplina koja omogućava donošenje optimalnih odluka korištenjem razvijenih matematičkih modela i metoda. Operacijska istraživanja imaju korijene u 2. svjetskom ratu, a proširila su se u poslovne i proizvodne sustave. U realnom svijetu imamo jako puno varijabli koje ne moraju biti linearno zavisne i u operacijskim istraživanjima razlikujemo linearno i nelinearno programiranje, a oba se mogu riješiti analitički, grafički ili numerički. U nelinearnom programiranju ne postoji jedna univerzalna metoda rješavanja zadataka, već se prilagođavaju postojeće [6].

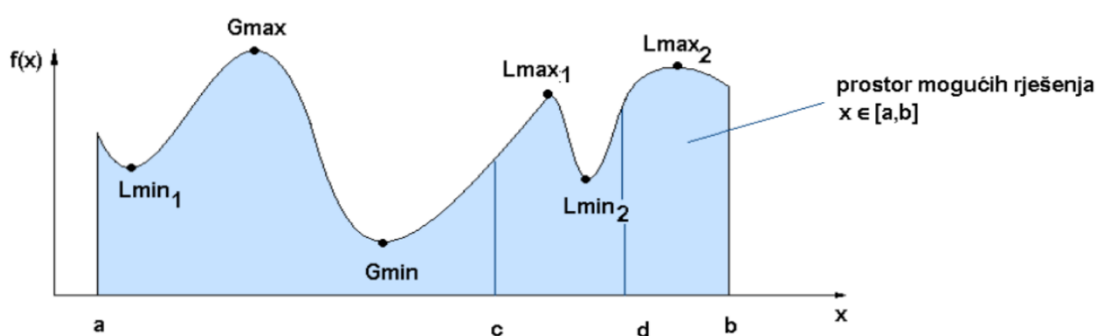


Slika 4. Vizualizacija optimiranja problema

Metode nelinearnog programiranja možemo podijeliti prema načinu rada i prema funkciji na one koje koriste derivacije (gradijentne) i one koje ne koriste derivacije (pretraživanje). Klasični model ima dvije komponente: funkciju cilja $F(x)$ i ograničenja $g(x)$. Rješenje problema je pronaći takve vrijednosti varijable x u kojima funkcija cilja poprima ekstremnu vrijednost – minimum ili maksimum uz zadovoljavanje uvjeta ograničenja što prikazuje Slika 5 [7].

$$F(x) = F(x_1, x_2, \dots, x_n)$$

$$g_i(x) = g_i(x_1, x_2, \dots, x_j, \dots, x_n) \begin{cases} \leq \\ = \\ \geq \end{cases} 0, \quad i = 1, 2, \dots, n \quad (1)$$



Slika 5. Grafički prikaz problema NP-a

Metode koje pronalaze rješenje u zadanom prostoru mogućih rješenja u kojima imamo jednu varijablu su [7]:

- Newtonova metoda
- Fibonaccijeva metoda
- metoda zlatnog reza
- metoda aproksimacije polinomom
- metoda uniformnog traženja.

Kod funkcije više varijabli možemo se poslužiti nekom od niže nabrojanih metoda [7]:

- metoda eliminacije
- metoda Lagrangeovih multiplikatora
- metoda koja se koristi kaznenim funkcijama
- metoda najbržeg uspona
- Monte Carlo
- Hooke Jeeves metoda.

Neki problemi nisu rješivi algoritmima u realnom vremenu poput određivanja rute vozila, optimizacije rasporeda radnih stanica ili planiranja održavanja alata i strojeva i u takvim slučajevima se koriste heurističke metode optimizacije.

2.2.1 *Heurističke metode*

Heurističke metode traže rješenja koja su blizu optimalnom rješenju u razumnom vremenu. Rješenje je blizu globalnog optimuma u razumno kratkom vremenu uz malu mogućnost pronalaska lošeg rješenja odlikuje kvalitetan heuristički algoritam [7]. Heurističke metode mogu se podijeliti na metaheuristiku i heuristiku za specifične probleme. Metaheuristika se može primijeniti na skoro svaki optimizacijski problem. U metaheuristici potrebno je voditi računa o diversifikaciji prostora pretraživanja kako bismo ispunili uvjet pronalaska suboptimalnog rješenja na globalnoj razini i eksploatacije najboljeg rješenja na lokalnoj razini. Najbolje rješenje u lokalnom području osigurava konvergenciju rješenja prema optimalnom rješenju na globalnoj razini. Obje komponente moraju biti ispunjenje jer ne želimo algoritam koji pretražuje samo lokalno područje, niti premalo eksploatacije jer algoritam onda ne konvergira prema globalnom optimumu. Naravno kod svake pretrage prostora treba uzeti i kriterij zaustavljanja – ponavljajuće ažuriranje trenutno najboljeg rješenja [8]. Algoritme možemo podijeliti na [8]:

- algoritmi inspirirani prirodom
 - evolucijski algoritmi, kolonija mrava, roj pčela
- algoritmi s mogućnošću i bez mogućnosti pamćenja
 - simulirano kaljenje
 - tabu pretraživanje
- algoritmi bazirani na populaciji i bazirani na jednom rješenju
- algoritmi sa stohastičkom i determinističkom funkcijom
- iterativni i pohlepni algoritmi.

3 Genetski algoritmi

Genetski algoritam je jedan od najkorištenijih heurističkih algoritama i pripada evolucijskim algoritmima koji su razvijeni uz pomoć inspiracije iz prirode. Darwinova teorija evolucije nam govori da se priroda stalno mijenja i da opstaju najjača populacija. U određenoj populaciji pokušava se očuvati sve ono korisno što je uzrokovalo preživljavanje, a sva loša svojstva zamijeniti boljima. Populacija se razmnožava kako bi se spasila od izumiranja, a razabiranje u kojem jedinke opstaju zove se selekcija. Jedinke koje prežive nasljeđuju svojstva roditelja, ali mogu se javiti i razne mutacije, pa se svojstva mijenjaju i čine ključ u prilagodbi populacije na nove uvjete u prirodi. Tako i genetski algoritmi sadrže operatore koji oponašaju razmnožavanje, mutaciju i selekciju kako bi se došlo do suboptimalnog rješenja u globalnom optimumu. Varijabla predstavlja fenotip jedinki i njihovim kodiranjem dobivamo kromosome koji predstavljaju genotipove jedinki. Svaki kromosom se sastoji od gena. Funkcija dobrote određuje kvalitetu rješenja svake iteracije odnosno selekcije. Evolucijski algoritam je zasnovan na populaciji rješenja koja se sastoji od mnoštva jedinki koje se križaju i mutiraju [9].

Pseudokod rada algoritma [10]:

```
GENERIRAJ početnu populaciju  $p$ ;  
DOK nije zadovoljen uvjet zaustavljanja {  
    SELEKTIRAJ bolje jedinke;  
    KRIŽAJ bolje jedinke i spremi ih u  $p$ ;  
    MUTIRAJ bolje jedinke iz  $p$ ; }  
ISPISI rješenje
```

 (2)

Pokretanjem algoritma generira se nasumična populacija iz domene koja se pretražuje nakon čega slijedi proces koji se sastoji od operatora SELEKTIRAJ, KRIŽAJ i MUTIRAJ nad jedinkama. Petlja se ponavlja sve do onog trena kada je zadovoljen uvjet dobrote. Tijekom operacije SELEKTIRAJ loše jedinke odumiru, a dobre se križaju i mutiraju. U konačnom rezultatu imamo mnoštvo jedinki koje čine populaciju potencijalnih rješenja, ali jedinka dobivena iz zadnje iteracije predstavlja optimum [10].

Genetski algoritmi se mogu koristiti u mrežnom planiranju kod planiranja aktivnosti projektnih zadataka, u rješavanju problema trgovačkog putnika ili planiranja toka materijala u proizvodno-logističkom procesu.

4 Umjetna inteligencija

Razvojem industrije dolazi do prenošenja fizičke aktivnosti s čovjeka na stroj. Izuzev fizički napornih poslova, potrebno je bilo i osloboditi čovjeka i od misaoni rutinskih poslova koje može obavljati robot (npr. robotska ruka u procesu premještanja proizvoda s jedne proizvodne linije na drugu). Početkom 20. st. rutinski poslovi nisu više jedini koji se pokušavaju zamijeniti već su tu i priprema i analiza podataka tako da to odrađuje algoritam, ali da služi kao potpora u čovjekovom odlučivanju i donošenju odluka. Tu se počela pojavljivati prva potreba za umjetnom inteligencijom. Istraživanje ni tu nije stalo i krenulo se razmatrati mogu li se spojiti umjetna inteligencija s konceptima iz robotike - razvoj autonomnih sustava poput autonomnih vozila u skladištima. Bitna stvar u razvoju umjetne inteligencije imaju etika i pravo gdje je bitno poštovati moralne kodekse, ali i stvarati tehnologiju koja je u skladu sa zakonom (ili ga mijenjati da bude u skladu s razvojem tehnologije). Možda najpoznatiji primjer [11] moralne dileme je izbor između dva slučaja gdje autonomno vozilo mora odabrati tko će umrijeti između dvije ponuđene opcije proveden na MIT (Massachusetts Institute of Technology) sveučilištu koji je uzeo koncepte iz filozofije i etike na problemu jurećeg tramvaja opisanog 1967. od strane engleske filozofkinje Philippe Foot.

4.1 Povijest i razvoj umjetne inteligencije

1956. održana je dvomjesečna radionica na Dartmouth College-u gdje se okupilo 10 znanstvenika koji se bave teorijom automata, neuronskim mrežama i proučavanjem inteligencije. Dartmutska konferencija nije donijela spektakularne rezultate, ali je ustanovila novo područje - umjetnu inteligenciju - područje različito od operacijskih istraživanja ili teorije upravljanja, koja su se prije toga bavila i nastojala odgovoriti na slična pitanja - proučiti kako strojevi rješavaju zadatke koji trenutno rješavaju ljudi, kako strojevi tvore i razumiju pojmove i jezik te kako mogu unaprijediti sami sebe. Nekoliko znanstvenika dalo je svoju definiciju umjetne inteligencije [12]:

- Patrick. H. Winston (MIT): Proučavanje postupaka koji mogućim čine percipiranje, rasuđivanje i reagiranje
- Marvin Minsky (MIT): Znanost o tome kako postići da strojevi izvode zadatke koji bi, kada bi ih radio čovjek, iziskivali inteligenciju
- Elain Rich (University of Texas at Austin): Umjetna inteligencija bavi se izučavanjem kako računalo učiniti sposobnim da obavlja poslove koje u ovom trenutku ljudi obavljaju bolje

- D. W. Patterson (1990.): Umjetna inteligencija grana je računarske znanosti koja se bavi proučavanjem i oblikovanjem računarskih sustava koji pokazuju neki oblik inteligencije. Takvi sustavi mogu učiti, mogu donositi zaključke o svijetu koji ih okružuje, oni razumiju prirodni jezik te mogu spoznati i tumačiti složene vizualne scene te obavljati druge vrste vještina za koje se zahtijeva čovjekov tip inteligencije.

Sada razlikujemo tri osnovna koncepta umjetne inteligencije [12]:

1. tehnička

- sustavi neizrazite logike

Najjednostavnije opisan kao matematičko-logički model koji slijedi zakonitost: ako-onda (AKO sat pokazuje 23h ONDA ugasi televizor). Za svaki ulaz-izlaz potrebno je definirati skup pravila, a ona eksponencijalno rastu. Za jedan izlaz, a više ulaza i ulaznih varijabli imamo jednadžbu:

$$Izlaz = Ulazne\ varijable^{Broj\ ulaza} \quad (3)$$

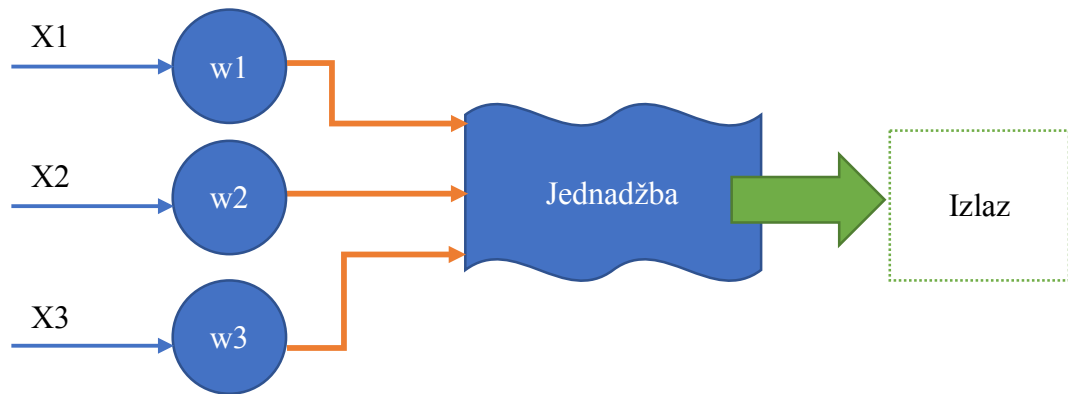
- evolucijski algoritmi

Imitira prirodni proces gdje jači gen pobjeđuje prema Darwinovoj teoriji selekcije. Manipulacijom genetskog stvaraju se nove jedinke, a dobre jedinke se prenose u novu populaciju i izlažu se određenim kriterijima, a najbolja jedinka se ponovno prenosi u novu populaciju sve dok se ne dođe do optimalnog rješenja.

- umjetne neuronske mreže

Sustav sastavljen od mnoštva umjetnih neurona (elemenata), koji su u određenoj interakciji jedni s drugima, ali i s okolinom sustava (težine neuronske mreže). Tako raspoređeni neuroni čine funkcionalnu cjelinu.

Slika 6. prikazuje pojednostavljen model umjetne neuronske mreže.



Slika 6. Ilustracija neuronske mreže

X1, X2, X3 – neuroni umjetne neuronske mreže

w1, w2, w3 – težine

- ekspertni sustavi

Ekspertni sustavi predstavljaju skup računalnih programa koji oponašaju ekspertizu čovjeka. Primjeri se mogu naći u šahovskim i kartaškim igrama gdje se predviđa da će računalo zamijeniti čovjeka s obzirom na mogućnosti predviđanja mogućih kombinacija igranja za svaki potez.

2. biotehnička

- sjedinjavanje biološke i tehničke komponente

Može služiti u rehabilitaciji osoba s invaliditetom ili oštećenjem spajanjem raznih čipova, senzora i aktuatora s čovjekom.

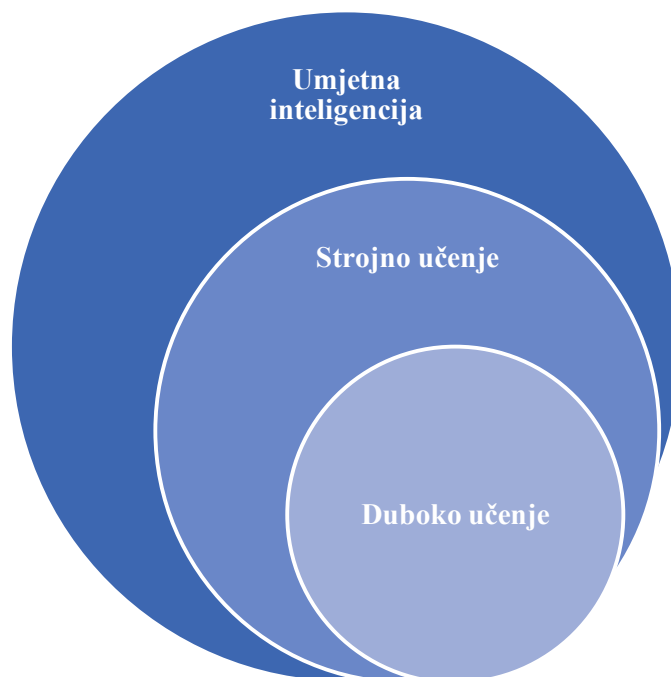
3. biološka

- genetički inženjering

Izrezivanje jednog dijela gena iz DNK i umetanje istog u DNK drugog organizma kako bi se postigli određeni benefiti poput industrijske proizvodnje bjelančevina poput inzulina.

Ekspertni sustavi su se pokazali neuspješnima kod složenijih problema jer je bilo potrebno unijeti veliki skup pravila za svaki ulaz, izlaz i ulazne varijable kako je i objašnjeno u jednadžbi (). Znanstvenici su pokušavali dokučiti postoji li jednostavniji način da računalo uči

i tu se stvorio koncept – strojno učenje. Odnos između raznih koncepata računalne inteligencije dan prikazuje Slika 7 [13].



Slika 7. Veza između UI, strojnog i dubokog učenja

4.2 Strojno učenje

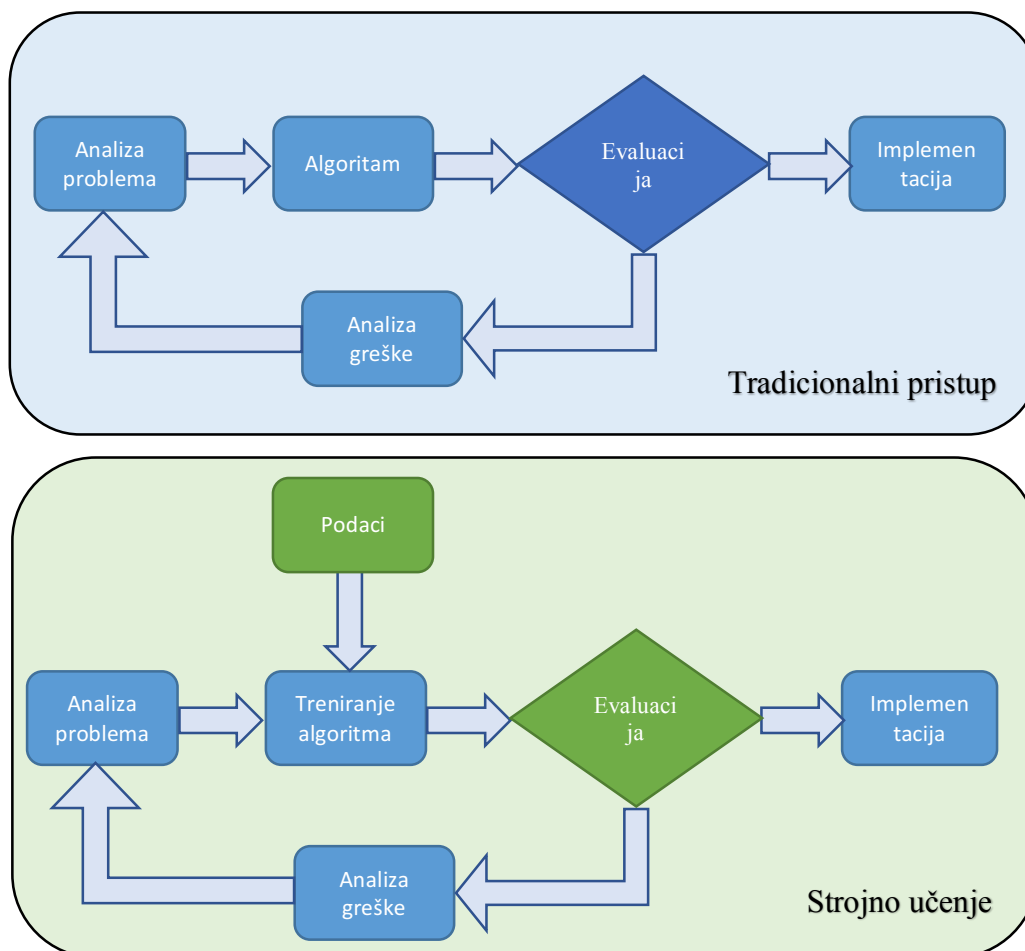
Najopćenitija definicija strojnog učenja je mogućnost unapređenja računalnog sustava kroz iskustvo. Strojno učenje objedinjuje matematiku i statistiku, psihologiju i filozofiju, upravljačku teoriju i računalnu teoriju. Razvoj modela za učenje je zapravo interdisciplinarno područje koje ima za svrhu [14]:

- otkrivanje znanja u skupovima podataka
 - rudarenje podataka u vrhu unapređenja medicinske dijagnostike, analize financijskih izvještaja, kreditni podaci, održavanje opreme, ...
- programske implementacije kompleksnih problema
 - prepoznavanje govora, računalni vid, ...
- dinamički, fleksibilni programi
 - podešavaju se prema korisniku, upravljaju proizvodnim procesima ovisno o uvjetima, ...

