

SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
VARAŽDIN

Petar Matišić

**RAČUNALNI VID I PREPOZNAVANJE  
OBJEKATA POMOĆU BIBLIOTEKE  
OPENCV**

**ZAVRŠNI RAD**

Varaždin, 2022.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Petar Matišić

Matični broj: 0016145882

Studij: Informacijski sustavi

## RAČUNALNI VID I PREPOZNAVANJE OBJEKATA POMOĆU BIBLIOTEKE OPENCV

ZAVRŠNI RAD

Mentor:

dr. sc. Bogdan Okreša Đurić

Varaždin, rujan 2022.

*Petar Matišić*

**Izjava o izvornosti**

Izjavljujem da je ovaj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

*Autor potvrdio prihvaćanjem odredbi u sustavu FOI Radovi*

---

## Sažetak

U ovome radu obrađuje se problematika računalnog vida, odnosno područje koje je posljednjih godina uspjelo napraviti veliki skok i uspjelo nadmašiti ljudе u nekim zadacima povezanim s otkrivanjem i označavanjem objekata. Predstavljena je povijest problema te njegov opis. Zatim su predstavljeni neki od načina rješavanja ovoga problema. Osim obrade navedenog, također, cilj izučavanja u ovom radu je i njena teorijska podloga koja seže iz dubokog učenja i neuronskih mreža koje danas predstavljaju temelj za izradu raznolikih modela vezanih za strojno učenje i njegove aplikacije. Nabrojana su područja primjene računalnog vida, te je prikazan rad kroz primjere zasnovan za konceptima navedenih područja. Vezano za primjere, na kraju su napravljene dvije različite implementacije, od kojih jedna prepoznaje gestikulacije ruke, a druga gdje se također pomoću ruke može prikazati upravljanje i manipulacija pokazivačem na zaslonu.

**Ključne riječi:** računalni vid; duboko učenje; neuronska mreža; python; ruka; opencv

# Sadržaj

<b>1. Uvod</b>	1
<b>2. Duboko učenje</b>	2
2.1. Definiranje pojmovra	3
2.2. Algoritmi	5
2.3. Neuronske mreže	6
2.3.1. Konvolucijske neuronske mreže	9
<b>3. Računalni vid</b>	11
3.1. Pregled povijesti	12
3.2. Digitalna obrada slike	16
3.3. Prepoznavanje	18
3.3.1. Prepoznavanje objekata	18
3.3.2. Prepoznavanje geste ruke	20
3.4. Primjena	23
<b>4. Implementacije praktičnog dijela</b>	25
4.1. Opis i svrha	25
4.2. OpenCV	26
4.3. Analiza korištenih metoda	27
<b>5. Zaključak</b>	32
<b>Popis literature</b>	35
<b>Popis slika</b>	37
<b>Popis isječaka koda</b>	38
<b>1. Programski kôd</b>	40

# 1. Uvod

Kraj 20. stoljeća započeo je sve bržim razvojem elektronike, računala i softvera koji je omogućio bolja računala. Danas se nove tehnologije i softver razvijaju tako brzo da se inovacije pojavljuju gotovo svaki dan u *IT* svijetu. "Obični" programeri više ne mogu pratiti sve trendove i pravce razvoja programskih jezika, već fokus rada prelazi na jedno područje. Jedno od takvih područja, koje je novo, ali se rapidno razvija, je računalni vid.

Računalni vid daje računalima sposobnost "vidjeti", odnosno razumjeti što je na fotografiji ili videu. Računalni vid nam omogućuje razvoj potpuno novih aplikacija za svakodnevnu upotrebu. Njegova glavna svrha može biti npr. sigurnost i zaštita, kao što su: zaštita ljudi, zaštita podataka, sprječavanje smrti, otkriće raka itd.