Za računalni program kažemo da uči kroz iskustvo E (podaci na kojima program uči), u odnosu na neki skup zadataka T (proučavani problem) i s na mjeru uspješnosti P (funkcija

koju je potrebno optimirati), ako se povećava njegova uspješnost obavljanja zadataka T kroz iskustvo E s definiranom mjerom uspješnosti P.

Slika 8. daje uvid u razliku između tradicionalnog pristupa umjetnoj inteligenciji i strojnog učenja [13].



Slika 8. Razlika između tradicionalnog pristupa i strojnog učenja

Ključni koraci u razradi programa strojnog učenja su odabir prikupljanje i priprema podataka te njihovo razdvajanje na testni skup i skup za treniranje modela, izbor modela i optimizacija parametara, evaluacija modela na testnim podacima i zatim dolazi implementacija modela [15].

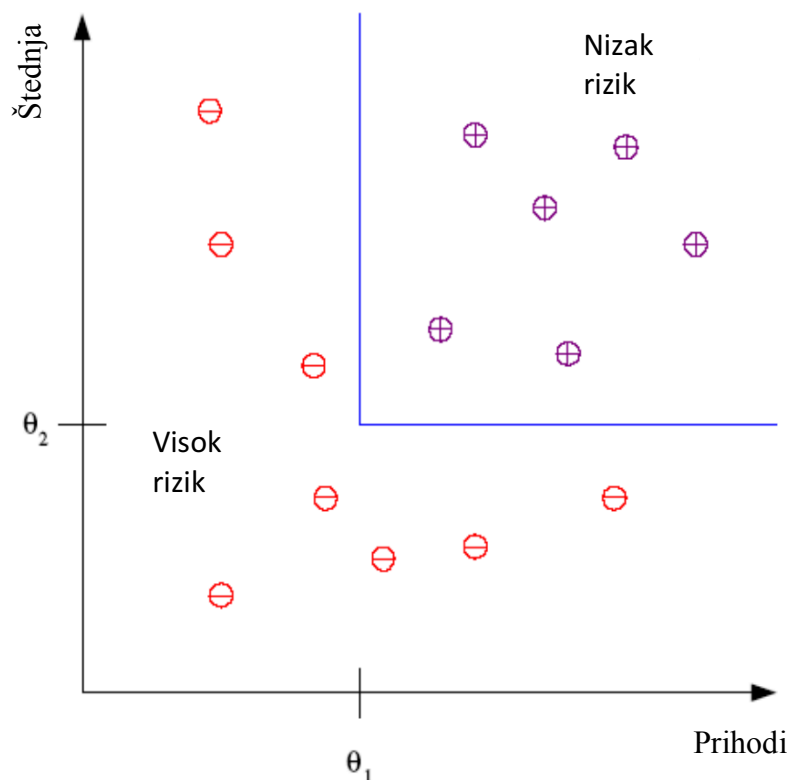
4.2.1 Metode strojnog učenja

Razlikujemo četiri glavne metode u strojnom učenju i njihove algoritme [15]:

1. nadzirano učenje
 - a. regresija

- i. ridge/lasso regresija
 - ii. linearna regresija
 - iii. polinomna regresija.
 - b. klasifikacija
 - i. naivni byes
 - ii. stabla odluke
 - iii. logistička regresija.
- 2. nenadzirano učenje
 - a. klasteriranje/grupiranje
 - i. aglomerativno klasteriranje
 - ii. fuzzy c-sredina
 - iii. k- sredina.
 - b. učenje asocijativnih pravila
 - i. fp rast
 - ii. Euclat
 - iii. Apriori.
 - c. smanjenje dimenzionalnosti
 - i. latentna semantička analiza LSA
 - ii. latentna dirichletova alokacija LDA
 - iii. dekompozicija na singularne vrijednosti SVD.
- 3. podržano učenje
 - a. genetski algoritmi
 - b. Q-učenje
- 4. ansambl
 - a. stacking
 - b. bagging
 - c. boosting.

U nadziranom učenju pojavljuje se mogućnost učenja funkcija (regresija) ili učenje raspoznavanja (klasifikacija). Klasifikacija se može primijeniti u analizi kreditne sposobnosti ili prepoznavanja lica. Grafički prikaz procjene kreditne sposobnosti prikazuje Slika 9 [14].



Slika 9. Ovisnost prihoda i štednje u procjeni kreditne sposobnosti

Algoritam bi radio klasifikaciju na sljedeći način () [14]:

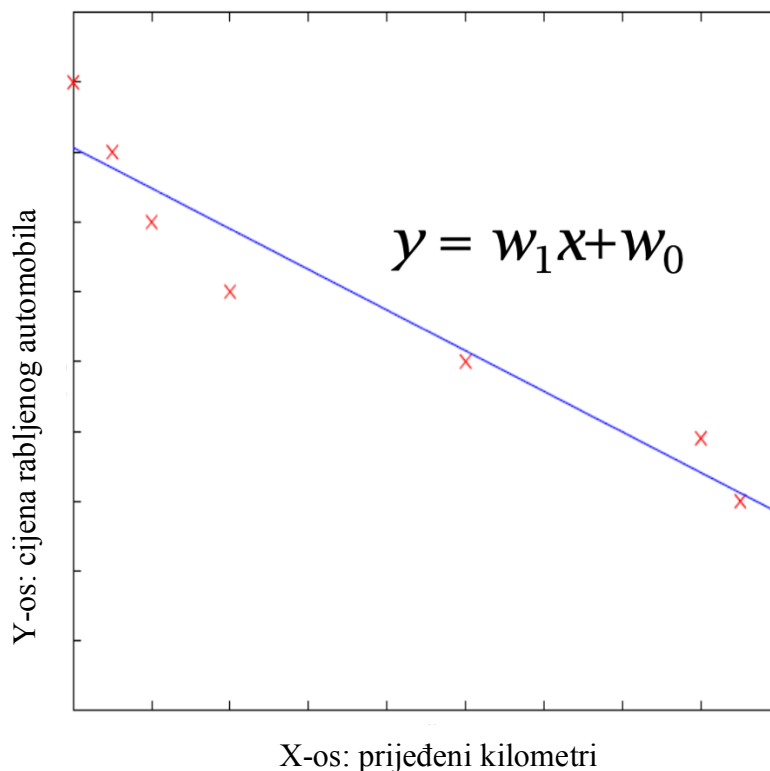
$$\begin{aligned} & \text{IF } \text{prihod} > \theta_1 \text{ AND } \text{štednja} > \theta_2 \\ & \text{THEN } \text{nizak rizik} \text{ ELSE } \text{visok rizik} \end{aligned} \quad (4)$$

Prva linija algoritma je uvjet koji se treba ispuniti da bi imali nizak rizik, a to je da su prihodi i uštedeni iznos veći od granica $\theta_{1,2}$. Ako jedan ili oba uvjeta nisu zadovoljeni onda je osoba visokorizična za odobravanje kredita. Regresija se može objasniti na primjeru odnosa cijene rabljenog automobila i prijeđenih kilometara.

Slika 10. prikazuje kako je cijena veća ako automobil ima manje prijeđenih kilometara, odnosno da cijena pada s povećanjem prijeđenih kilometara. U jednadžbi (5) vidimo da je problem linearan u parametrima i da cijena automobila ovisi o prijeđenim kilometrima [14].

$$\begin{aligned} y &= w_1 x + w_0 \\ y &= g(x | \theta) \end{aligned}$$

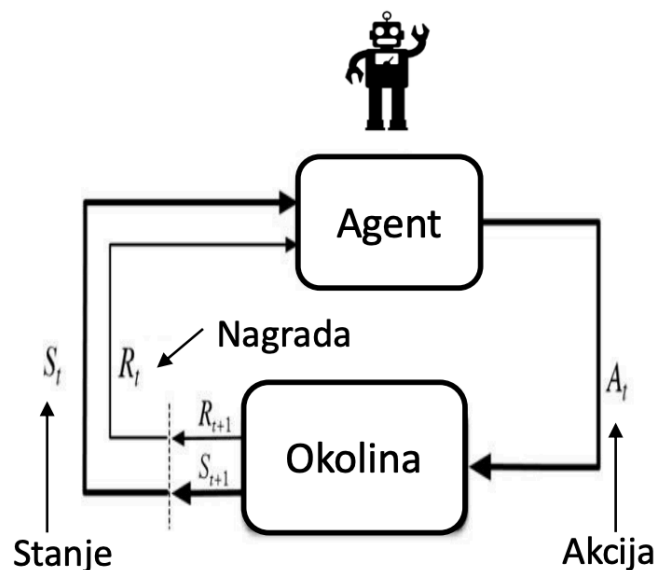
$$\begin{aligned} g & - \text{model, } \theta - \text{parametri } w, \\ x & - \text{atributi automobila, } y - \text{cijena} \end{aligned} \quad (5)$$



Slika 10. Odnos cijene i prijeđenih kilometara automobila

U nenadziranom učenju imamo veliki skup varijabli bez pridružene vrijednosti odzivne varijable. Modeli koji koriste nenadzirano učenje u pravilu traže strukturu ili čak odstupanja u podacima i grupiraju ih (eng. *clustering*). Primjena je u bioinformatiči, kompresiji slike ili segmentaciji korisnika u CRM (eng. *Customer Relationship Management*) sustavima [14].

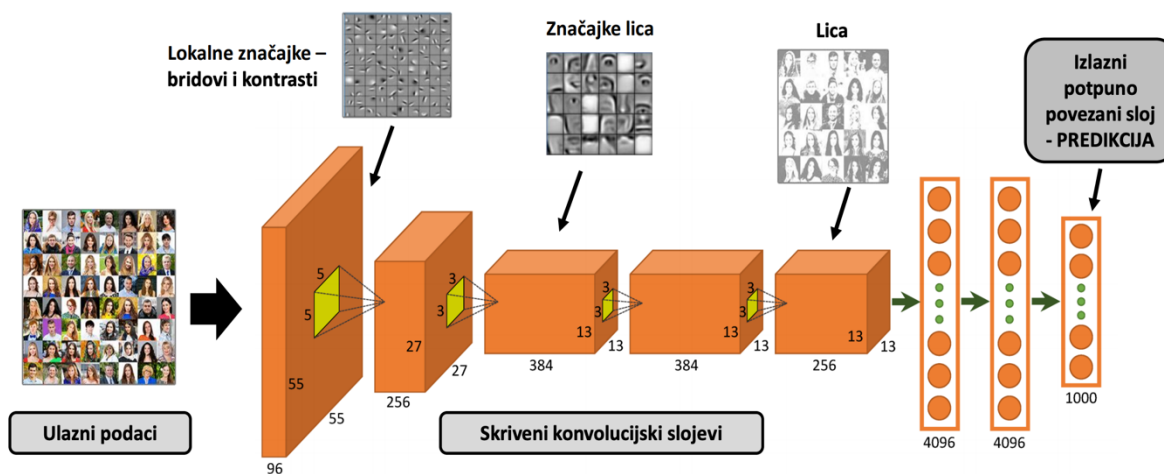
Podržano učenje je metoda za rješavanje sekvencijalnih problema donošenja odluke, u kojoj je simulirani agent u interakciji sa svojom okolinom. Cilj agenta je maksimizirati akumuliranu nagradu kako prikazuje Slika 11 [15].



Slika 11. Princip rada podržanog učenja

4.3 Duboko učenje

Duboko učenje je grana strojnog učenja koja je posebno prikladna za rješavanje problema iz područja umjetne inteligencije. Duboko učenje se temelji na predstavljanju podataka složenim reprezentacijama do kojih se dolazi slijedom naučenih nelinearnih transformacija. Metode dubokog učenja svoju primjenu pronalaze u izazovnim područjima gdje je dimenzionalnost podataka iznimno velika: računalnom vidu, obradi prirodnog jezika ili razumijevanju govora [16]. Računalni vid je sve korišteniji pristup u detekciji i identifikaciji lica gdje raniji slojevi algoritma dubokog učenja raspoznaju rubove i obrise lica, a dublji slojevi prepoznaju varijable / značajke (eng. *features*) lica poput nosa, očiju itd. Algoritmi dubokog učenja rade na principu neuronskih mreža uz dodatne skrivene konvolucijske slojeve neurona što prikazuje Slika 12 [15].



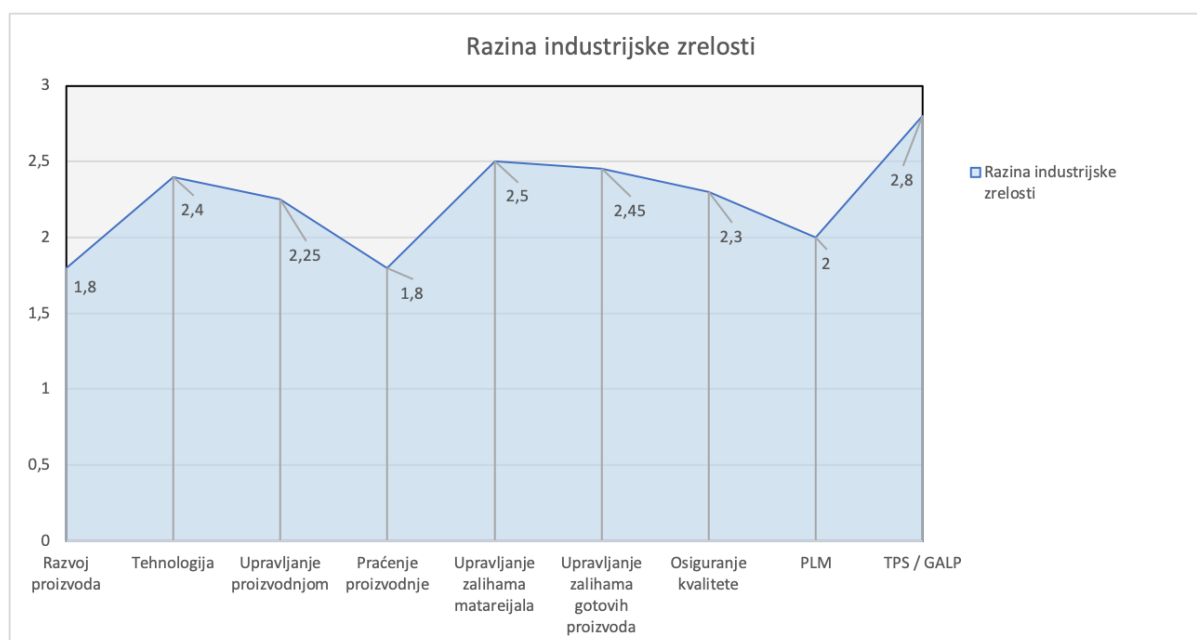
Slika 12. Prikaz rada algoritma dubokog učenja

Umjetni neuron može se prikazati vizualno i matematički. Ulazi u prijenosnu funkciju su definirani kao x_i sa svojim pripadajućim težinama w_i i pragom θ . Izlaz je definiran kao y , a prijenosna funkcija slovom f .

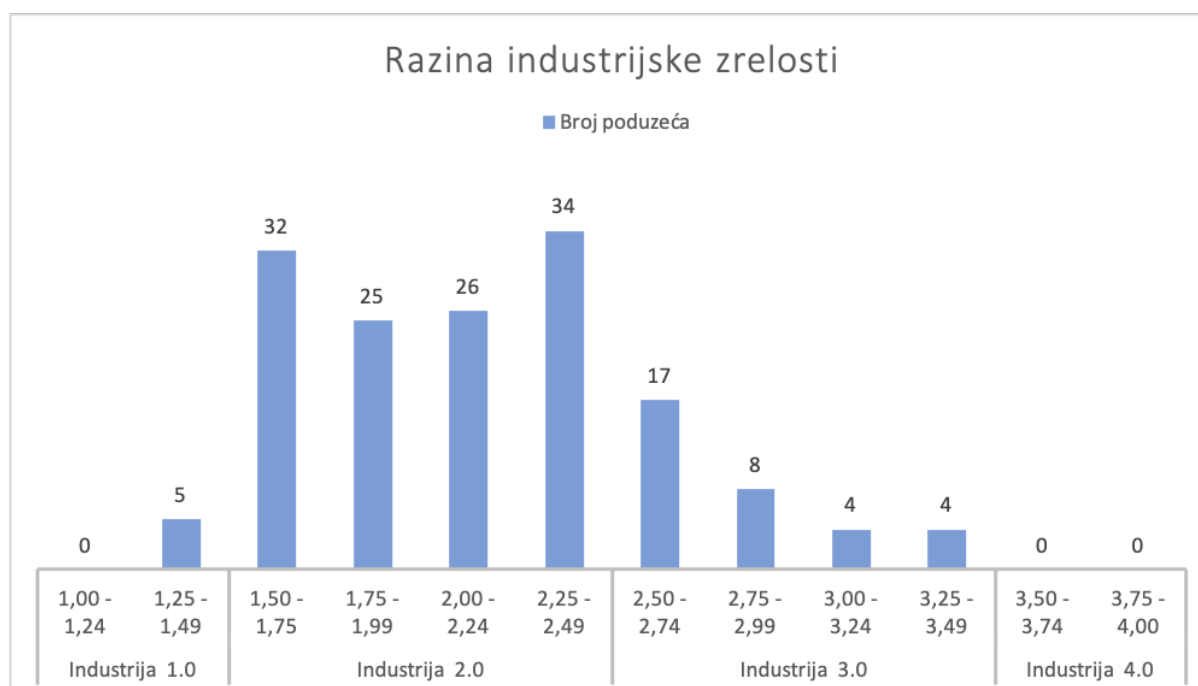
$$y = f\left(\sum_{i=1}^n x_i w_i + \theta\right) \quad (6)$$

4.4 Umjetna inteligencija u Hrvatskoj i svijetu

Istraživanje provedeno 2015. godine u okviru projekta Inovativno Pametno Poduzeće – INSENT (eng. *Innovative Smart Enterprise*) u kojem je sudjelovalo 161 tvrtki iz područja prerađivačke industrije u Republici Hrvatskoj zaključilo je da je prosječna razina automatizacije u RH na razini 2,15, a niti jedna tvrtka nema razinu iznad 3,5. Rezultate ispitivanja po segmentima prikazuje Slika 13., a Slika 14. prikazuje kumulativnu razinu zrelosti tvrtki za Industriju 1.0, Industriju 2.0, Industriju 3.0 i Industriju 4.0. [13].



Slika 13. Razina industrijske zrelosti u RH po segmentima

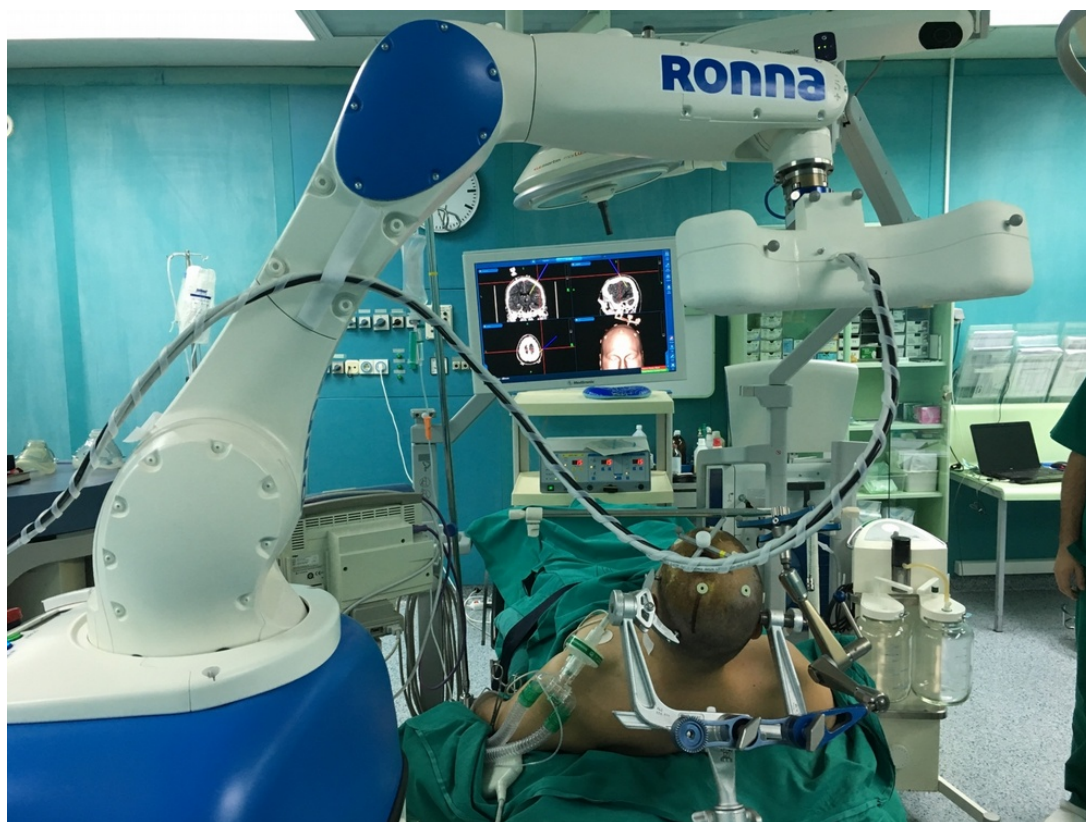


Slika 14. Razina industrijske zrelosti u RH

Niti jedna tvrtka nema razinu zrelosti iznad 3,5 što nam govori da 2015. godine nismo imali tvrtke koje se mogu svrstati u Industriju 4.0. te da samo mali broj tvrtki ima uveden neki oblik automatizacije i korištenja novih tehnologija. Možda naoko lošim rezultatima ne ide u prilog mali skup ispitanih tvrtki, ali i to što se Industrija 4.0 počela intenzivnije razvijati unazad zadnjih nekoliko godina. Hrvatska je tranzicijska zemlja i iako nije na svjetskoj razini u razini proizvodnje, automatizacije i korištenja novih tehnologija u poslovanju, postoji sve veći broj projekata i tvrtki kojima je uporište upravo u korištenju nekog oblika umjetne inteligencije.