Uz pomoć računalnog vida, troškovi se mogu smanjiti u usporedbi s troškovima obavljanja istih poslova pri korištenju ljudskih usluga, uz relativno nisku ekonomsku potrošnju. Daljnjim razvojem računalnog vida može se smanjiti broj pogrešaka u analizi određenih informacija u odnosu na ljudski faktor. U svakodnevnom životu, računalni vid može pomoći u rješavanju kriminalnih radnji, detektirao bi osobe koje su počinile neku vrstu nedjela ili kaznenog djela. Ljudska svijest o postojanju nekakvog sustava nadzora smanjuje učestalost kriminalnih aktivnosti. Isto tako, njegova pojava može spriječiti mnoge prometne nesreće, pomoći i razviti medicinske usluge, poput otkrivanja bolesti kod ljudi, otkrivanja tumora itd. Računalni vid svakim danom ima nove zahtjeve koje treba implementirati u svakodnevne aktivnosti, a jedan od njih je prepoznavanje kupaca i njihovih dosadašnjih kupovnih navika i potreba. Danas, dakle, postoje automobili koji se upravljaju na vlastiti pogon, mobiteli koji se mogu otključati samo uz pomoć algoritama za prepoznavanje lica i mnoge druge inovacije u svakodnevnom životu, zahvaljujući pojedincima koji su prepoznali mogućnosti koje nudi računalni vid. Još uvjek nismo svjesni činjenice koliko je ovaj segment snažan i primjenjiv za razvoj te kolika će biti njegova korisnost za buduće sinergije u suradnji računala/stroja i ljudi (npr. zajednički rad u proizvodnji, vožnji i sl.).

Rad bi trebao objasniti kako računalo može prepoznati postoji li određena vrsta objekta na slici, poput prometnog znaka, ljudskog lica, tumora na medicinskoj slici i još mnogo toga. Računala mogu koristiti širi raspon valnih duljina svjetlosti za računalni vid za razliku od ljudskog oka. Znanje računalnog vida primjenjivo je u svim znanstvenim disciplinama računarstva i mnogo drugih znanosti.

## 2. Duboko učenje



Slika 1: Neuronska mreža i polutka mozga; prema: [1]

*“Predviđanje budućnosti nije magija, to je umjetna inteligencija.” – Dave Waters [1].*

## 2.1. Definiranje pojma

Tijekom proteklih nekoliko godina duboko učenje postalo je popularan pojam u svijetu tehnologije. Prije nego što se detaljnije objasni duboko učenje, njegovi algoritmi i primjene, prva stvar koja je najbitnija u razumijevanju dubokog učenja je što je to stvarno duboko učenje.

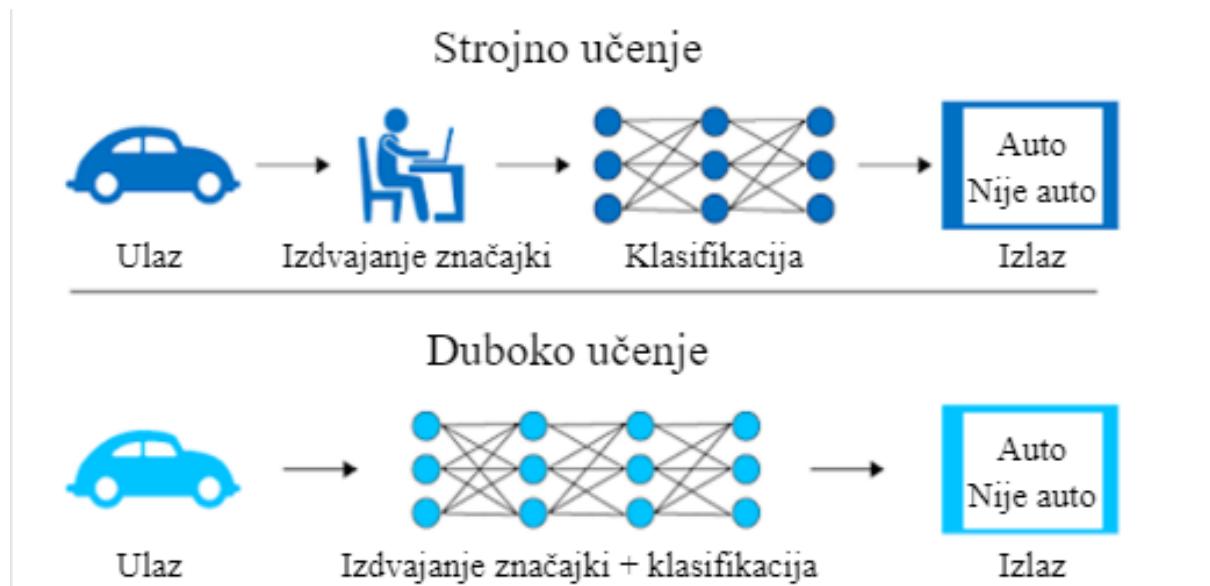
Duboko učenje [2] je potpodručje strojnog učenja koje se bavi algoritmima inspiriranim strukturom i funkcijom mozga. Duboko učenje je automatski i podskup umjetne inteligencije jer samim time strojno učenje spada pod područje umjetne inteligencije.



Slika 2: Potpodručja umjetne inteligencije; prema: [2]

Glavni cilj [2] umjetne inteligencije je osigurati skup algoritama i tehnika koje se mogu koristiti za rješavanje problema koji su intuitivni i samoizvršivi za ljudi, ali predstavljaju veliki izazov za računala. Dobar primjer ove vrste problema je tumačenje i razumijevanje sadržaja slike, nešto što ljudi mogu učiniti bez napora, a što je strojevima vrlo teško, kao što je spomenuto u uvodu. Razlika između dubokog učenja, strojnog učenja i umjetne inteligencije je prikazana na slici 2.

Kada je riječ o pojmu dubokog učenja, obično se misli na neuronsku mrežu koja većinom sadrži velik broj slojeva [2]. Dakle, duboko učenje zapravo je vrlo širok pojam jer ne postoji univerzalno prihvaćen broj slojeva koje neuronska mreža mora imati da bi se klasificirala kao duboka neuronska mreža. Za razliku od tradicionalnog strojnog učenja koje uzima skup slika i primjenjuje ručno definirani algoritam izdvajanja značajki te, ili čak ručno izdvajanje značajki, provodi obuku klasifikatora koji se temelje na tim značajkama (Slika 3), duboko učenje temelji se na principu slaganja sloj-po-sloj, koje automatski uče složenije i apstraktnije značajke u usporedbi s tradicionalnim strojnim učenjem.



Slika 3: Razlika dubokog i strojnog učenja; prema: [2]

Za većinu problema, neki klasični algoritmi strojnog učenja proizvest će dovoljno dobre modele [3]. Za druge, ozbiljnije probleme, klasični algoritmi strojnog učenja u prošlosti nisu imali dovoljno dobre rezultate. Jedno od područja gdje se duboko učenje često naglašava je obrada prirodnog jezika, uključujući prijevod jezika, automatsko sažimanje, analitiku, morfološku segmentaciju, analizu osjećaja i prepoznavanje govora [4].