Znanstvenici sa zagrebačkog Fakulteta elektrotehnike i računarstva – FER u suradnji sa Sveučilištem u Dubrovniku, Elektrotehničkim fakultetom u Sarajevu i Sveučilištu u Limericku napravili su najveću bespilotnu letjelicu i ronilicu u sklopu NATO programa „Znanost za mir i sigurnost“ u sklopu projekta MORUS. Recentniji projekt FER-ovog Laboratorija za robotiku i inteligentne kontrole sustave je povezan s izradom autonomnog vozila, manipulatora i bespilotne letjelice za poljoprivredne svrhe financiran od strane Hrvatske zaklade za znanost. Fakultet strojarstva i brodogradnje također ne zaostaje za projektima u području umjetne inteligencije. U suradnji s Kliničkim bolničkim centrom Dubrava i Hrvatskog instituta za istraživanje mozga neurokirurg dr. Darko Chudy i prof.dr.sc. Bojan Jerbić razvili su dvoručni robotski neurokirurški sustav RONNA (eng. *Robotic Neurosurgical Navigation*). Iz projekta RONNA izrodio se projekt NERO (eng.

Neurosurgical Robot) u suradnji s Institutom za nuklearnu tehnologiju - INETEC kojem je cilj razviti još jednostavnijeg robota koji će raditi kompleksnije operacije i asistirati kirurzima [17]. Slika 15. prikazuje RONNA sustav u operacijskoj sali [18].



Slika 15. Robot RONNA u operacijskoj sali

Tvrtka Gideon Brothers u suradnji s Hrvatskom Poštom 2019. godine krenula je u pilot-projekt korištenja autonomnih robota u visokoregalnom skladištu i sortirnici HP. Gideonovi roboti će pružati pomoć prilikom prijevoza i komisioniranja robe u skladištu, a u sortirnici će prevoziti pošiljke iz vozila do mjesta za razradu i pripremu za dostavu. Roboti mogu prenositi 800 kg tereta i sigurno prolaziti pored ljudi, opreme i drugih objekata. Opremljeni su tehnologijom autonomije koja se bazira na vizualnoj percepciji, a kombinira dubinsko učenje sa stereoskopskim kamerama čime se omogućava robotski vid [19]. Slika 16. prikazuje jedan takav robot u skladištu.

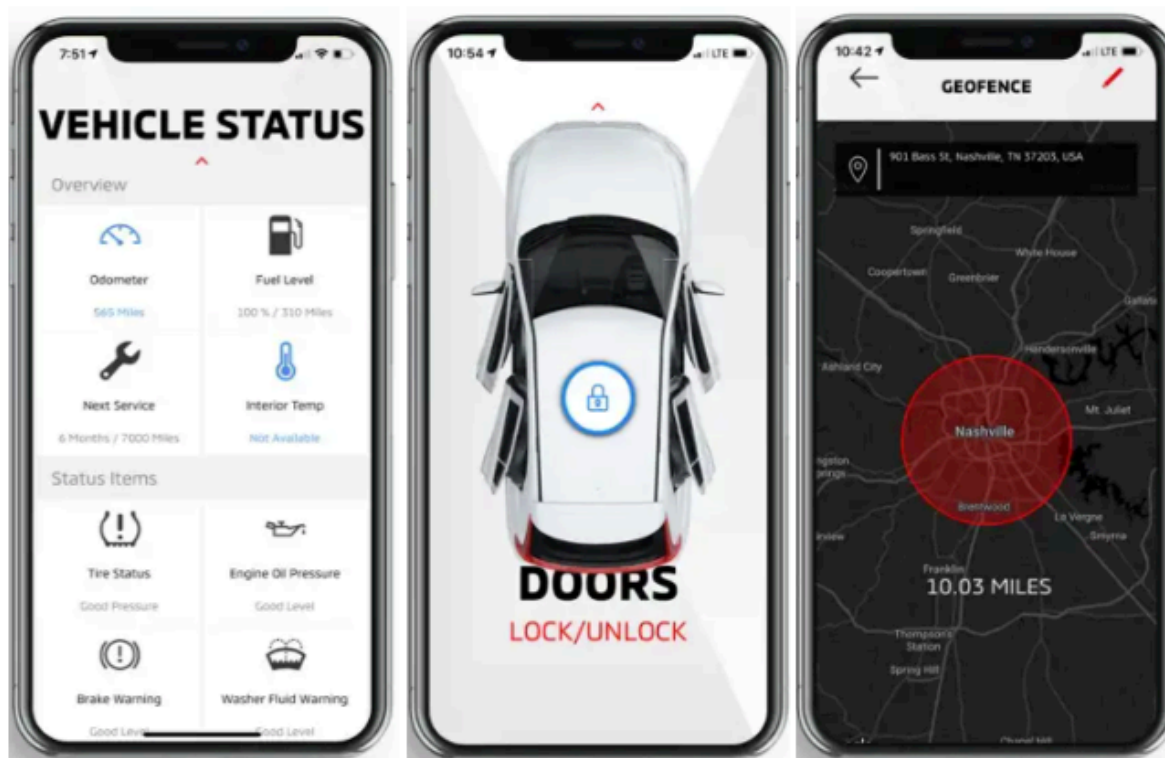


Slika 16. Autonomni robot Gideon Brothersa

4.4.1 Mitsubishi Motors

Pametni automobili su novi trend u proizvodnji automobila. Više no ikada je omogućeno umrežavanje više uređaja putem internetske veze. Putem mobitela se može provjeriti jesu li prozori automobila ostali otvoreni, kada je potrebno vrijeme za servis automobila i zamjenu dijelova i omogućeno je daljinsko pokretanje motora i navigacije uz popratne zabavne sadržaje što vožnju čini personaliziranom i prodajnom točkom za mlađu populaciju. Proizvođači automobila krenuli su u proizvodnju pametnih automobila kako bi odgovorili na zahtjeve tržišta, a to nije zaobišlo ni japansku tvrtku Mitsubishi Motors. Kako je Mitsubishi iskusan u proizvodnji automobila, ali ne i u proizvodnji softvera i pametnih automobila, iskoristili su dostupne resurse i u suradnji s tvrtkom Aeris Communications (Google Cloud partner tvrtka) svu mrežnu komunikaciju prebaciti na Google Cloud. Razlika između izrade novog automobila i proizvodnje softvera za automobil su u linearnostima izrade – automobili su relativno linearni, dok je softver nelinearan, stalno se mijenja i nadograđuje čak i onda kada automobil napusti tvornicu. Remont cijele flote automobila zahtijeva vrijeme i potrebnu ekspertizu pa je Mitsubishi ugradio IoT tehnologiju u svoju Eclipse Cross seriju vozila [20]. Eclipse vozila imaju ugrađene značajke poput glasovne navigacije za upravljanje glazbom i navigacijom. Opremljena su i gumbima za pozivanje medicinske pomoći tijekom vožnje ili asistencije u vožnji [21]. Sva analitika i interakcija korisnika se sprema u bazu podataka kojoj se pristupa preko Google Cloud oblaka, a obrađuje se koristeći interne ML/AI algoritme kako

bi se kreirali novi poslovni modeli i poboljšali softver i hardver vozila. Slika 17. prikazuje neke od funkcionalnosti mobilne platforme [21].



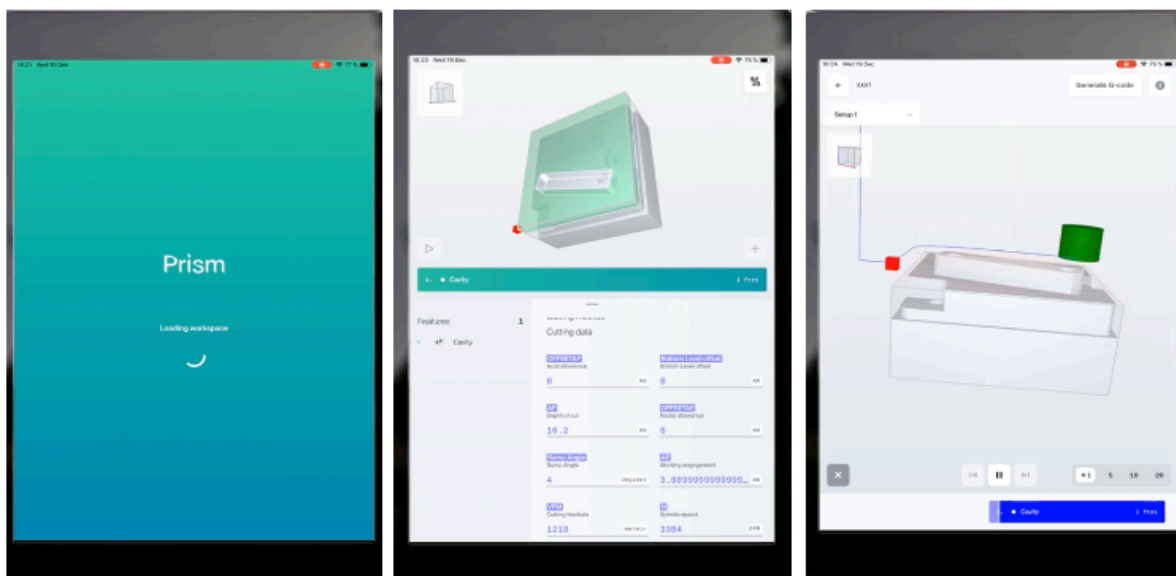
Slika 17. Mitsubishijeva mobilna platforma

Uspostavom računarstva u oblaku, IoT tehnologije i analitike na velikoj količini podataka, Mitsubishi ima povećanje prodaje od 20 %. Osim same prodaje jača se i povjerenje brenda u očima korisnika jer putem svoje nove platforme mogu ostati povezani sa svojim korisnicima čak i nakon završetka kupnje automobila [20].

4.4.2 Sandvik

1860.-ih godina inženjer Göran Fredrik Göransson pionir je Bessemerovog postupka, stvarajući prvi masovno proizvedeni čelik u industrijskim razmjerima. Njegova inovacija tada je transformirala inženjerstvo omogućivši izradu čeličnih proizvoda na brz i jednostavan način diljem svijeta. Tvrtka koju je Göransson osnovao zove se Sandvik i primarno se bavi izradom CNC strojeva za strojnu obradu proizvoda. U svijetu je trenutno preko 2 milijuna CNC strojeva čiji je značaj neprocjenjiv, ali pisanje koda kojim će stroj obrađivati materijal se nije previše promijenio u zadnjih 80 godina. Inženjer mora satima pisati kod, najčešće na stroju, i temeljem iskustva ga razraditi tako da se stroj ne ošteti i da proizvod bude ispravno obrađen. Za vrijeme pisanja koda stroj ne može raditi što je gubitak u produktivnosti jer tvrtke zarađuju

samo onda dok strojevi rade [22]. Zato je Sandvik krenuo u izgradnju aplikacije Prism [23] koja koristi 3D vizualizaciju i pojednostavljeno sučelje za smanjenje vremena pisanja koda. Oni su također u suradnji s Google Cloud partnerom (Avalon Solutions) integrirali svoju aplikaciju na Google Cloud platformu. Pomoću Cloud DataStore i Cloud Functions moguće je brzo mijenjati i redizajnirati pozadinske i podatkovne modele što je omogućilo da budu fleksibilniji i agilniji. Prema prvim testovima pisanje CNC koda smanjeno je za 94 %, a stopa pogreške smanjila se za gotovo identičnu brojku. Povećanje proizvodnosti je poraslo s oko ~3 % na ~15 % u odnosu na industrijski standard do sad [22]. Slika 18. prikazuje mogućnosti aplikacije Prism [23].



Slika 18. Aplikacija Prism

4.4.3 Fujitsu

Tvrtka Fujitsu uvela je umjetnu inteligenciju u svoju proizvodnu liniju. Sustav za prepoznavanje slika ima dvostruku svrhu – osiguravanje proizvodnje dijela optimalne kvalitete i nadgledanje postupka montaže. Objedinjenjem dva procesa u jedan sustav kontrole kvalitete povećalo je učinkovitost tvrtke. Fujitsu se odlučio za taj korak jer trenutno stanje industrije ne napreduje u skladu sa zahtjevima korisnika gdje je nedostajala fleksibilnost ove tradicionalno vođene, japanske, tvrtke. Uvođenje AI u proizvodnju dobila se potrebna fleksibilnost i nije potrebna velika prilagodba i kalibracija softvera za proizvodnju novih proizvoda. Fujitsu, u gradu Oyama, je smanjio vrijeme razvoja svojih proizvoda za 80 % i krenuo je u integraciju AI u sve svoje tvornice [24].

5 Produktivnost proizvodnog poduzeća

Svako proizvodno poduzeće ima određeni input (ulazna varijabla) i output (rezultat). Inputi su svi resursi – ljudi, sirovine, energija, informacije, novac itd. kojima se osigurava postizanje željenog outputa. Output su svi proizvodi, energija, usluge ili rad proizvedeni na nekom stroju, tvornici, poduzeću ili ljudskim radom.

Produktivnost se onda definira kao omjer:

$$\text{Produktivnost } (P) = \frac{\text{Output } (O)}{\text{Input } (I)} \quad (7)$$

Povećanje produktivnosti ostvaruje se onda kada je za istu vrijednost inputa dobiveno povećanje outputa, ili kada se isti output dobiva korištenjem manjeg broja inputa u odnosu na trenutno stanje. Indikator produktivnosti na makroekonomskoj skali je bruto domaći proizvod (BDP) koji izražava vrijednost proizvedenih dobara u državi tijekom jedne godine, a izražava se u novčanim jedinicama te države. U proizvodnom poduzeću produktivnost se onda računa kao broj proizvedenih proizvoda u odnosu na utrošene sate radnika. Naravno, radni sati ne odražavaju u potpunosti input za izračun, ali su korisna referentna vrijednost kod izračuna u stvarnim analizama [25].

Inputi se mogu podijeliti na 4 glavne skupine [25]:

1. ljudski kapital – znanje i vještine radnika koje se dobivaju iskustvom i edukacijom
2. fizički kapital – oprema i sve strukture korištene u proizvodnji
3. prirodni resursi – obnovljivi izvori energije, voda, ulja i minerali itd.
4. tehnološko znanje – privatno ili javno; privatno je u vlasništvu poduzeća dok je javno dostupno svima.

5.1 Produktivnost vs Učinkovitost

Kod mjerenja produktivnosti zaposlenika bitno je pratiti broj outputa kroz vremenski period, ali bez kvalitete i stvorenog otpada. To bi značilo da zaposlenici koji rade dvostruko brže možda stvaraju više otpada i rade veći broj grešaka, ali su prema definiciji iz jednadžbe 7. produktivniji. Takvi proizvodi u konačnici mogu stvoriti veće troškove poduzeću ako korisnici vraćaju proizvode ili pišu žalbe što može negativno utjecati na imidž poduzeća [25]. Kako produktivnost ne bi bila jedina mjera, potrebno je uvesti i praćenje učinkovitosti. Učinkovitost se definira kao proizvodnja uz stvaranje što je manje moguće otpadnog

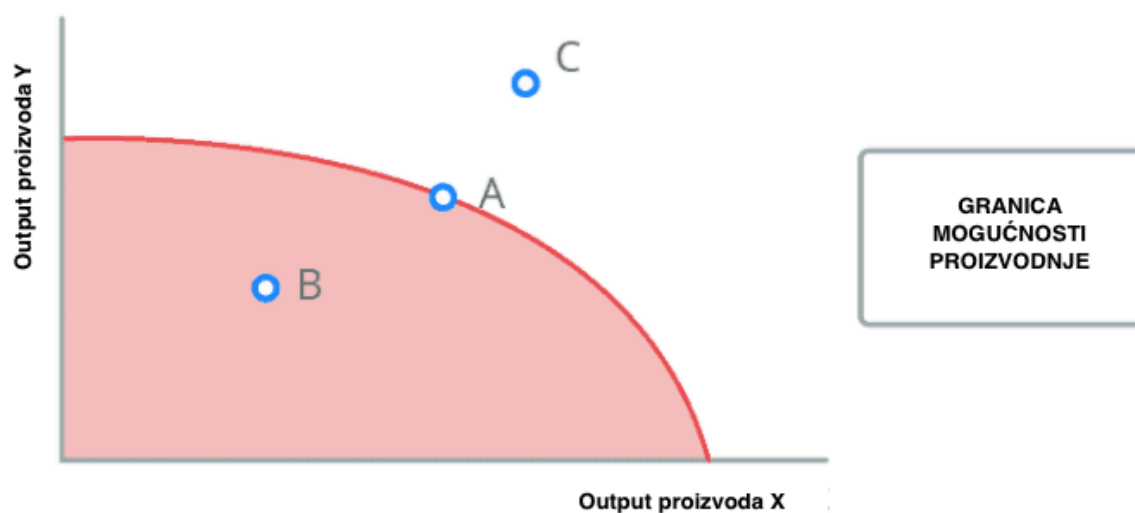
materijala, energije ili vremena i najčešće se izražava kao postotak [%] gdje je 100 % mjera idealne učinkovitosti. Potreban je balans između produktivnosti i učinkovitosti – poduzeće kojem je fokus samo jedno od tog dvoje ne može opstati na tržištu u vremenu kada je konkurencija jača nego ikad. Ako je fokus samo na produktivnosti da se poveća broj proizvoda u određenom vremenu, može se smanjiti učinkovitost. Isto tako ako se povećava učinkovitost, možda neće postojati dovoljna količina proizvoda koja će zadovoljiti potrebe tržišta. Pokazatelj koji objedinjuje ta dva indikatora naziva se proizvodna učinkovitost koja opisuje stanje u kojem poduzeće ne može više proizvesti više proizvoda bez smanjenja proizvodnosti drugog proizvoda. Ono je zapravo mjera maksimalnog kapaciteta gdje se svi dostupni resursi iskorištavaju optimalno kako bi se proizveo neki proizvod. Proizvodna učinkovitost događa se kada poduzeće dosegne granicu mogućnosti proizvodnje. Proizvodna učinkovitost računa se prema formuli 8. [25]:

$$\text{Proizvodna učinkovitost} = \frac{\text{Brzina outputa}}{\text{Standardna brzina outputa}} \times 100 \quad (8)$$

Standardna brzina outputa je maksimalni volumen rada obavljen u jedinici vremena korištenjem standardne metode. Stopostotna proizvodna učinkovitost u praksi je gotovo pa nemoguća i poduzeća se trude balansirati između optimiziranja korištenja resursa, brzine i količine proizvodnje proizvoda ili povećanja kvalitete proizvoda [25].

5.2 Granica mogućnosti proizvodnje

Graf koji prikazuje mogućnost proizvodnje različitih proizvoda i učinak raznih kombinacija prikazuje Slika 19 [25]. On prikazuje različite kombinacije outputa koji se mogu simultano proizvesti koristeći sve dostupne resurse.



Slika 19. Granica mogućnosti proizvodnje

Sve točke (A) koje leže na crvenoj krivulji GMP ilustriraju kombinacije outputa koji su proizvodno učinkoviti. Ako se želi proizvesti više proizvoda X, uz ostvarenje proizvodne učinkovitosti, mora se proizvesti manje proizvoda Y prema krivulji GMP. Točke (B) koje leže ispod krivulje (crvena površina) su proizvodno neučinkovite – proizvodi se više određenog proizvoda no što je potrebno ili se ne proizvodi dovoljno, a točke (C) su trenutno neostvarive upotrebom trenutno dostupnih resursa. Kako se krivulja pomiče prema van i širi, to indicira na pozitivni utjecaj na ekonomiju. Ako se krivulja smanjuje i pomiče prema unutra, to može ukazivati na nedovoljno dobro iskorištavanje resursa i proizvodnje.