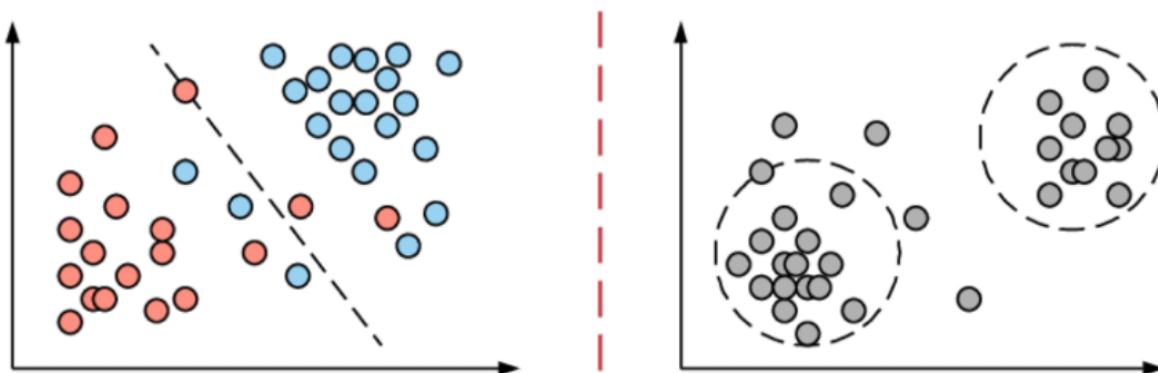
Još jedno glavno područje dubokog učenja je klasifikacija slika [4], koja primarno uključuje detekciju objekta na slici, prepoznavanje lica i identiteta osobe, identificiranje objekata na slikama kao što su znakovi za zaustavljanje, pješaci i oznake traka. Nadalje, duboko učenje se uspješno koristi za predviđanje molekularnih interakcija kako bi farmaceutske tvrtke mogle dizajnirati nove lijekove.

## 2.2. Algoritmi

Algoritmi dubokog i strojnog učenja [1, str. 98-105] obično se dijele na nadzirane (podaci o obuci označeni su točnim odgovorima) i nenadzirane (odgovori se ne prikazuju tijekom obuke) (Slika 4). Problemi algoritama nadziranog strojnog učenja dalje se dijele na klasifikaciju (predviđanje nenumeričkih odgovora) i regresiju (predviđanje numeričkih odgovora). Nenadzirano učenje dalje se dijeli na grupiranje (pronalaženje grupe sličnih predmeta), pridruživanje (pronalaženje zajedničkih nizova predmeta) i smanjenje dimenzionalnosti (projekcija, odabir značajki i ekstrakcija značajki).

Kada se govori o nadziranom učenju [4], takvo učenje se definira korištenjem označenih skupova podataka za obuku algoritama koji klasificiraju podatke ili točno predviđaju ishode. Kako se ulazni podaci unose u model, on prilagođava svoje vrijednosti sve dok se model pravilno ne prilagodi. Kao što je spomenuto, algoritmi nadziranog učenja mogu se podijeliti s obzirom na klasifikaciju i regresiju. Problem klasifikacije [4] zahtijeva izbor između dvije ili više klasa, obično s vjerojatnošću za svaku klasu. Primjerice, najpoznatiji algoritmi [1, str. 137] koji rješavaju probleme klasifikacija su stabla odlučivanja, k-najbližih susjeda i algoritam stroja potpornog vektora (eng. *Support Vector Machine (SVM)*). Neki primjeri [3] klasifikacije su česti primjeri filtriranja neželjene pošte ili spam-a, otkrivanje jezika na određenim web stranicama ili dokumentima, dok primjeri regresije su predviđanje cijena dionica nekih tvrtki, analiza ponude i potražnje itd.

Problem regresije koji je također jedan od problema nadziranog učenja, traži od modela predviđanje numeričkih vrijednosti. Na primjer, linearna regresija [1, str. 105, 106] koristi se za određivanje odnosa između zavisne varijable i jedne ili više nezavisnih varijabli, obično za predviđanje budućih ishoda. Kako se broj nezavisnih varijabli povećava, linearna regresija postaje višestruka. Uz linearu, postoji i logistička regresija koja se odabire kada je zavisna varijabla kategorička, što znači da imaju binarne izlaze kao što su "točno" i "netočno" ili "da" i "ne".



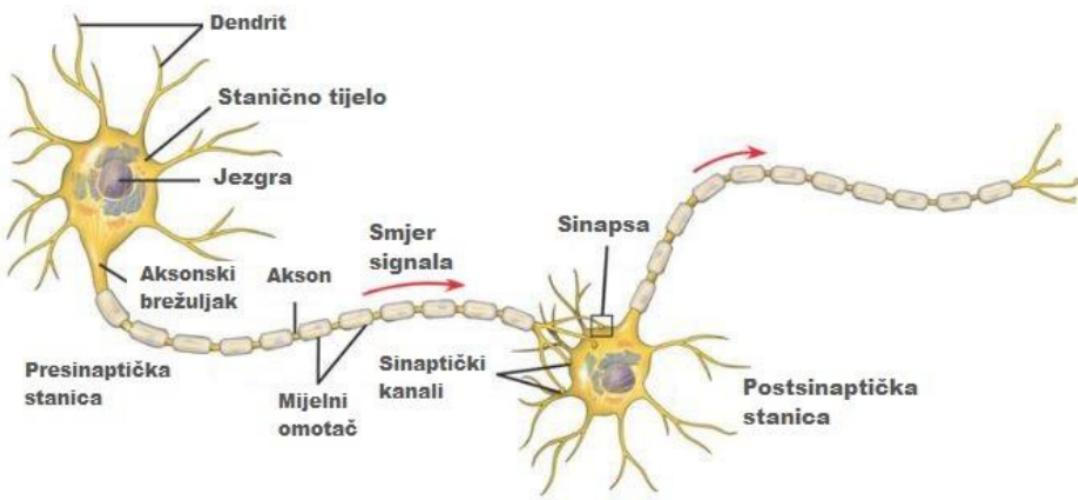
Slika 4: Nadzirano učenje (lijevo) i nenadzirano učenje (desno); prema: [1]

Za razliku od nadziranog, nenadzirano učenje [1], [4] koristi neoznačene podatke. Iz tih podataka pronalazi uzorke koji pomažu u rješavanju problema klasteriranja ili povezivanja. Uobičajeni algoritmi klasteriranja su hijerarhijski, k-srednje vrijednosti i Gaussovi modeli [1, str. 142].

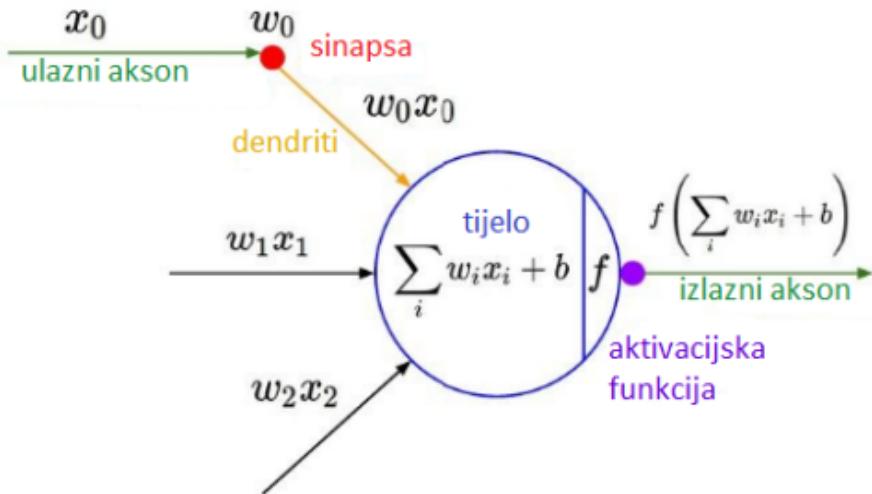
## 2.3. Neuronske mreže

Sada kada je poznat pojam dubokog učenja, njegove primjene i važnosti, može se prijeći na pojam umjetne neuronske mreže i njenih operacija.