5.3 Zašto se mijenja razina produktivnosti?

Postoji nekoliko utjecajnih faktora zašto se mijenja produktivnost poduzeća i oni se mogu pojaviti u različitim organizacijskim trenucima i mogu biti izraženiji u jednom dijelu dok su drugi u takvom postotku da su zanemarivi.

Utjecajni faktori su [25]:

1. tehnički

Osiguravanje (ispravne) opreme za odrađivanje radnog zadatka je preduvjet u poslovanju. Osim opreme tehnički faktori mogu biti vezani uz pravilan raspored rada, lokacija, veličina postrojenja i strojeva, mogućnost automatizacije itd.

2. proizvodni

Proizvodnja treba biti pravilno isplanirana, koordinirana i kontrolirana. Time se postiže optimalan raspored zaliha i korištenja sirovina za izradu novih proizvoda kao i standardizacija procesa.

3. organizacijski

Svaki zaposlenik treba imati jasno definiranu odgovornost i radne zadatke kako bi se izbjeglo sukob, preklapanje istih radnih zadataka između više zaposlenika, ali i jasnoća radne uloge u organizaciji.

4. ljudski

Menadžment ljudskih potencijala je sve više ključan faktor kod izbora kvalitetnih radnika, ali i njihove edukacije i razvoja. Faktori koji utječu na zadovoljstvo radnika mogu biti intrinzični (samoaktualizacija, radoznalost) i ekstrinzični (novac, nagrada, priznanje).

5. menadžerski

Postoje razne metode i stilovi menadžmenta (autokratski, demokratski, slobodni, konzultativni, ...), a odabir ispravnog stila ovisi o organizaciji u kojoj se osoba nalazi. Dobri menadžeri optimalno iskorištavaju dostupne resurse za postizanje maksimalnog učinka uz čim niže troškove koristeći moderne procese i tehnike proizvodnje. Dobar menadžer je ključan za povećanje produktivnosti jer će utjecati i na odabir kvalificiranih i motiviranih radnika, ali i za donošenje odluka vezanih uz unaprjeđenje poslovanja.

6. financijski

Uz svako zapošljavanje i ulaganje u organizaciju (ljudi, oprema, itd.) veže se određeni financijski trošak. Loše upravljanje financijama može uzrokovati smanjenje produktivnosti. Pravilno budžetiranje, ulaganje i praćenje troškova je značajan faktor svake organizacije.

7. lokacijski

Geopolitička pitanja, dostupna infrastruktura, blizina dobavljača, kupaca i radne snage se moraju uzeti u obzir kod odabira mjesta proizvodnje.

5.4 Povećanje produktivnosti

Organizacija može utjecati na povećanje produktivnosti ulaganjem u zaposlenike, strojeve, procese itd. Neki od osnovnih preduvjeta za povećanje produktivnosti su korištenje pravih alata za izvršavanje radnih zadataka, radna okolina u kojoj se nalazi dovoljno prirodnog svjetla, da su čistoća i sigurnost na nivou koji zadovoljava sve zakonske standarde i da je raspored strojeva i ureda optimalan. Smanjenje digitalnog otpada (slanje nepotrebnih e-mail poruka) uz fleksibilne radne sate i održavanje samo bitnih sastanaka utječu na fokus i zadovoljstvo zaposlenika [26].

5.4.1 Vitka proizvodnja (*Lean*) i smanjenje otpada

Lean proizvodnja je metoda popularizirana u Toyoti 1930. godine. Glavni cilj Lean proizvodnje je minimizacija proizvodnje otpada i troškova uz maksimizaciju produktivnosti i kvalitete s mogućnošću kontinuiranog poboljšavanja.

Lean metodologija kategorizira osam tipova otpada [26]:

1. prekomjerna proizvodnja

Previše proizvoda znači potrošnju resursa – novac, energija, sirovine.

2. inventar
Prevelika proizvodnja utječe i na skladištenje, vezani kapital i transport.
3. pokret
Nepotrebni pokreti zaposlenika tijekom rada i oštećenja strojeva utječu na troškove održavanja.
4. defekti
Nedostatak odgovarajuće dokumentacije, velike razlike u inventaru, loš dizajn.
5. pretjerana obrada
Bilo koji dio proizvodnog procesa koji je nepotreban – proizvodnja izvan opsega projekta, radi loše komunikacije ili dupliciranih podataka.
6. čekanje
Čekanje ljudi, materijala ili strojeva direktno utječe na izgubljeno vrijeme kada je proizvodnja usporena ili obustavljena.
7. transport
Premještanje materijala s jednog mjesta na drugo – ne dodaje vrijednost proizvodu!
8. neiskorišteni talent
Otpad koji nije povezan s proizvodnjom, već s neiskorištavanjem potencijala zaposlenika.

Svaki od ovih osam otpada se javlja u većem ili manjem obujmu unutar svakog proizvodnog poduzeća bez obzira na razlike u proizvodnji ili vrsti proizvoda koji se izrađuje i prodaje. Lean proizvodnja direktno utječe na smanjenje troškova i povećanje vrijednosti poduzeća primjenom sistematičnog i kontinuiranog procesa micanja aktivnosti koji ne dodaju vrijednost.

5.4.2 Strategije smanjenja otpada

Identifikacijom otpada kojeg poduzeće proizvodi može se značajno smanjiti njegov obujam. Neke od metoda su [26]:

1. implementacija efikasnog inventarnog sustava
Umjesto ručnog unošenja informacija o stanju inventara i proizvodnje (što može dovesti do ljudske pogreške u vidu krivo zapisanih informacija), mogu se koristiti specijalizirani programski paketi koji pružaju potporu u informiranijem i pametnijem donošenju odluka.
2. vrsta pakiranja proizvoda

Korištenjem materijala koji se mogu reciklirati, obnoviti ili ponovno koristiti smanjuje se količina otpada.

3. postojanje rasporeda za održavanje

Preventivno održavanje je puno jeftinije od korektivnog i zato treba uložiti u redovno održavanje strojeva i alata, pogotovo za strojeve koji se često koriste. Dodatno se može napraviti i plan rada ako neki od strojeva prestane funkcionirati te ga podijeliti sa zaposlenicima kako bi se smanjilo vrijeme čekanja.

4. organizacija skladišta

Jedna od najčešćih pogrešaka prilikom dizajniranja skladišta je loš raspored elemenata skladišta. Efikasna skladišta skladište proizvode čim bliže otpremnom području kako bi se smanjilo vrijeme i put za prihvat. Ovisno o tipu proizvodnog poduzeća i sezonalnosti proizvoda, potreban je češći osvrt na raspored elemenata i funkcionalnost skladišta.

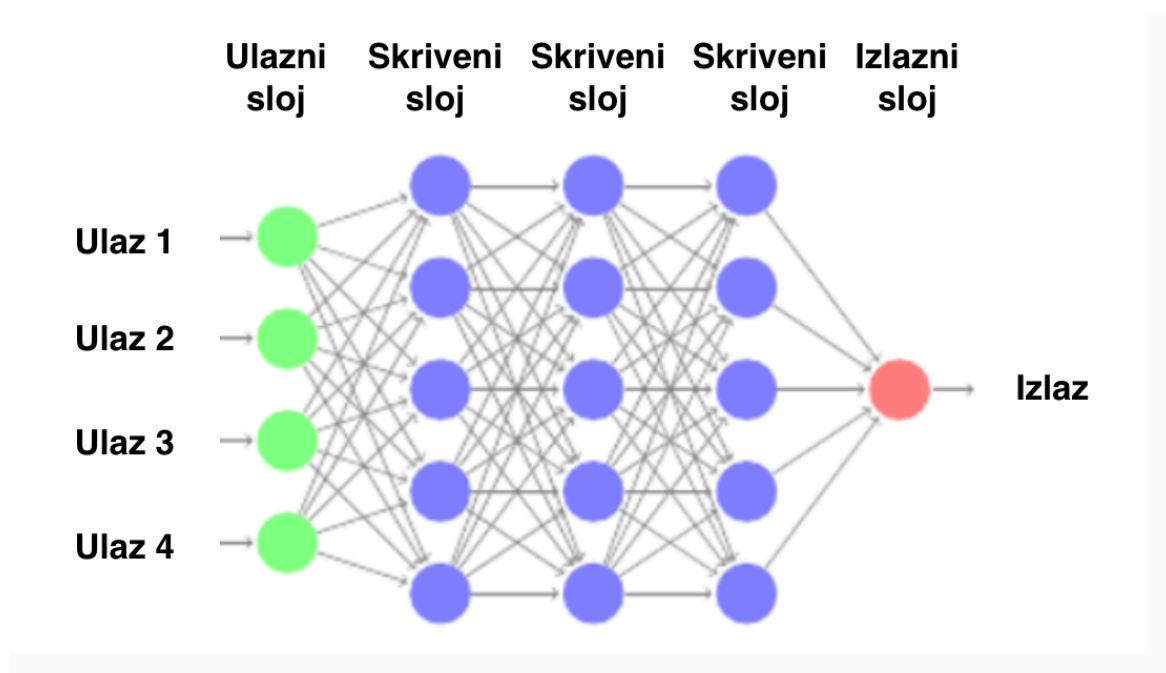
5. označavanje

Označavanje proizvoda, polica, ruta prolaska je bitno u skladištima kako bi se smanjila mogućnost prihvata i otpreme krivog proizvoda.

Pametna skladišta i tvornice koriste metode umjetne inteligencije, senzore i ostalu tehnologiju za povećanje proizvodnosti i efektivnosti.

6 Duboko učenje

Jedan od algoritama kako se gradi process učenja neuronske mreže naziva se perceptron koji se sastoji od ulazne jedinice koja je povezana s izlaznim čvorom, a učenje uzoraka se definira preko aktivacijske funkcije. Za rješavanje složenijih problema kreirane su umjetne neuronske mreže (eng. Artificial Neural Network – ANN) koje se nazivaju višeslojnim perceptronima. Proces učenja odvija se tako da svaki put kad je uvedena nova metrika, težine i pristranosti se prilagođavaju pomoću novog treniranja i korekcije greške preko algoritma povratne propagacije, a taj postupak se iterira više puta. Dodavanjem više skrivenih slojeva u mrežu stvara se dublja struktura koja može učiti raspoznavati kompleksnije, i brojnije, značajke i uzorke. Arhitekture mreža s više slojeva se nazivaju dubokim neuronskim mrežama. Problem koji se javlja kod učenja takve neuronske mreže je taj da se ispravljanje greške, dodavanjem novih slojeva mreže, u učenju ne širi prema početnom sloju mreže i time se ometa process učenja [27]. Jednostavna arhitektura duboke neuronske mreže prikazuje Slika 20 [27]:



Slika 20. Jednostavni prikaz neuronske mreže

Duboko učenje (eng. Deep Learning) je specijalizirano područje unutar domene umjetne inteligencije koje uči na strukturiranim i nestrukturiranim podacima. Arhitektura dubokog učenja sastoji se dubokih slojeva neuronskih mreža kao što su ulazni sloj, skriveni slojevi i izlazni sloj. Skriveni slojevi koriste se za razumijevanje podataka, a neuronsku mrežu nije potrebno programirati jer ona uči na velikim količinama podataka [27].

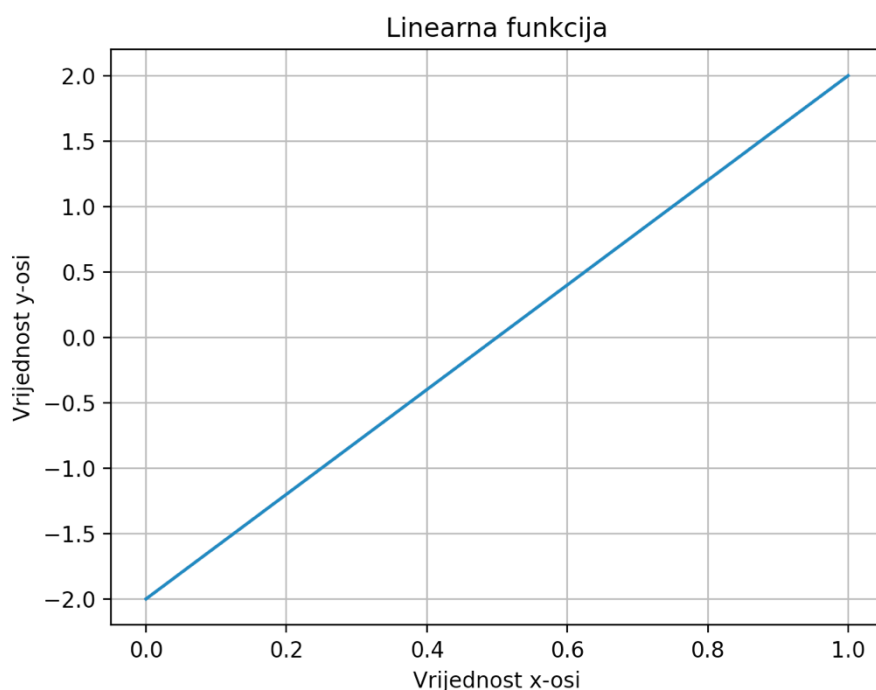
6.1 Aktivacijske funkcije

U praksi se najčešće koriste četiri tipa aktivacijskih funkcija:

1. linearna
2. sigmoidalna
3. step
4. linearna rektifikacijska funkcija – ReLU.

Linearna aktivacijska funkcija preslikava svoj ulaz koji se množi s određenom konstantom i daje izlaznu vrijednost, a definira se kao:

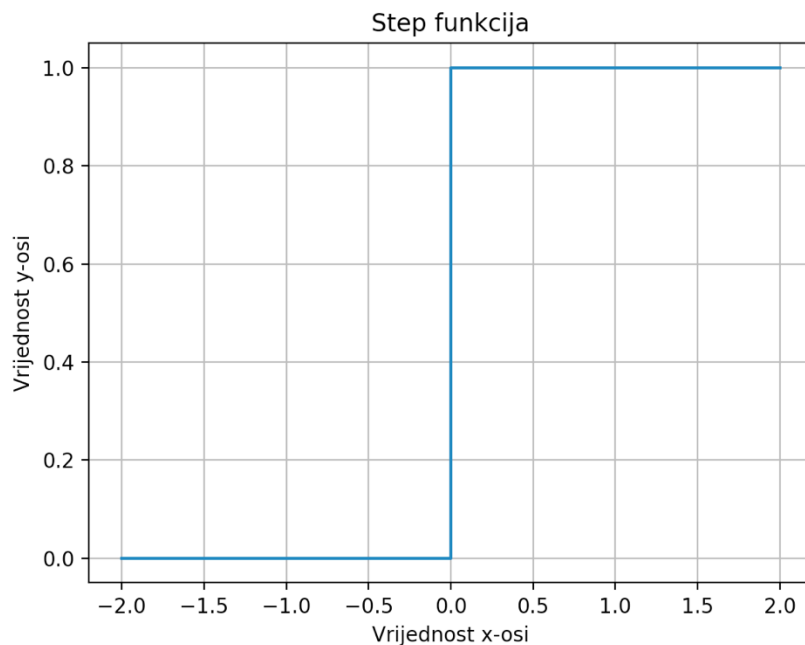
$$f(x) = ax, \quad A \in \mathbb{R} \quad (9)$$



Slika 21. Linearna funkcija

Step funkcija je binarna funkcija koja ima dvije vrijednosti – nula i jedan. Definiira se tako da je vrijednost izlaza jednaka 1 ako je izlaz veći od neke težine w , a u suprotnom je 0. Step funkcija za $w=0$ izgleda ovako:

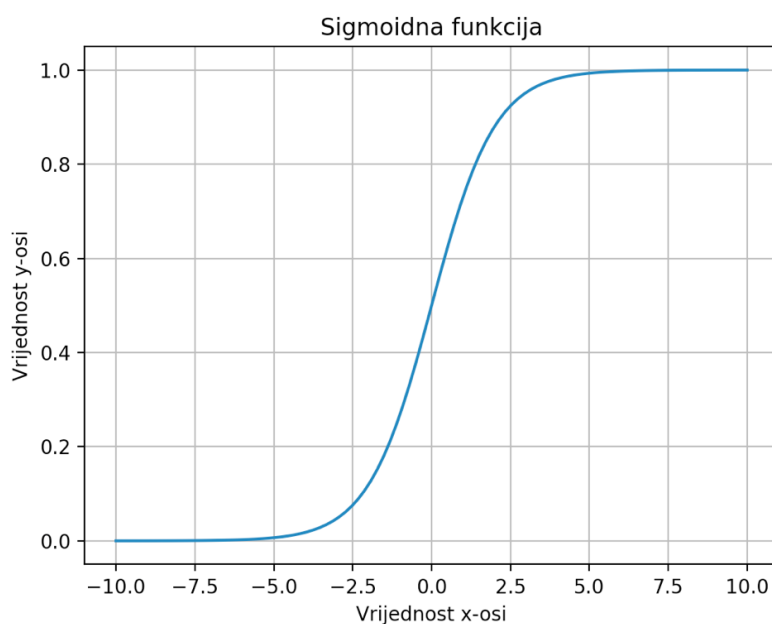
$$f(x) = \begin{cases} 1, & \text{za } x \geq 0 \\ 0, & \text{za } x \leq 0 \end{cases} \quad (10)$$



Slika 22. Step funkcija

Sigmoidalna funkcija je vrlo slična step funkciji, ali nije toliko skokovita niti je binarna već ima glatke rubove. Za razliku od step funkcije, ona omogućuje svojstvo nelinearnosti i može se koristiti kao točniji estimator klasifikacije. Matematički se zapisuje kao:

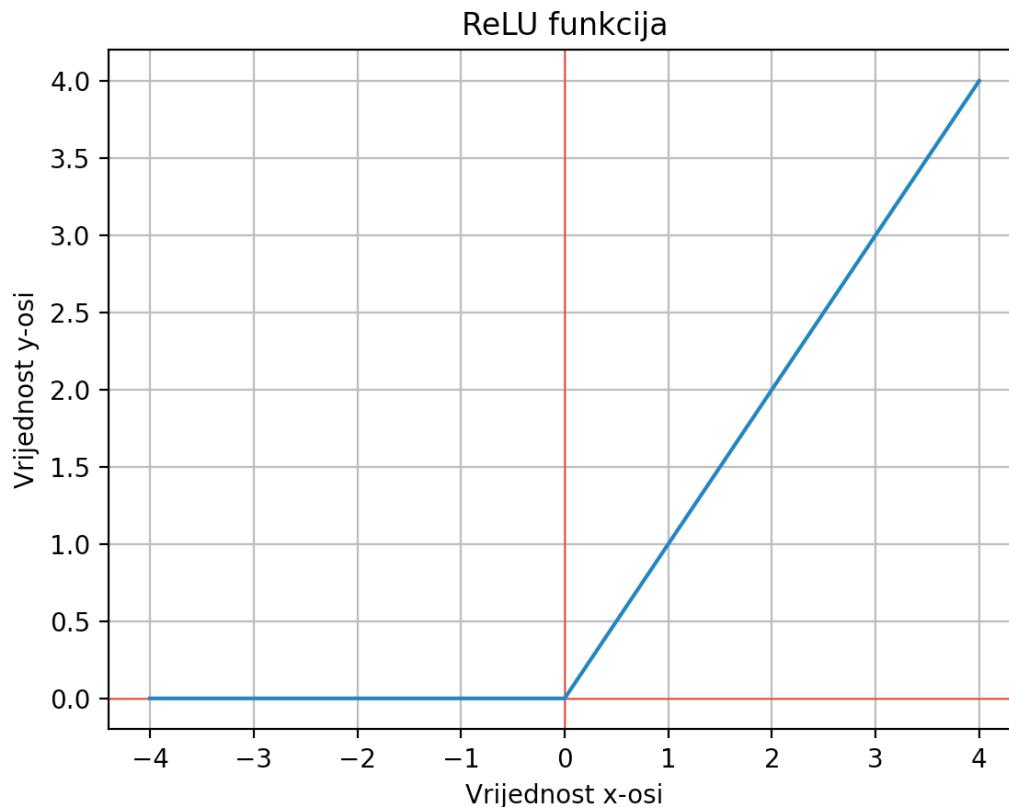
$$f(x) = \frac{1}{1 + e^{-f}} \quad (11)$$



Slika 23. Sigmoidna funkcija

ReLU funkcija je korisna radi svoje nelinearnosti i gradijenta koji se uvijek nula ili jedan, a definira se kao:

$$f(x) = \max(0, x) \quad (12)$$

**Slika 24. ReLU funkcija****6.2 Hiperparametri**

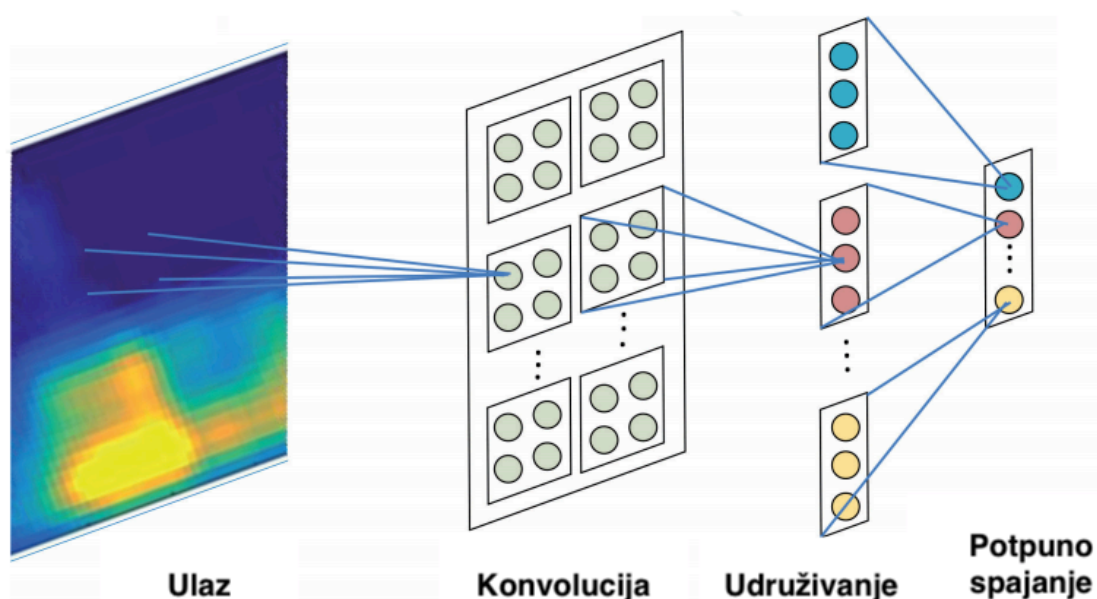
Hiperparametri su varijable koje je potrebno definirati prije učenja podataka, a optimizacija parametara uključuje pronalazak optimalnih hiperparametara mreže. U hiperparametre ubrajamo [28]:

- stopa učenja
- broj epoha
- arhitektura mreže
- aktivacijska funkcija
- inicijalizacija težina

- obrada ulaza
- slojevi poolinga
- funkcija pogreške.

6.2.1 Konvolucijska neuronska mreža

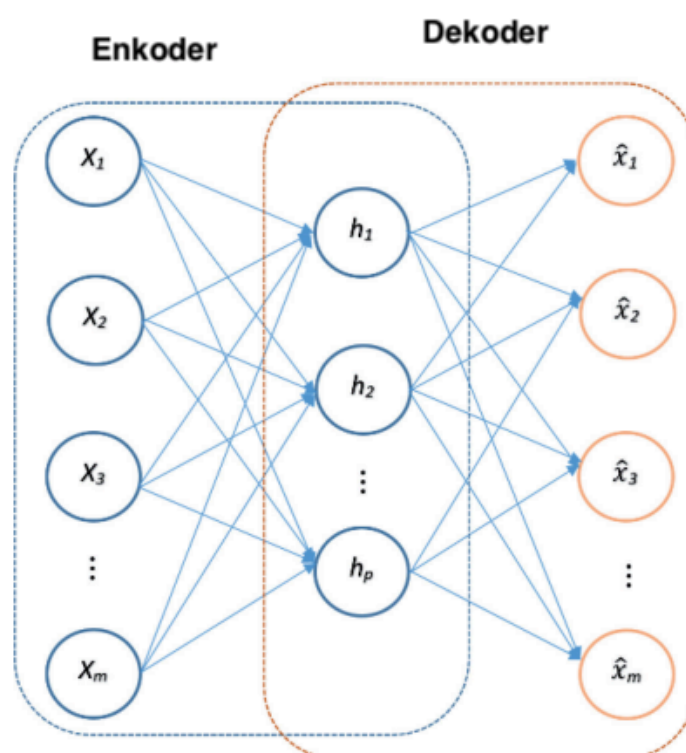
Konvolucijska neuronska mreža (CNN) je višeslojna umjetna neuronska mreža kojoj je prvotna namjena bila dvodimenzijaska obrada slike. Također se u nekoliko slučajeva koristi za obradu prirodnog jezika (eng. Natural Language Processing - NLP) i prepoznavanje govora. CNN ima tri glavna obilježja – konvolucija, udruživanje i potpuno povezivanje. Kod CNN-a učenje značajki je dobiveno izmjenjivanjem i slaganjem konvolucijskih slojeva i operacijama udruživanja. Slojevi se konvertiraju sa sirovim ulaznim podacima koristeći višestruke lokalne filtere (neurone) i generiraju lokalne značajke. Slojevi za udruživanje ekstrahiraju najznačajnije značajke fiksne duljine pomoću operatora maksimalnog združivanja i prosječnog združivanja. Maksimalno udruživanje izabire maksimalnu vrijednost jedne regije na mapi značajki kao najznačajniju. Prosječno udruživanje izračunava srednju vrijednost jedne regije i koristi ju kao vrijednost udruživanja. Maksimalno udruživanje pogodno je za izdvajanje rijetkih značajki dok se postupak udruživanja na cijelom skupu podataka ne koristi često jer nije optimalan. Nakon višeslojnog učenja značajki, potpuno povezani slojevi pretvaraju dvodimenzionalnu kartu značajki u jednodimenzionalni vektor i ubacuju ga u *softmax* funkciju za kreiranje modela [29]. Konvolucijsku neuronsku mrežu prikazuje Slika 25 [30].



Slika 25. Prikaz konvolucijske mreže

6.2.2 Autoenkoder

Autoenkoder (eng. Auto Encoder – AE) je nenadzirani algoritam korišten za ekstrakciju značajki iz ulaznih podataka bez anotiranih informacija. Uglavnom se sastoji od dva dijela – enkoder i dekoder. Enkoder može napraviti kompresiju podataka (bitno kada je riječ o višedimenzionalnosti podataka) preslikavanjem ulaza na skriveni sloj. Dekoder može rekonstruirati aproksimaciju ulaza [30]. Slika 26 prikazuje jednostavnu shemu AE [30].

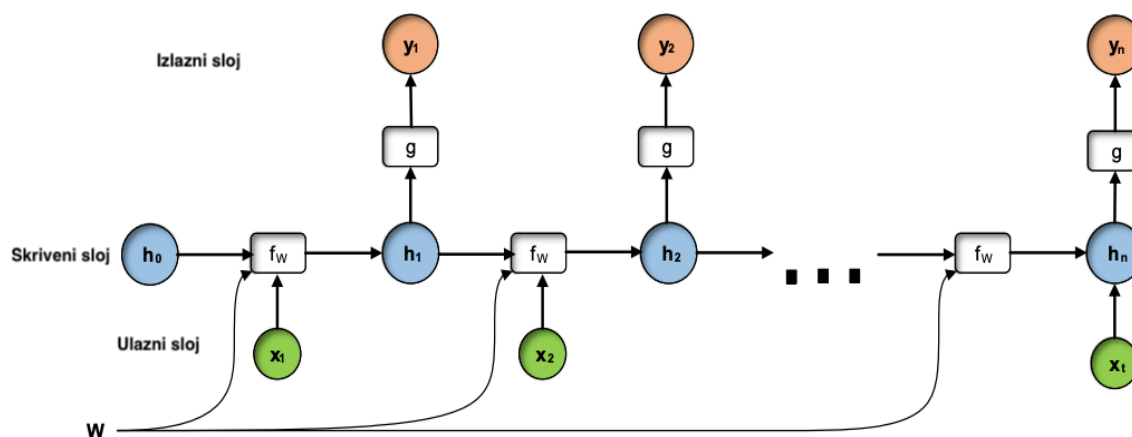


Slika 26. Prikaz autoenkodera

6.2.3 Ponavljajuća neuronska mreža

Ponavljajuća neuronska mreža (RNN) ima unikatne karakteristike topoloških veza između neurona koji su stvorili usmjerene cikluse za sekvence podataka. RNN je pogodan za učenje značajki iz sekvence podataka. Omogućuje zadržavanje informacija u skrivenom sloju i bilježi stanja od nekoliko koraka prije. Uzme li se sekvencijalni set podataka kao ulaz, trenutno stanje skrivenog sloja može se izračunati iz dva dijela kroz istu aktivacijsku funkciju (sigmoidna funkcija ili hiperbolički tangens – tanh). Prvi dio izračunava se ulazom, a drugi se računa iz skrivenog stanja u prethodnom vremenskom koraku. Izlaz se zatim može izračunati preko trenutnog skrivenog stanja koristeći *softmax* funkciju. Obradom cijele sekvence, skriveno

stanje je naučeni prikaz ulaznih sekvenci podataka. Slika 27. prikazuje jednostavnu shemu RNN-a [29].



Slika 27. Prikaz ponavljajuće neuronske mreže

Razlike između CNN, AE i RNN prikazuje Slika 28 [29].

<i>Model</i>	<i>Princip korištenja</i>	<i>Prednosti</i>	<i>Nedostaci</i>
<i>CNN</i>	Značajke su učene koristeći sloj za uzorkovanje i naslagani konvolucijski sloj	Smanjen broj parametara, nepromjenjivost pomaka, skale i distorzije	Visoka računaska složenost, vremenski dugo trajanje optimizacije združenih parametara
<i>AE</i>	Nenadzirano učenje značajki i dimenzionalnosti podataka smanjeno je korištenjem enkodera	Irelevantnost u ulazu je eliminirana i sačuvani su samo smisleni podaci	Proširivanje pogreške sloj po sloj i rijetke reprezentacije nisu garantirane
<i>RNN</i>	Vremenski uzorak pohranjen u ponavljajuće neuronske veze i distribuirana skrivena stanja za podatke vremenskih serija	Kratkoročne informacije su zadržane, a vremenske korelacije su zadržane u sekvencijskim podacima	Teško je trenirati model i imati dugotrajnu ovisnost

Slika 28. Razlike između neuronskih mreža

6.3 Funkcija gubitka

Svaka iteracija zahtjeva izračun greške i ta greška se može računati na nekoliko načina. Funkcija gubitka modela bira se u ovisnosti o vrsti problema te tako može biti [31]:

- gubitak za regresiju
- gubitak za klasifikaciju
 - binarna
 - više-klasna.

Spomenuti optimizacijski algoritmi se mogu koristiti za minimizaciju ili maksimizaciju funkcije gubitka i oni mogu biti algoritmi s derivacijom prvog stupnja gdje se koristi gradijent funkcije za određivanje vrijednosti funkcije u nekoj točki dok se funkcije s derivacijom drugog stupnja koriste za izračun pada ili rasta njenog gradijenta [31].

6.3.1 MSE

Srednja kvadratna pogreška (eng. Mean Square Error – MSE) je prosjek kvadrata svih grešaka . Pogreška je zapravo razlika između promatrane i vrijednosti koju želimo predvidjeti bez određivanja smjera funkcije (rast ili pad). Predviđanja dalja od stvarne vrijednosti imaju veću MSE i vrijednost greške je tim veća (radi operacije kvadriranja). Ova funkcija ima stabilan smjer kretanja u odnosu na početni minimum i prikazuje povećanje pogreške na određeni način za razliku od apsolutne pogreške (eng. Mean Absolute Error – MAE) koja prikazuje prosjek apsolutne vrijednosti [31]. Formula prema kojoj se računa MSE je:

$$MSE = \frac{\sum_{i=1}^n (y_i - \bar{y}_i)^2}{n} \quad (13)$$

6.3.2 Negativna log-izglednost

Negativna log-izglednost (eng. Negative Log Likelihood – NLL) je funkcija greške za problem klasifikacije. Pogreška se računa kao odstupanje vrijednosti koju želimo predvidjeti i stvarne vrijednosti. Ovo je funkcija vjerojatnosti kojom se dobiva postotak točnosti klasifikacije s vrlo malom pogreškom [31]. Formula za izračun je:

$$NLL = -(y_i \log(\bar{y}_i) + (1 - y_i) \log(1 - \bar{y}_i)) \quad (14)$$

7 Primjena dubokog učenja u proizvodnji

Razvojem tehnologija pametne tvornice fokusiraju se na stvaranje proizvodne inteligencije koja će imati pozitivan učinak na cijelo poduzeće. Današnja proizvodnja bilježi rast u dostupnosti senzornih podataka koji se sastoje od različitih struktura, semantike i formata. Podaci su prikupljeni s raznih aspekata proizvodnog poduzeća – proizvodna linija, oprema, proces, radna aktivnost i okolišni uvjeti. Radi tog se javlja potreba za modeliranjem i analizom podataka. Rudarenjem znanja identificiraju se obrasci u podacima koji omogućuju informirano donošenje odluka. Metode analize podataka uključuju [29]:

- deskriptivnu
- dijagnostičku
- prediktivnu
- preskriptivnu analitiku.

Deskriptivna analitika ima za cilj sažeti informacije o tome što se događa koristeći proizvodne uvjete, okoliš i operativne parametre. Dijagnostička analitika koristi se kod ispitivanja uroka smanjena performansi proizvoda ili kvara opreme. Prediktivna analitika koristi statističke modele za predviđanje mogućnosti buduće proizvodnje ili degradacije opreme koristeći dostupne historijske podatke. Preskriptivna analitika ide i korak dalje tako da preporuča jedan ili više načina djelovanja na temelju analiziranih podataka. Za svaku odluku može se izračunati vjerojatnost ishoda. Uz naprednu analitiku koju pruža duboko učenje, proizvodnja se pretvara u visoko optimizirane pametna postrojenja. Dobivanje znanja o proizvodnji korištenjem dubokog učenja omogućeno je pomoću nekoliko arhitektura – konvolucijske neuronske mreže (eng. Convolutional Neural Network), autoenkoderi (eng. Auto Encoder) i ponavljajuća neuronska mreža (eng. Recurrent Neural Network) [29].

7.1 Deskriptivna analiza

Jedna od najčešćih primjena deskriptivne analize je kod korištenja računalnog vida i procesuiranja slike za obradu i detekciju površinskih grešaka tijekom izrade proizvoda. Iako postoje mnoga gotova rješenja za analizu slika, kod proizvodnje je potrebno uračunati u varijacije u proizvodima i njihovu potencijalnu kompleksnost kako bi se tijekom izrade modela koji analizira proizvod uzelo u obzir sve značajke tog proizvoda. Kod tradicionalnog strojnog učenja je to veliki problem zato što inženjer mora sam definirati koje su to značajke na kojem se algoritam uči, no upravo je tu prednost dubokog učenja koje ne zahtijeva točne

specifikacije značajki za učenje. CNN mreže su originalno bile zamišljene za analizu slike i zato im je česta primjena upravo u analizi površinskih oštećenja i kontrole kvalitete proizvoda. Arhitektura dubokih konvolucijskih neuronskih mreža je dizajnirana tako da su hiperparametri optimizirani korištenjem povratne propagacije i algoritma stohastičkog gradijentnog spusta [29].

7.2 Dijagnostička analiza

Proizvodni sustavi i strojevi su podložni trošenju i kvarovima uzrokovani degradacijom, manjkom održavanja, neispravnim korištenjem ili lošim uvjetima u kojima se strojevi nalaze. Svaki kvar uzrokuje dodatan trošak, smanjenje produktivnosti i u konačnici zastoje u radu. Korištenjem senzora i automatiziranih sustava uz algoritme dubokog učenja moguće je pratiti i klasificirati uzorke kvarova. CNN i AE se mogu koristiti za dijagnostiku zrakoplovnih motora, vjetroturbina ili manjih dijelova poput ležajeva ili mjenjača [29].

7.3 Prediktivna analiza

Uz već spomenute analize, bitno je imati razvijen sustav prediktivne analitike kako bi produktivnost ostala zadržana uz smanjene troškove održavanja. Razvojem tehnologije moguće je predvidjeti kada je potrebno uložiti vrijeme u održavanje opreme i strojeva. Kako bi se dobili podaci potrebni za učenje nekog od algoritma potrebno je imati historijske zapise o vremenima održavanja, načinima održavanja, cijenama itd [29].

8 Prikaz implementacije modela neuronskih mreža

U radu su navedene razlike između tradicionalnog pristupa izradi modela tradicionalnih neuronskih mreža i algoritama u odnosu na izradu modela dubokog učenja. Tradicionalni pristup zahtjeva veliku pažnju posvećenju definiranju ključnih značajki koje donose vrijednost u predviđanju ili klasifikaciji određene varijable, dok DL to ne zahtijeva. U oba slučaja je potrebno imati čim veći i raznovrsniji/reprezentativniji set podataka na kojima se modeli uče. Prije samog učenja potrebno je uložiti vrijeme u analizu i predobradu seta podataka, a duljina trajanja tog procesa ovisi o izabranoj arhitekturi i modelu mreže.

Za pokretanje modela korišten je *open-source* programski alat PyCharm (integrirana razvojna okolina) verzije 2020.2.3., a jezik pisanja koda je Python verzija 3.6.

OpenCV (verzija 4.4.0.42) je javno dostupna biblioteka koja ima set predefiniраниh funkcija dostupnih za korištenje u realnom vremenu za rješavanje problema računalnog vida.

8.1 Tradicionalni pristup rješavanju problema klasifikacije

Stablo odlučivanja (eng. Decision Tree - DT) je struktura gdje svaki čvor predstavlja jednu značajku (atribut), grana predstavlja odluku, a svaki list je zapravo rezultat te odluke. Prvi čvor je korijenski čvor stabla iz kojeg se donose odluke. Stablo odlučivanja je vizualizacijski prikaz podataka oblika dijagrama toka što olakšava razumijevanje u donošenju odluka umjesto analize samo numeričkih zapisa. Rastom broja značajki i odluka povećava kompleksnost DT-a, ali je on generalno brži od nekih drugih algoritama strojnog učenja. DT je neparametarski algoritam koji ne ovisi o pretpostavkama raspodjele vjerojatnosti i mogu s velikom točnošću predviđati visokodimenzionalne podatke [32].

Algoritam se uči tako da se dostupni podaci raspodijele na skup za učenje i testiranje najčešće u omjeru 30/70. Odabrana značajka je prvi čvor i na temelju nje se podaci raspoređuju na manje podskupove. Stablo se rekurzivno gradi dok se ne zadovolji jedan od uvjeta [32]:

1. svi korijeni pripadaju istom atributu
2. nema više preostalih atributa
3. nema više instanci.

Odabir značajke se najčešće može odviti na dva načina koristeći informacijski dobitak (eng. Information Gain) ili Gini indeks (eng. Gini Indeks).

Koncept entropije koja mjeri nečistoću ulaznog skupa se koristi kod informacijskog dobitka. Dobitak je smanjenje entropije odnosno to je razlika između entropije prije raspodjele skupa podataka i prosječne vrijednosti entropije nakon raspodjele za određene vrijednosti značajki.

Gini indeks definira vjerojatnost da određeni korijen u skupu D pripada klasi C . Za binarnu raspodjelu atributa A za podatke D u D_1 i D_2 onda se Gini indeks računa kao [32]:

$$GINI_A(D) = \frac{|D_1|}{|D|} GINI(D_1) + \frac{|D_2|}{|D|} GINI(D_2) \quad (15)$$

Za diskretne vrijednosti atributa, podskup koji ima najmanju vrijednost Gini indeksa se koristi kao atribut za raspodjelu.