Da bi se razumjelo kako radi umjetni neuron, prvo se treba razumjeti kako radi biološki neuron. Dakle, neuron ima tijelo, dendrite i akson [4] (Slika 5). Dendriti primaju informacije ili signale od drugih neurona koji se s njima povezuju. Obrada informacija odvija se u tijelu stanice odnosno ono preuzima sve informacije koje dolaze iz različitih dendrita i obrađuje te informacije. Aksoni šalju izlazni signal drugom neuronu za protok informacija.



Slika 5: Biološka neuronska mreža; prema: [2]



Slika 6: Umjetni neuron; prema: [5]

Gledajući u sliku 6, može se primjetiti kako umjetni neuron replicira biološki neuron.

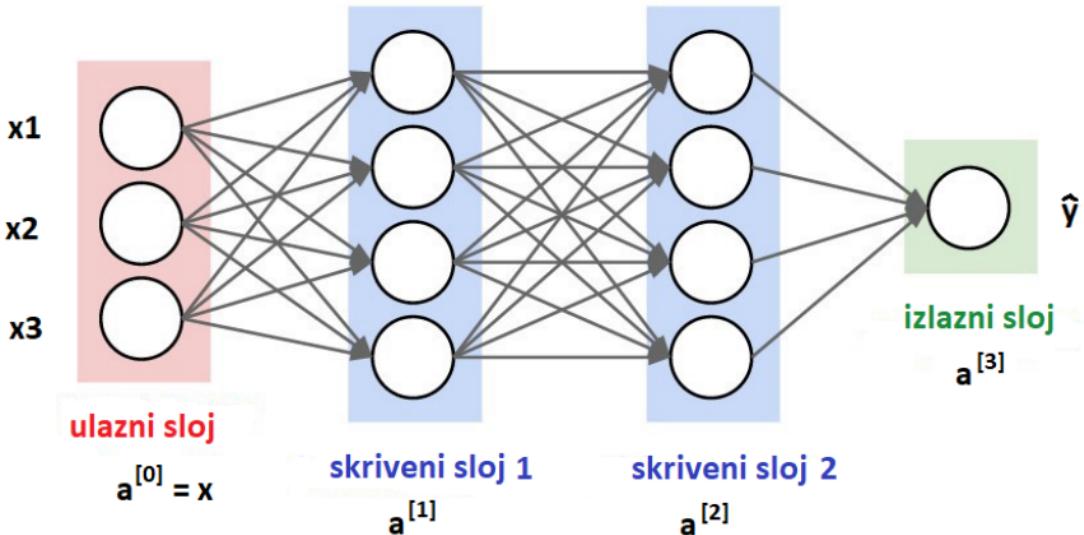
Matematički, on se prikazuje kao [5]:

$$y_k = \varphi \left( \sum_{i=1}^m x_i \cdot w_{ki} + b_k \right)$$

gdje

- $x_1, x_2, x_3, \dots, x_m$  su ulazni signali,
- $w_1, w_2, w_3, \dots, w_m$  su odgovarajuće težine neurona,
- $b_k$  je pomak koji može imati pozitivnu ili negativnu vrijednost te imati utjecaj na povećanje ili smanjenje vrijednosti ulaznih signala u aktivacijsku funkciju,
- $\varphi$  je aktivacijska funkcija koja prima ulazne signale od susjednih neurona te ih zatim zbraja i na taj zbroj djeluje nekom određenom transformacijom uzimajući u obzir vrijednost pomaka  $b_k$ .

Ulaz u neuron predstavlja ulazni sloj, sami neuron je u biti skriveni sloj, a izlaz do posljednjeg neurona je izlazni sloj [2].