8.1.1 Implementacija i analiza rezultata

Za učitavanje podataka korištena je *open-source* biblioteka *pandas* koja omogućava rad s podacima u tabličnom formatu, a za dodatne vizualizacije je korištena *open-source* biblioteka *seaborn*. Podaci su dobiveni na Fakultetu strojarstva i brodogradnje u Zagrebu koji su služili za istraživački rad na temu „Solving problems of interruptions and multitasking in the pharmacy of a large hospital center“ [33]. Podaci su u izvornom obliku bili dostavljeni u *.csv* formatu kako prikazuje Slika 29

	Naziv_dobavljacka	Vrsta_robe	Datum_isporuke	Vrijeme_dolaska_tp	Vrijeme_odlaska_tk	Vrijeme_rada_tdob	km_velicina	km_velicina4	velicina_org	velicina_2	period
1	Agmar	Lijekovi	17.11.2015	11:00:00	11:04:00		4 mala	mikro	mali	mali	11.00-11.30
2	Agmar	Lijekovi	2.12.2015	10:49:00	10:51:00		2 mala	mikro	mali	mali	10.30-11.00
3	B.Braun	Potrosni materijal	17.11.2015	10:22:00	10:43:00		21 mala	mala	srednji	srednji	10.00-10.30
4	B.Braun	Potrosni materijal	2.12.2015	10:07:00	10:09:00		2 mala	mikro	mali	mali	10.00-10.30
5	Bioelektronika	Potrosni materijal	17.11.2015	10:57:00	10:58:00		1 mala	mikro	mali	mali	10.30-11.00
6	Boston Medical	Potrosni materijal	2.12.2015	09:28:00	09:30:00		2 mala	mikro	mali	mali	9.00-9.30
7	Boston Medical	Potrosni materijal	3.12.2015	10:45:00	10:46:00		1 mala	mikro	mali	mali	10.30-11.00
8	Copan Zagreb	Potrosni materijal	17.11.2015	10:30:00	10:32:00		2 mala	mikro	mali	mali	10.30-11.00
9	Copan Zagreb	Potrosni materijal	3.12.2015	10:37:00	10:48:00		11 mala	mala	mali	mali	10.30-11.00
10	Copan Zagreb	Potrosni materijal	4.12.2015	10:39:00	10:45:00		6 mala	mikro	mali	mali	10.30-11.00

Slika 29. Tablični podaci za obradu

Učitavanje podataka radi se koristeći funkciju *read_csv* gdje se definira putanja na kojoj se nalazi datoteka, a *data.info* je funkcija koja daje informacije o podacima (broj redova, nazivi stupaca itd.). Analizom je utvrđeno da su neki stupci redundantni i oni su uklonjeni koristeći funkciju *data.drop*.

```

import pandas as pd
import seaborn as sns

data = pd.read_csv('/Users/karlopuzak/Downloads/analizaLSS.csv') #Učitavanje .csv datoteke
data.info() #Prikaz informacija o podacima

columns_to_drop = ['Unnamed: 0', 'Naziv_dobavljacka', 'Datum_iskoruke', 'Vrijeme_dolaska_tp', 'Vrijeme_odlaska_tk',
                  'km_velicina', 'km_velicina4', 'velicina_2', 'period'] #Brisanje redundandnih stupaca

data = data.drop(columns_to_drop, axis=1)
data.info()
sns.heatmap(data.isnull()) #Vizualizacija podataka za istraživanje postoje li redovi bez vrijednosti
print(sns.heatmap) #Prikaz vizualizacije

```

Slika 30. Učitavanje podataka u Python

Korištenjem funkcije *data.info* dobiven je uvid u broj redova i stupaca te njihove vrijednosti kao što prikazuje Slika 31. Ostavljeni su stupci *Vrsta_robe*, *Vrijeme_rada_tdob* i *velicina_org* jer se na temelju vrste robe i vremena obrade želi predvidjeti koja je veličina organizacije u pitanju.

```

RangeIndex: 162 entries, 0 to 161
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Unnamed: 0            162 non-null   int64
1   Naziv_dobavljacka     162 non-null   object
2   Vrsta_robe            162 non-null   object
3   Datum_iskoruke       162 non-null   object
4   Vrijeme_dolaska_tp   162 non-null   object
5   Vrijeme_odlaska_tk   162 non-null   object
6   Vrijeme_rada_tdob     162 non-null   int64
7   km_velicina          162 non-null   object
8   km_velicina4         162 non-null   object
9   velicina_org         162 non-null   object
10  velicina_2           162 non-null   object
11  period                162 non-null   object
dtypes: int64(2), object(10)
memory usage: 15.3+ KB

```

Slika 31. Informacije o podacima prije obrade

```

RangeIndex: 162 entries, 0 to 161
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Vrsta_robe            162 non-null   object
1   Vrijeme_rada_tdob     162 non-null   int64
2   velicina_org         162 non-null   object
dtypes: int64(1), object(2)

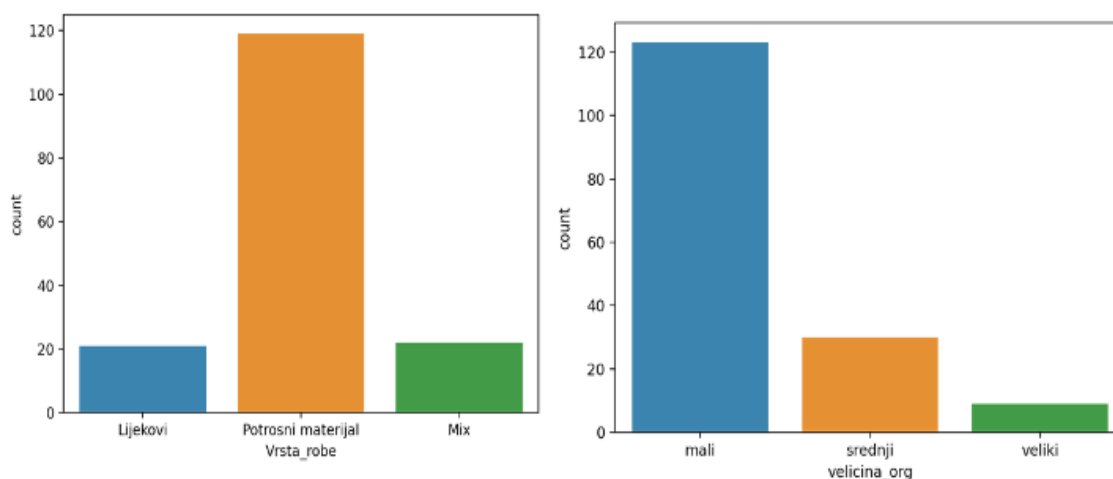
```

Slika 32. Podaci poslije obrade

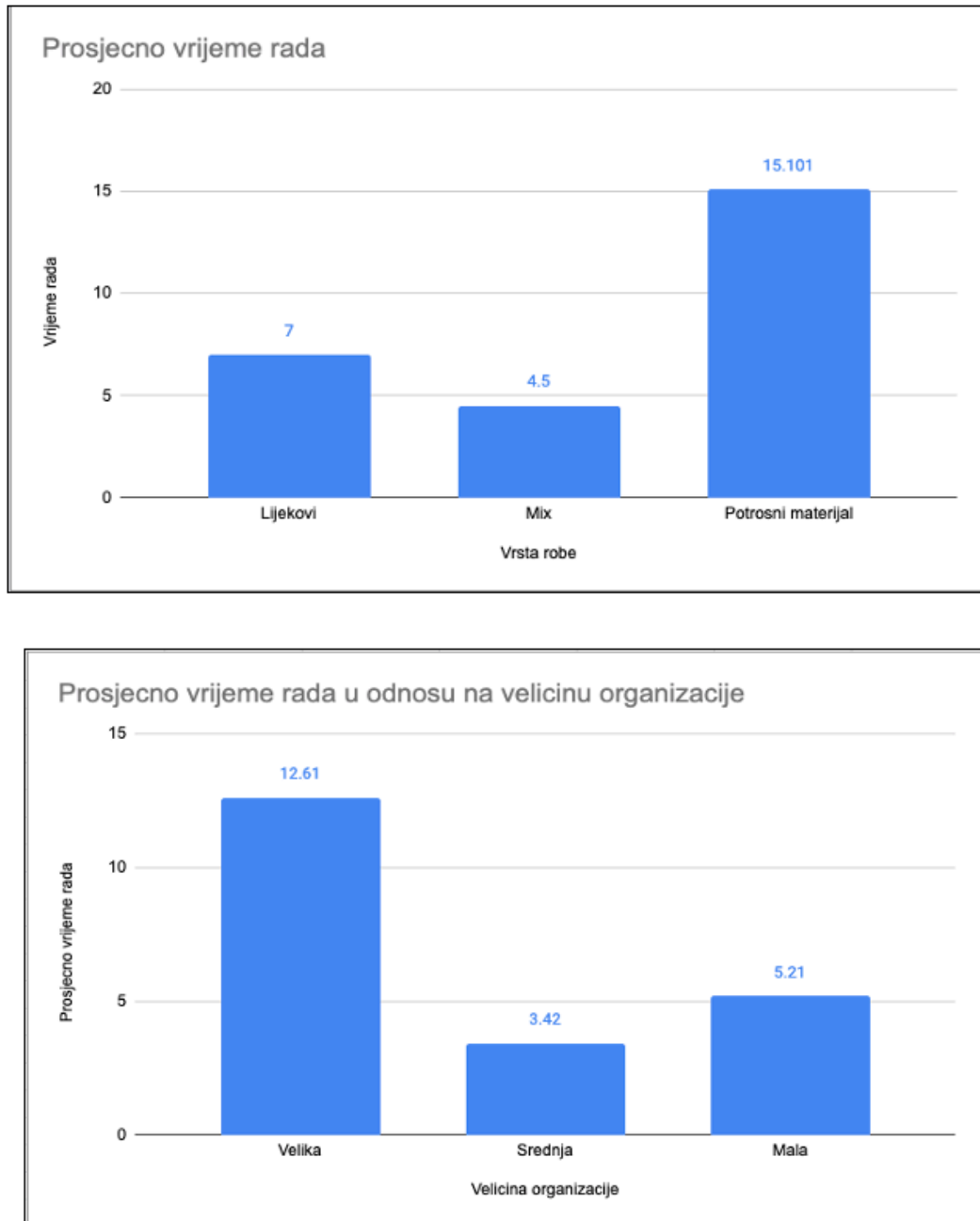
Funkcija `isnull()` za skup podataka naziva dana pretražuje sve stupce i redove i traži ćelije koje nemaju vrijednosti u sebi, a vizualizacija je napravljena pomoću funkcije `heatmap` kako prikazuje. gdje je vidljivo da niti jedna ćelija nije prazna, odnosno da za svaki stupac i red postoji definirana vrijednost.



Slika 33. Pretraživanje praznih vrijednosti



Slika 34. Suma ključnih podataka za obradu



Slika 35. Prosječna vremena rada

Vidljivo je da najviše vremena na rad odlazi u velikim organizacijama i da se najviše koristi potrosni materijal što je i logično jer ga ima najviše pa je i vrijeme obrade najduže.

S obzirom na to da se koristi metoda stabla odlučivanja, potrebno je tekstualne podatke pretvoriti u numeričke vrijednosti kako je prikazano na slici Slika 36.


```
VrstaDummy = pd.get_dummies(data['Vrsta_robe']) #Kreiranje dummy vrijednosti kako bi se dobile integer vrijednosti
X = pd.concat((VrstaDummy, data['Vrijeme_rada_tdob']), axis=1) #Spajanje vrijednosti za vrstu robe i vrijeme
y = data['velicina_org'].values #Definiranje vrijednosti koja se traži
```

Slika 36. Optimizacija podataka

Open-source biblioteka sklearn sadrži razne algoritme strojnog učenja i ovdje je odabran algoritam DecisionTreeClassifier. Set podataka je podijeljen na 30 % podataka za učenje i 70 % za testiranje modela kako prikazuje Slika 37. Postotak točnosti modela je: 0,979592 %.

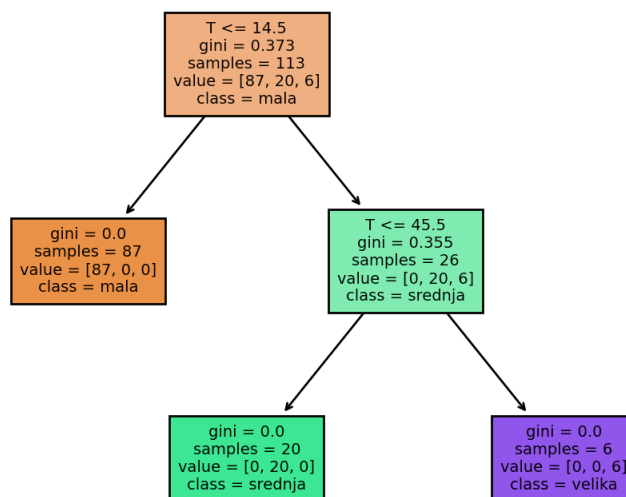
```
from sklearn.model_selection import train_test_split #Učitavanje modula za raspodjelu dataseta
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0) #Raspodjela dataseta

from sklearn import tree #Učitavanje modula za korištenje DT-a
dt_clf = tree.DecisionTreeClassifier(max_depth=9) #Definiranje vrijednosti DT-a
dt_clf.fit(X_train, y_train) #Podešavanje podataka za treniranje algoritma

dt_clf.score(X_test, y_test) #Podešavanje vrijednosti predviđanja
y_pred = dt_clf.predict(X_test) #Definiranje varijable koja se predviđa

print("Postotak točnosti Decision Tree:", dt_clf.score(X_test, y_test))
tree.plot_tree(dt_clf, fontsize=7, filled=True, feature_names=("P1", "P2", "P3", "T"),
               class_names=("mala", "srednja", "velika")) #Ispis rezultata predviđanja
```

Slika 37. Učitavanje modela



Slika 38. Grafički prikaz rada modela

8.1.2 Razlike između algoritama

Usporedbe radi korištena su još dva algoritma – algoritam slučajnih šuma i k-NN algoritam. Algoritam slučajnih šuma (eng. Random Tree Forest – RTF) se sastoji od više stabala odluka. Razlika između RTF i DT je ta što RTF uzima nasumične vrijednosti atributa dok se kod DT-a uzimaju svi atributi za treniranje. Programski kod za RTF algoritam prikazan je na slici Slika 39.

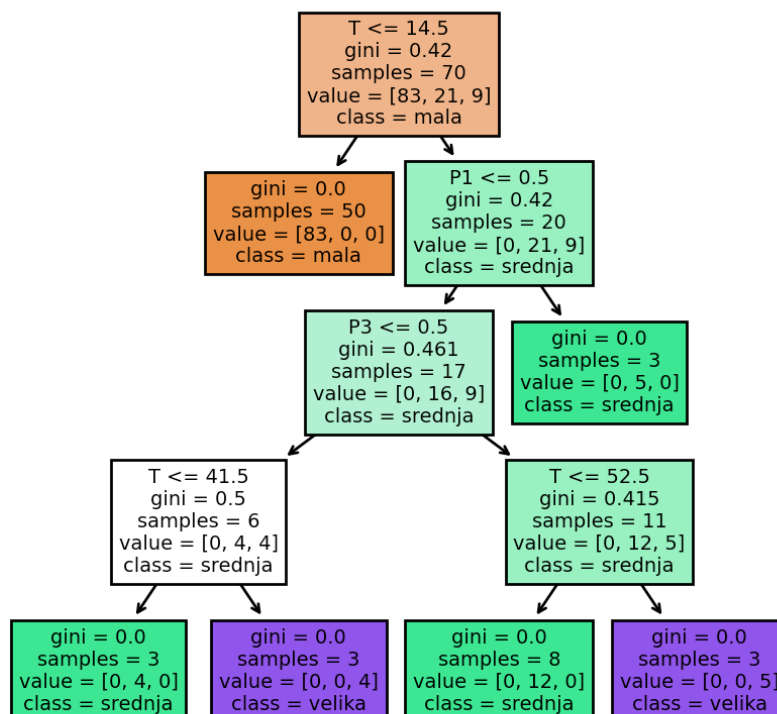
```

from sklearn import ensemble #Učitavanje RTC-a
rf_clf = ensemble.RandomForestClassifier(n_estimators=200) #Definiranje parametara
rf_clf.fit(X_train, y_train) #Podešavanje podataka za treniranje algoritma
rf_clf.score(X_test, y_test) #Podešavanje vrijednosti predviđanja

print("Postotak točnosti Random Forest:", rf_clf.score(X_test, y_test)) #Ispis rezultata
tree.plot_tree(rf_clf.estimators_[0],
               feature_names=("P1", "P2", "P3", "T"),
               class_names=("mala", "srednja", "velika"),
               filled = True) #Vizualizacija algoritma

```

Slika 39. Učitavanje modela slučajnih šuma



Slika 40. Prikaz rada modela RTF

Algoritam k najbližih susjeda (eng. k-Nearest Neighbors – k-NN) pohranjuje uzorke iz skupa na kojem se uči umjesto izgradnje modela. Klasifikacija se radi na principu glasanja gdje se za svaku značajku dodjeljuje klasa koja ima najviše glasova od k najbližih susjeda. Veći broj susjeda smanjuje šum, ali potencijalno smanjuje izraženost granice klasifikacije. Programski kod za algoritam k-NN s 3 susjeda prikazan je na slici Slika 41.

```
from sklearn.neighbors import KNeighborsClassifier #Učitavanje k-NN klasifikatora
knn_clf = KNeighborsClassifier(n_neighbors=3, weights='distance') #Definiranje parametara
knn_clf.fit(X_train, y_train) #Podešavanje vrijednosti za treniranje
knn_clf.score(X_test, y_test) #Definiranje vrijednosti za predviđanje

print("Postotak točnosti k-NN:", knn_clf.score(X_test, y_test)) #Ispis rezultata
```

Slika 41. Učitavanje modela k-NN

Prikaz točnosti ovih algoritama u odnosu na stablo odlučivanja prikazuje Slika 42.

```
runfile('/Users/karloPuzak/PycharmProjects/pythonProject1/main.py', wdir='/Users/karloPuzak/PycharmProjects/pythonProject1')
Backend MacOSX is interactive backend. Turning interactive mode on.
Postotak točnosti Decision Tree: 0.9795918367346939
Postotak točnosti Random Forest: 1.0
Postotak točnosti k-NN: 0.9795918367346939
```

Slika 42. Usporedba točnosti modela

Točnost je svugdje oko 100 % što je i očekivano s obzirom na podatke koji su bili dostupni, Moć ovakvih algoritama je ta što im nisu dane nikakve poveznice između varijabli koje se predviđaju, ali je bilo potrebno definirati značajke što kod dubokog učenja nema.

8.2 Deep Learning pristup u detekciji slike

Za pokretanje modela korišten je *open-source* programski alat PyCharm (integrirana razvojna okolina) verzije 2020.2.3., a jezik pisanja koda je Python verzija 3.6.

OpenCV (verzija 4.4.0.42) je javno dostupna biblioteka koja ima set predefiniраниh funkcija dostupnih za korištenje u realnom vremenu za rješavanje problema računalnog vida.

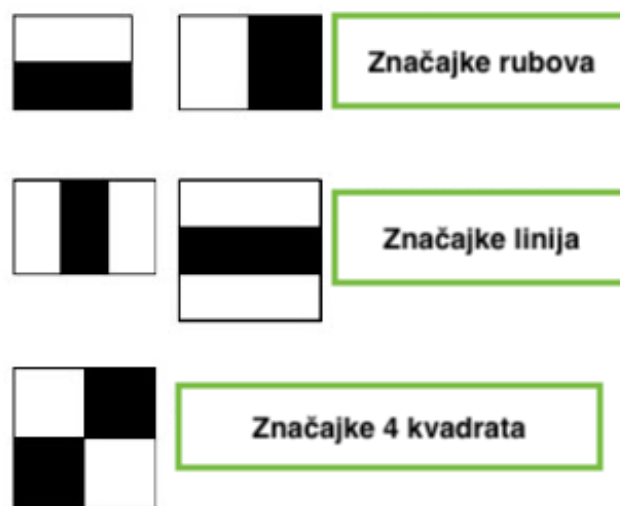
Napravljen je jednostavni detektor lica koristeći javno dostupni klasifikator predložen od strane Paula Viole i Michaela Jonesa 2001. godine. Haar Cascade je klasifikator koji ima četiri koraka tijekom izvršavanja [34]:

1. odabir značajki
2. kreiranje integralnih slika

3. adaboost treniranje
4. kaskadni klasifikatori.

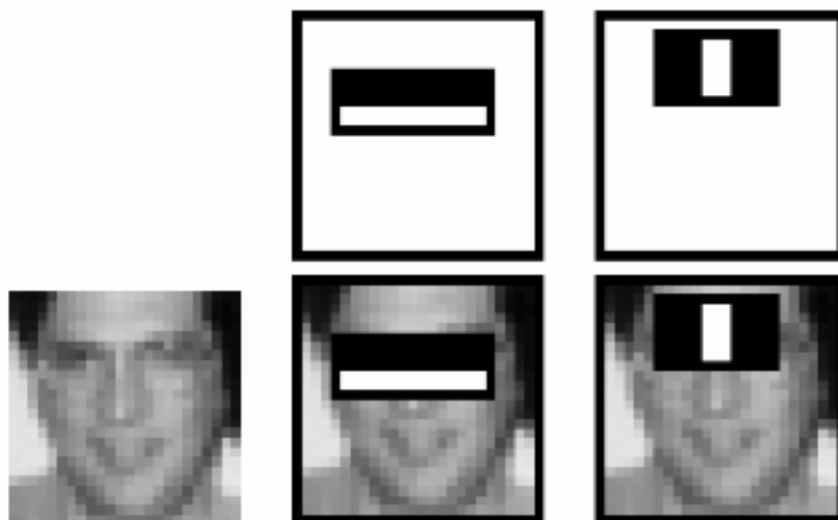
Najčešće se koristi za prepoznavanje lica i dijelova tijela, no može se koristiti za prepoznavanje bilo kojeg objekta ako postoji dovoljno veliki skup slika objekta koji se želi prepoznati.

Odabir značajki se radi na skupu „pozitivnih“ i „negativnih“ slika gdje pozitivne slike predstavljaju objekt koji se želi prepoznati, a negativne slike su one u kojima se objekt ne nalazi (ne smije se čak niti rubno nazirati na slici). Haar odabir značajki uzima u obzir susjedna pravokutna područja na određenom mjestu u detekcijskom prozoru, sumira intenzitet piksela svake regije i računa razliku sume između bijelih i crnih piksela [34]. Prikaz značajki dan je na slici Slika 43 [34].



Slika 43. Određivanje značajki

Kada su poznate sve lokacije potrebno je izračunati veliki broj značajki (regija od 24x24 piksela rezultira sa 160 000 značajki), a za lakše računanje koriste se integralne slike. Integralne slike omogućuju računanje značajki sa samo 4 piksela neovisno o veličini slike. Na slici Slika 44. [34] prva značajka je odabrana zato što je regija gdje se nalaze oči često tamnija od regije gdje se nalaze obrazi ili nos. Druga značajka je odabrana zato što je regija očiju tamnija nego regija mosta nosa [34].



Slika 44. Integralne slike

Kako bi se odabrale optimalne značajke koristi se Adaboost treniranje. U tu svrhu svaku značajku primjenjujemo na svim slikama odabranim za treniranje mreže. Za svaku značajku se nalazi najbolji prag koji će klasificirati pozitivne i negativne slike. Nijedan algoritam nije savršen, pa tako i ovaj može pogriješiti. Svaka slika dobiva istu vrijednost težine na početku procesa, a nakon svake klasifikacije povećavaju se ponderi pogrešno klasificiranih slika. Postupak se zatim ponavlja i izračunavaju se nove stope pogrešaka i težina dok se ne dosegne zadovoljavajuća razina točnosti ili dok se ne pronađe potreban broj značajki. Konačni klasifikator je nusprodukt težinskih suma svih lošijih klasifikatora. Uzme li se u obzir da je obično veličina slike koja se obrađuje zapravo šum (pozadina) potrebno je brzo provjeriti je li regija koja se provjerava pozadina ili dio lica. Ako nije dio lica – odbacuje se. Time je omogućeno brže izračunavanje i radi toga je uveden koncept kaskadnog klasifikatora. Kaskadni klasifikator neće provjeravati svih N značajki na određenoj regiji već se one grupiraju po različitim fazama klasifikacije. Ako je regija pozitivna na nju se primjenjuje druga faza provjere značajki, a u suprotnom se odbacuje. Viola-Jones algoritam ima preko 6000 značajki podijeljenih u 38 faza gdje je 1, 10, 25, 25 i 50. značajki raspoređeno kroz prvih 5 faza, a za 200 pronađenih značajki dobiva se točnost od 95 % [34].

Pokretanje kamere kroz programsko sučelje PyCharm radi se kroz naredbu *VideoCapture* koju je potrebno učitati kroz biblioteku *cv2*. Klasifikator se učitava koristeći *CascadeClassifier* funkciju u kojoj se definira putanja u kojoj se nalazi datoteka klasifikatora kako je prikazano na slici Slika 45. Korišten je klasifikator za prepoznavanje prednje strane lica.

```
import cv2 #Učitavanje modula za računalni vid
cap = cv2.VideoCapture(0) #Kreiranje naredbe za pokretanje video kamere računala
cascade_classifier = cv2.CascadeClassifier(
    './Users/karlopuzak/PycharmProjects/pythonProject1/venv/lib/python3.6/site-packages/cv2/data/haarcascade_frontalface_alt.xml'
) #Učitavanje open-source klasifikatora
```

Slika 45. Naredba za pokretanje kamere

Dodan je prikaz kvadrata koji prati detektirano lice i tekst koji prikazuje o kojoj se osobi radi. Također su definirane dodatne naredbe za gašenje prozora kamera i parametri povezani uz kvadrat za detekciju i tekst kako je prikazano na slici Slika 46.

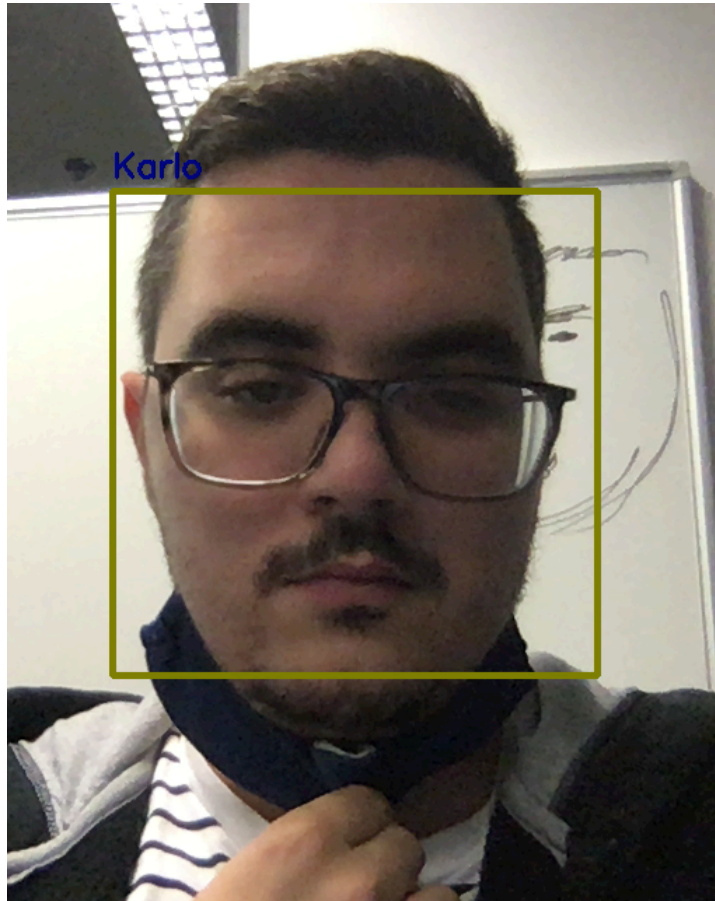
```
while True: #Uvjet
    ret, frame = cap.read()
    frame = cv2.cvtColor(frame, 0)
    detections = cascade_classifier.detectMultiScale(frame) #Korištenje prepoznavanja više lica na videu

    if (len(detections) > 0): #Uvjet
        (x, y, w, h) = detections[0]
        frame = cv2.rectangle(frame, (x,y), (x+w, y+h), (0, 128, 128), 3) #Definiranje parametara boxa klasifikatora
        cv2.putText(frame, "KarLo", (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (128, 0, 0), 2) #Definiranje teksta boxa
        cv2.imshow('frame', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'): #Pristikom na tipku q izlaz iz sučelja
            break
    cap.release()
    cv2.destroyAllWindows()
```

Slika 46. Definiranje kvadrata lokalizacije lica

Potrebno je naglasiti da je ovo detektor, a ne i klasifikator! Ovaj detektor je već učen na javno dostupnim slikama lica, ali ga je moguće naučiti i da detektira i klasificira objekt ili lice po želji. Jednostavna implementacija omogućava brze rezultate, ali je potreban veliki dataset na kojem će se klasifikator učiti kako bi dao optimalne rezultate.

Slika 47. i Slika 48. prikazuju detekcije lica, dok Slika 49. prikazuje lažno pozitivnu detekciju što pokazuje da algoritam nije savršen i potrebno ga je dodatno optimizirati korištenjem većeg, definiranog, pročišćenog skupa podataka kako se ne bi javljali lažno pozitivni ili negativni rezultati. Sve slike snimljene su integriranom kamerom prijenosnog računala rezolucije 720 piksela koja daje poprilično šumovitu sliku.



Slika 47. Detekcija lica

Tijekom testiranja korištena je i zaštitna maska kako bi se dobio bolji dojam koji problemi postoje u detekciji i klasifikaciji slike lica. Maska je prekrila gotovo pa 50 % lica čime je onemogućena ispravna detekcija. Ovo otvara i mogućnost učenja klasifikatora da predviđa koristi li osoba masku ili npr. zaštitnu opremu tijekom rada u nekom postrojenju (vizir tijekom zavarivanja ili rukavice kod montaže osjetljivih dijelova).



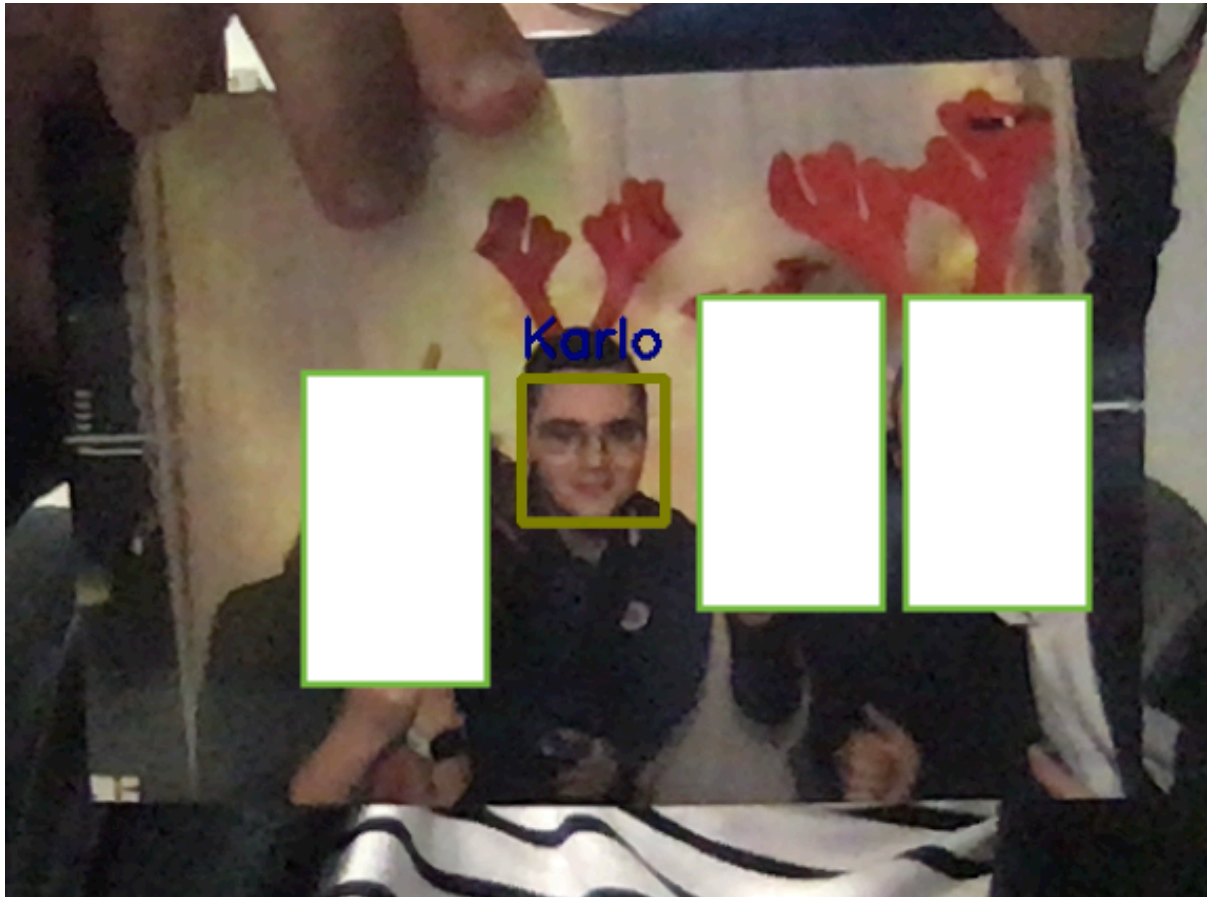
Slika 48. Nema detekcije lica

Prikaz lažno pozitivnog rezultata dan je na slici X.



Slika 49. Lažno pozitivna slika

Osim na primjeru u stvarnom vremenu iskorištena je i isprintana slika gdje je klasifikator prepoznao lice osobe u relativno lošim uvjetima i s nešto lošijom kamerom prijenosnog računala, a ostala lica su radi zaštite podataka anonimizirane bijelim kvadratima.



Slika 50. Detekcija lica na isprintanoj slici

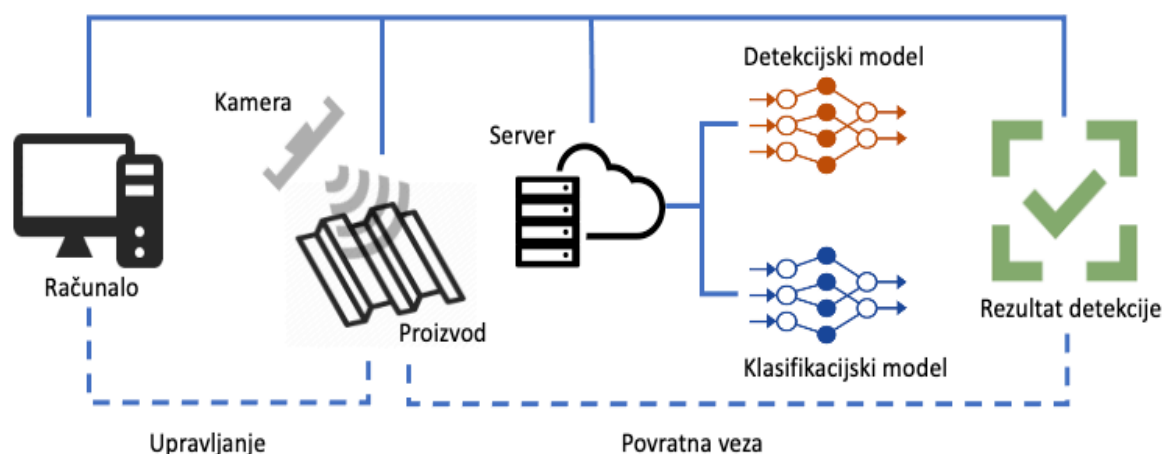
Ovo otvara mogućnosti za korištenjem dubokog učenja u kontroli kvalitete za provjeru oštećenja proizvoda u proizvodnji, analizu pokreta radnika tijekom montaže dijelova, praćenje koristi li se zaštitna oprema ili čak za praćenje ulaska i izlaska ljudi iz poduzeća. Naravno duboko učenje se može koristiti i za predviđanje varijabli u optimizaciji financija i troškova.

8.2.1 Prikaz algoritma na realnim podacima

Površinska oštećenja mogu imati značajan utjecaj na kvalitetu proizvoda i zadovoljstvo kupaca. Iz tog je razloga bitno osigurati kvalitetnu i brzu kontrolu kvalitete. Ljudska kontrola podložna je greškama uzrokovanim umorom, repetitivnim ponavljanjem istih radnji, subjektivnosti u kontroli i padom točnosti kontrole protekom vremena. Umjetna inteligencija nema te probleme i zato je pogodna za korištenje u industrijama kod proizvodnje LCD monitora, metalnih limova i ploča, tkanina i kod vizualne analize komponenti [35].

Metalne površine imaju neujednačenu iluminaciju i odsjaj što stvara problem u njihovoj analizi. Također i greške koje se javljaju na površini nisu lokalizirane ni dimenzijski velike, a ovise o uvjetima u kojima se proizvod nalazi (prašina i vlakna se mogu neispravno klasificirati kao oštećenje). Konvolucijske neuronske mreže kao ulaz dobivaju regiju na kojoj se oštećenje nalazi, a kao izlaz daju klasifikaciju odnosno vrstu oštećenja koja se nalazi na površini materijala. Problem analize metalnih površina je zato dvojak problem – određivanje lokacije oštećenja (lokalizacija/detekcija) i određivanje vrste oštećenja (klasifikacija). Bitna je značajka da ulazna slika u mrežu mora biti crnobijela kako bi se izbjegle različitosti u procesuiranju slika u boji (aluminij, bakar ili željezo s raznim oštećenjima mogu davati drugačije odzive u mreži) [35].

GC10-DET set podataka prikazuje različite vrste klasifikacije realnih problema dobivenih u raznim industrijskim postrojenjima. Podaci se prikupljaju na način kako je prikazano na shemi [35]:



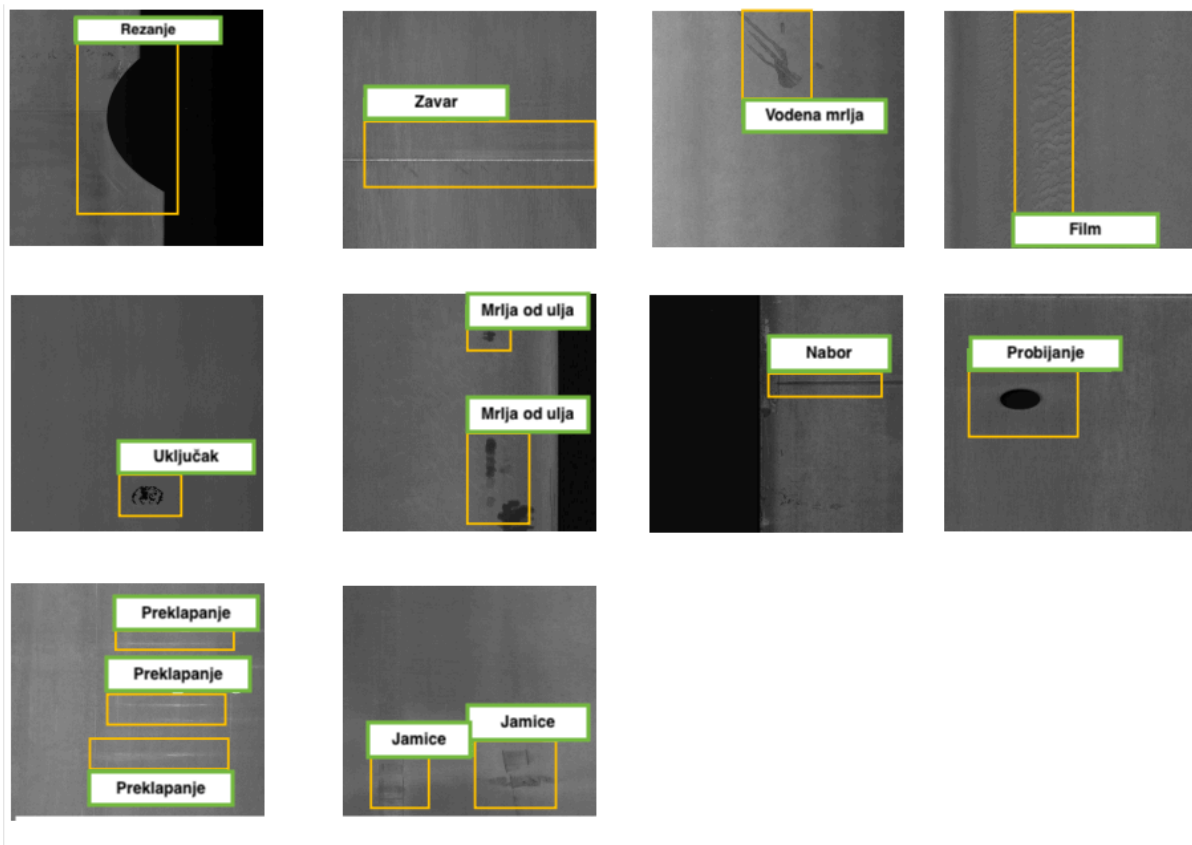
Slika 51. Prikupljanje podataka u postrojenju

Računalo upravlja cijelim procesom kontrole kvalitete, kamerom se snima slika proizvoda i šalje se na obradu na server koji može biti fizički postavljen u postrojenju ili u oblaku. Nad slikama se izvode klasifikacijski i detekcijski modeli koji za izlaz preko povratne veze daju rezultate detekcije i klasifikacije greške u proizvodu. S obzirom na to da postoji povratna veza, nove slike se mogu koristiti i spremiti u bazu podataka za dodatno treniranje i učenje mreže. Jedan od problema koji se javlja je snimanje slike u realnim uvjetima tijekom proizvodnje i za takve je slučajeve potrebno koristiti kameru visoke učinkovitosti snimanja slika, a ovisno o veličini proizvoda mogu se koristiti više serijski spojenih kamera koje kreiraju potpunu sliku. Ako se radi o materijalima s nejednakim i velikim odsjajem potreban je i dodatan filter/stabilizator slike kako okolišni utjecaji ne bi imali neželjene posljedice na sliku i u konačnici davali krive rezultate detekcije ili klasifikacije [35].

GC10-DET dataset je dostupan na Github [36] stranici i uključuje 3570 crnobijelih slika s klasificiranim oštećenjima [35]:

1. probijanje
2. oštećenja od zavarivanja
3. oštećenja od rezanja
4. mrlje od ulja – nastaju špricanjem lubrikanta kod obrade proizvoda
5. film – površinsko oštećenje neujednačene gustoće koje se javlja na rubovima proizvoda radi nejednake temperature ili tlaka tijekom obrade
6. uključci
7. jamičasta površina
8. nabor
9. preklapanja.

Vizualni prikaz oštećenja prikazuje Slika 52 [35].



Slika 52. Prikaz mogućih oštećenja površine

Informacije o skupu podataka prikazuje Slika 54. gdje je vidljivo da su slike podijeljene u 10 foldera od kojih svaki ima 146 slika što je ukupno 2306 slika.

```
import os
import glob
path = '/Users/karlopuzak/Downloads/archive/images/images' #učitavanje putanje foldera
dirlist = os.listdir(path)
print("Folderi sa klasificiranim slikama:", dirlist) #ispis svih foldera u kojima se nalaze slike

list = os.listdir('/Users/karlopuzak/Downloads/archive/images/images/waist folding') #učitavanje putanje foldera
number_files = len(list) #Definiranje varijable za zbrajanje
print("Slika u folderu za Preklapanja ima:", number_files) #Ispis

list2 = os.listdir('/Users/karlopuzak/Downloads/archive/images/images/inclusion')
number_files2 = len(list)
print("Slika u folderu za Uključke ima:", number_files2)
```

Slika 53. Učitavanje podataka

```
Folderi sa klasificiranim slikama: ['waist folding', '.DS_Store', 'inclusion', 'water_spot', 'crescent_gap', 'welding_line', 'crease', 'silk_spot', 'oil_spot', 'punching_hole', 'rolled_pit']
Slika u folderu za Preklapanja ima: 146
Slika u folderu za Uključke ima: 146
Slika u folderu za Vodu: 146
Slika u folderu za Rezanje ima: 146
Slika u folderu za Zavarivanje ima: 146
Slika u folderu za Nabor ima: 146
Slika u folderu za Film ima: 146
Slika u folderu za Ulje ima: 146
Slika u folderu za Probijanje ima: 146
Slika u folderu za Valjanje ima: 146
```

Slika 54. Struktura podataka

Nakon analize i inicijalne obrade podataka dobivena je jednoznačna struktura tako da svaka slika ima svoje pripadajuće anotacije s definiranim nazivima.

```

Broj slika za train: 2280
Broj anotiranih slika: 2280
      Types
0      crescent_gap
1      crescent_gap
2      crease
3      silk_spot
4      silk_spot
...      ...
2275   silk_spot
2276   silk_spot
2277   silk_spot
2278   silk_spot
2279   crescent_gap
[2280 rows x 1 columns]

```

Slika 55. Nazivlje podataka

Model se sastoji od 4 sloja, aktivacijske funkcije su ReLu i softmax, a broj epoha (iteracija potrebnih da se obradi set podataka – *batch size*) je 20. Optimizacijska funkcija je Adam, a funkcija gubitka izračunava se preko funkcije *Categorical_crossentropy* [37].

```

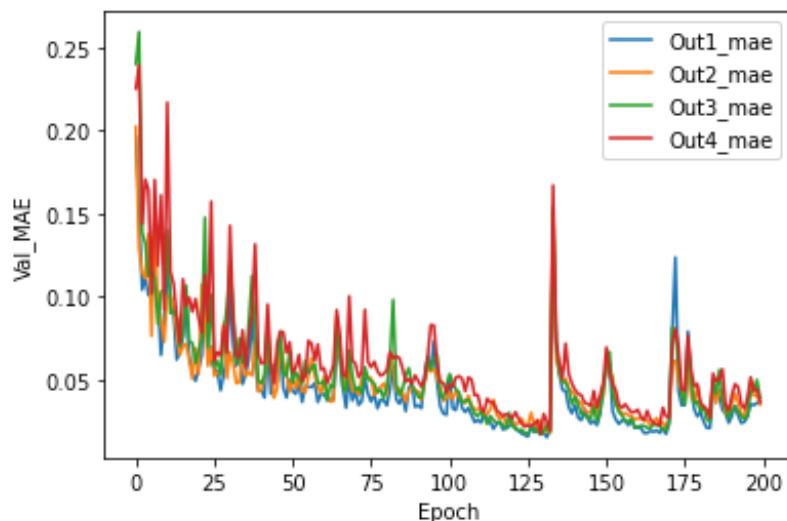
model.compile(keras.optimizers.Adam(0.0003),
              loss={'out1':'mse',
                   'out2':'mse',
                   'out3':'mse',
                   'out4':'mse',
                   'out_item':'categorical_crossentropy'},
              metrics=['mae','acc'])

lr_reduce = keras.callbacks.ReduceLROnPlateau('val_loss', patience=6, factor=0.5, min_lr=1e-6)
history = model.fit(train_dataset,
                    steps_per_epoch=train_count//BATCH_SIZE,
                    epochs=200,
                    validation_data=test_dataset,
                    validation_steps=test_count//BATCH_SIZE)

```

Slika 56. Definiranje modela klasifikatora

Veći broj ukazuje na veću pogrešku, a manji broj na bolji rezultat. Da je rezultat MSE jednak nuli, onda bi model bio savršen, no MSE dosta ovisi o izabranom modelu i nije jednoznačna mjera točnosti modela. Slika 57. i Slika 58. prikazuju u Pythonu, u obliku dijagrama i numerički, po epohama za MSE točnost modela. MSE postotak je oko 0,038 % dok je točnost klasifikacije 0,991 % [37].

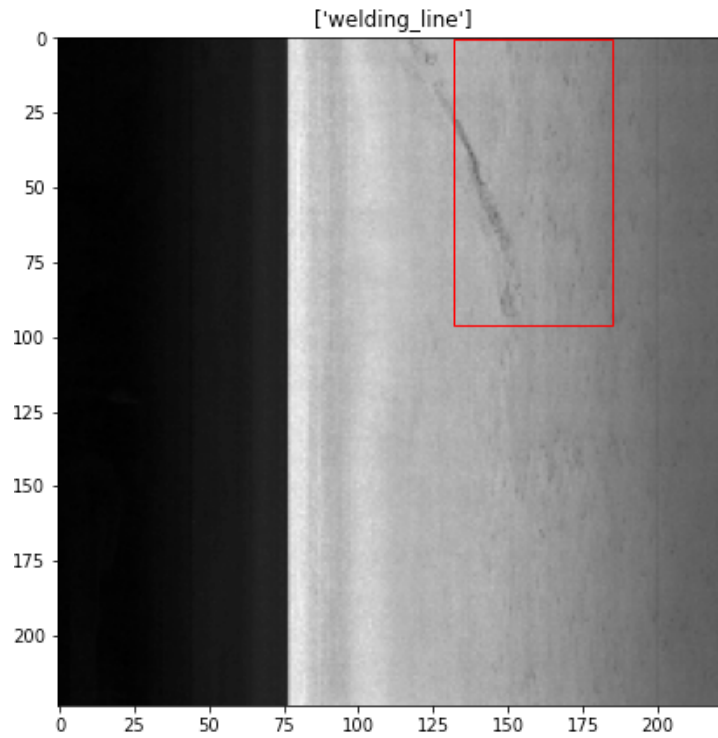


Slika 57. Grafički prikaz MSE

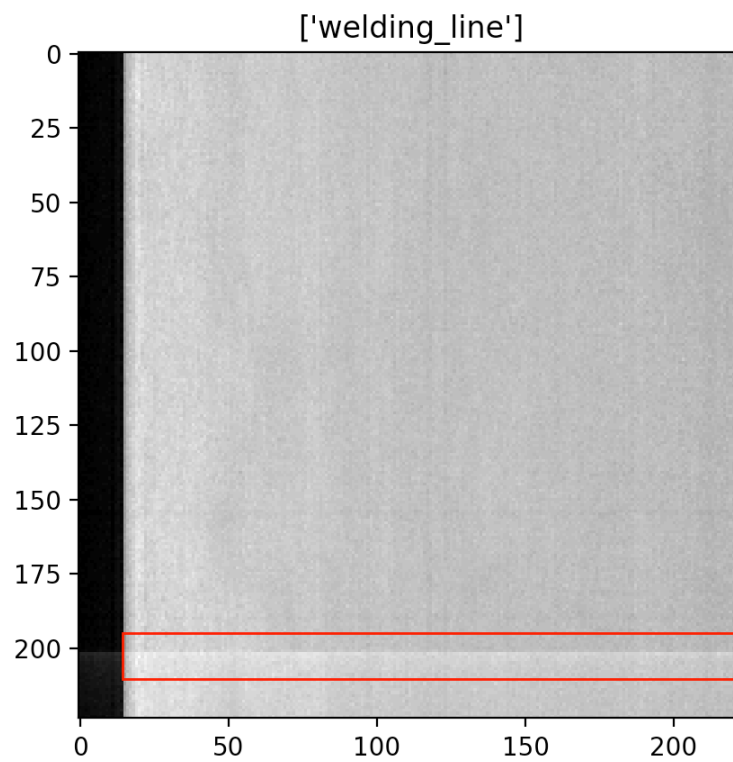
```
out1_mae in test:0.036932073533535004
out2_mae in test:0.03723800554871559
out3_mae in test:0.03834963217377663
out4_mae in test:0.041412170976400375
class_label in test:0.9980811476707458
```

Slika 58. Numerički prikaz točnosti

Na slici Slika 59. i Slika 60. je prikazana jedna nasumično odabrana slika gdje je u gornjem dijelu prikazan rezultat klasifikacije – *welding_line* što je oštećenje od zavarivanja. Vidljiv je i pravokutnik lokalizacije oštećenja koji, iako uz izuzetnu točnost klasifikacije, nije ispravno postavljen oko oštećenja već je obuhvaćen veći dio materijala. Točnost same lokalizacije je očekivano niska radi samih podataka – siva slika s tamnim oštećenjima na materijalu koji nema savršeno ujednačenu boju pozadine.



Slika 59. Prva nasumično odabrana slika za klasifikaciju



Slika 60. Druga nasumično odabrana slika za klasifikaciju

9 Zaključak

Razvojem tehnologije strojarstvo kao disciplina koja objedinjuje konstrukcije, energetiku, menadžment, inženjerstvo materijala, logistiku itd. mijenja se sve više iz klasičnog u interdisciplinarno područje koje uključuje i implementaciju računalnih programa i algoritama. Tvornice i skladišta postaju pametne tvornice i skladišta 4.0 čime se povećava i razvoj automatiziranosti i agilnosti cijele industrije – Industrija 4.0. Senzori, big data, računarstvo u oblaku i ostali koncepti objašnjeni u radu mijenjaju način na to kako tvrtke posluju, ali i daju velike mogućnosti za napretkom, digitalizacijom i otvaranjem novih tržišta po vertikali i horizontali. Razvoj računalnih algoritama pomaže u optimizaciji poslovnih procesa tako da više čovjek nije taj koji radi kompleksne numeričke kalkulacije već to radi računalo koje mu služi kao potpora u donošenju odluka i smanjenju fizičkog rada što je i vidljivo u tvrtki Gideon Brothers gdje postoji sinergija između robota u skladištu i radnika. Korištenjem dubokog učenja dobiva se uvid u nove poslovne mogućnosti korištenjem velikih količina podataka koji su preduvjet za svaki rad u tom području. U radu je prikazana jednostavan detektor lica osobe koji se može prenamijeniti za detekciju bilo kojeg objekta (vijaka i malih dijelova, oštećenja površine itd.) u kontroli kvalitete. Kako bi tvrtka krenula u implementaciju novih tehnologija potrebno je ispuniti nekoliko uvjeta – osigurati kvalificiranu radnu snagu, alate i podatke, ali prije svega napraviti studiju isplativosti ulaganja u nove tehnologije.

10 Literatura

- [1] F. Lima, C. N. Carvalho, M. B. S. Acardi, E. G. Santos, G. B. Miranda, R. F. Maia i A. A. Massote, »Digital Manufacturing Tools in the Simulation of Collaborative Robots: Towards Industry 4.0,« *Brazilian Journal of Operations & Production Management*, svez. 16., br. 2., pp. 261-280, 2019..
- [2] Y. Zhang, Y. Cheng, X. V. Wang, R. Y. Zhong, Y. Zhang i T. Fei, »Data-driven smart production line and its common factors,« *The International Journal of Advanced Manufacturing Technology*, br. 103, p. 1211–1223, 2019..
- [3] »Cloud Computing,« [Mrežno]. Available: https://en.wikipedia.org/wiki/Cloud_computing#Characteristics. [Pokušaj pristupa 05. 11. 2020.].
- [4] K. Bates, »Seeva group,« 02. 07. 2018.. [Mrežno]. Available: <https://sevaa.com/blog/2018/07/cloud-computing/>. [Pokušaj pristupa 05. 11. 2020.].
- [5] M. Krsnik, »Digitalizacija proizvodnje i proizvodnih procesa,« 01. 2019.. [Mrežno]. Available: http://repositorij.fsb.hr/9094/1/Krsnik_2019_zavrsnirad.pdf. [Pokušaj pristupa 20. 10. 2020.].
- [6] N. Štefanić, »Operacijska istraživanja I,« FSB, Zagreb, 2019..
- [7] N. Štefanić, »Operacijska istraživanja II,« FSB, Zagreb, 2019..
- [8] S. Lojen, »Diplomski rad: Primjena heurističkih metoda u mjeriteljstvu,« 2016.. [Mrežno]. Available: http://repositorij.fsb.hr/7117/1/Lojen_2016_Diplomski.pdf. [Pokušaj pristupa 05. 02. 2020.].
- [9] G. Radanović, »Algoritmi bazirani na populaciji rješenja,« FER, [Mrežno]. Available: http://www.zemris.fer.hr/~golub/ga/studenti/seminari/2007_radanovic/populacija.html. [Pokušaj pristupa 15. 02. 2020.].
- [10] M. Golub, »Genetski algoritmi, Prvi dio,« 27. 09. 2004.. [Mrežno]. Available: http://www.zemris.fer.hr/~golub/ga/ga_skripta1.pdf. [Pokušaj pristupa 15. 02. 2020.].
- [11] I. Rahwan, »Moral Machine,« [Mrežno]. Available: <http://moralmachine.mit.edu/>. [Pokušaj pristupa 15. 02. 2020.].
- [12] B. M. Novakovic, »Umjetna Inteligencija i robotika,« 2005.
- [13] N. Štefanić, »Proizvodni menadžment,« FSB, Zagreb, 2018..
- [14] B. D. Bašić, »Strojno učenje - uvod,« FER, Zagreb, 2009.
- [15] N. Štefanić, »Industrija 4.0 i umjetna inteligencija,« FSB, Zagreb, 2019.
- [16] S. Šegvić, »Duboko učenje,« FER, Zagreb, 2014.
- [17] A. Kovač, »Primjena umjetne inteligencije: prednosti i opasnosti,« FFZG, Zagreb, 2019.
- [18] B. Jerbić, »Projekt RONNA (Robotska neuronavigacija),« 2020.. [Mrežno]. Available: <https://100.fsb.hr/hr/119/FSB+robot+RONNA+u+neurokirurgiji+u+Zagrebu>. [Pokušaj pristupa 20. Ožujak 2020.].

- [19] »Roboti u službi čovjeka,« Hrvatska Pošta, 20. 12. 2019.. [Mrežno]. Available: <https://www.posta.hr/hrvatska-posta-pokrenula-pilot-projekt-s-gideon-brothersom/8138>. [Pokušaj pristupa 01. 04. 2020.].
- [20] Google Cloud, »Mitsubishi Case Study,« [Mrežno]. Available: <https://cloud.google.com/customers/mitsubishi-motors/>. [Pokušaj pristupa 07. 11. 2020.].
- [21] Mitsubishi Motors, »2020 Mitsubishi Eclipse Cross Technology,« [Mrežno]. Available: <https://www.mitsubishicars.com/eclipse-cross/2020/features/technology>. [Pokušaj pristupa 07. 11. 2020.].
- [22] Google Cloud, »Sandvik Case Study,« [Mrežno]. Available: <https://cloud.google.com/customers/sandvik>. [Pokušaj pristupa 07. 11. 2020.].
- [23] Prism, »CNC Programming Software for Simple Parts - Prism,« [Mrežno]. Available: <https://www.makewithprism.com/>. [Pokušaj pristupa 07. 11. 2020.].
- [24] K. Budek, »Spot the flaw - visual quality control in manufacturing,« 19. 04. 2018.. [Mrežno]. Available: <https://deepsense.ai/spot-the-flaw-visual-quality-control-in-manufacturing/>. [Pokušaj pristupa 07. 11. 2020.].
- [25] Unleashed Software, »Understanding Manufacturing Productivity Guide,« [Mrežno]. Available: <https://www.unleashedsoftware.com/manufacturing-productivity-guide>. [Pokušaj pristupa 01. 11. 2020.].
- [26] Unleashed Software, »How to improve productivity in manufacturing,« [Mrežno]. Available: <https://www.unleashedsoftware.com/manufacturing-productivity-guide/how-to-improve-productivity-in-manufacturing>. [Pokušaj pristupa 01. 11. 2020.].
- [27] D. Partha, »Quality inspection in manufacturing using deep learning based computer vision,« 18. 12. 2018.. [Mrežno]. Available: <https://towardsdatascience.com/quality-inspection-in-manufacturing-using-deep-learning-based-computer-vision-daa3f8f74f45>. [Pokušaj pristupa 25. 10. 2020.].
- [28] D. Smolčić, »Raspoznavanje objekata konvolucijskim neuronskim mrežama,« 06. 2015. [Mrežno]. Available: <http://www.zemris.fer.hr/~ssegvic/multiclod/students/smolcic15bs.pdf>. [Pokušaj pristupa 27. 10. 2020.].
- [29] J. Wang, Y. Ma, L. Zhang, R. X. Gao i D. Wu, »Deep learning for smart manufacturing: Methods and applications,« *Journal of Manufacturing Systems*, br. 48: 144-156, 10.1016/j.jmsy.2018.01.003, 2018..
- [30] F. Arellano Espitia i L. Ruiz Soto, »Novel Methods Based on Deep Learning Applied to Condition Monitoring in Smart Manufacturing Processes,« *IntechOpen*, 10.5772/intechopen.89570.
- [31] L. Pleše, »Duboko učenje [seminar],« 05. 2019.. [Mrežno]. Available: <http://www.zemris.fer.hr/~ssegvic/project/pubs/plese19sem1.pdf>. [Pokušaj pristupa 15. 10. 2020.].

- [32] A. Navlani, »Decision Tree Classifier in Python,« 28. 12. 2018.. [Mrežno]. Available: <https://www.datacamp.com/community/tutorials/decision-tree-classification-python>.
[Pokušaj pristupa 10. 11. 2020.]
- [33] M. Gudlih, M. Hegedić, H. Cajner i N. Štefanić, »SOLVING PROBLEMS OF INTERRUPTIONS AND MULTITASKING IN THE PHARMACY OF A LARGE HOSPITAL CENTRE,« Fakultet strojarstva i Brodogradnje, Zagreb, 2020..
- [34] »OpenCV: Cascade Classifier,« [Mrežno]. Available: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html.
[Pokušaj pristupa 10. 11. 2020.]
- [35] X. Lv, F. Duan, J.-J. Jiang, X. Fu i L. Gan, »Deep Metallic Surface Defect Detection: The New Benchmark and Detection Network,« The State Key Lab of Precision Measuring Technology and Instruments, Tianjin University.
- [36] X. Lv, »GC10-DET dataset,« 02. 2020.. [Mrežno]. Available: <https://github.com/lvxiaoming2019/>.
[Pokušaj pristupa 15. 11. 2020.]
- [37] Via, »Class Location,« 06. 2020.. [Mrežno]. Available: <https://www.kaggle.com/zhangyunsheng/class-location>.
[Pokušaj pristupa 15. 11. 2020.]

11 PRILOZI

I. CD-R disk