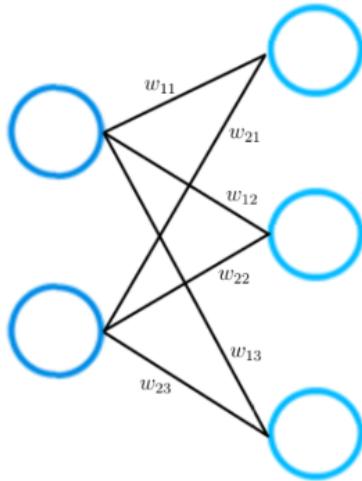


Slika 7: Arhitektura neuronske mreže; prema: [2]

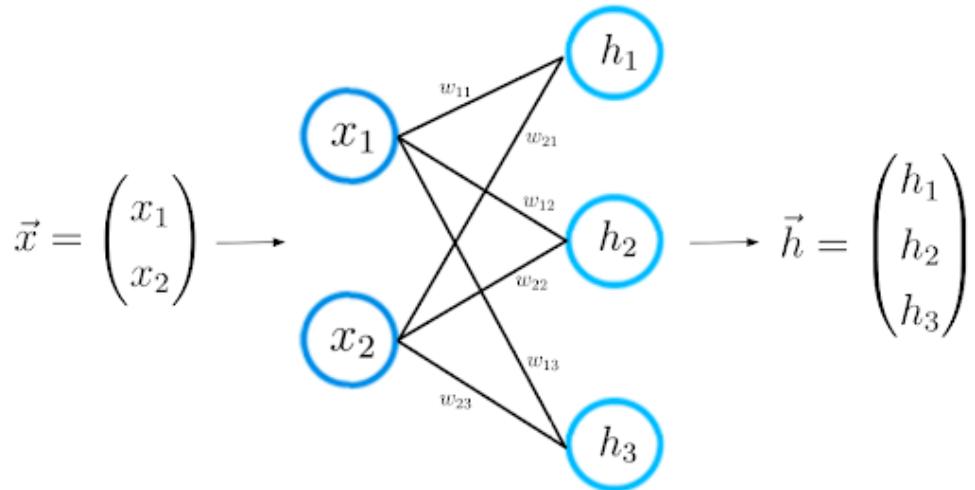
Tipična arhitektura neuronske mreže prikazana je na slici 7 i sastoji se od nekoliko slojeva od kojih su već prije spomenuti najbitniji slojevi – ulazni, skriveni i izlazni. Na sljedećim slikama istaknut je matematički proces učenja neuronske mreže [2].

Veza između dva neurona predstavljena je s vrijednošću  $w$  koja sadrži indeks (Slika 8). Prva vrijednost indeksa predstavlja broj neurona u sloju iz kojeg veza dolazi, a druga vrijednost predstavlja broj neurona u sloju do kojeg vodi veza.

Za zadani ulazni vektor  $x$ , neuronska mreža izračunava vektor predviđanja koji je nazvan  $h$ .



Slika 8: Veze slojeva; prema: [2]



Slika 9: Širenje; prema: [2]

„Na slici 9 se prikazuje proces širenja unaprijed. Koristeći ulazni vektor  $x$  i maticu vrijednosti  $W$  koja povezuje dva sloja neurona, izračunava se skalarni produkt između vektora  $x$  i matrice  $W$ . Rezultat ovog skalarnog produkta je drugi vektor, nazvan  $z$ .

$$\begin{aligned}\vec{x}^T \cdot W &= (x_1, x_2) \cdot \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix} \\ &= (x_1w_{11} + x_2w_{21}, x_1w_{12} + x_2w_{22}, x_1w_{13} + x_2w_{23}) \\ &= (z_1, z_2, z_3) = \vec{z}, \vec{h} = \sigma(\vec{z})\end{aligned}$$

Konačni vektor predviđanja  $h$  dobiva se primjenom takozvane aktivacijske funkcije na vektor  $z$ . Aktivacijska funkcija je označena slovom  $\sigma$  i ona predstavlja nelinearnu funkciju koja izvodi nelinearno preslikavanje iz  $z$  u  $h$ .“ [2]

Jačina veze [2] između neurona iskazuje se kao linearna kombinacija težinskih vrijednosti po-

vezanih neurona iz prethodnog sloja, u odnosu na promatrani neuron.

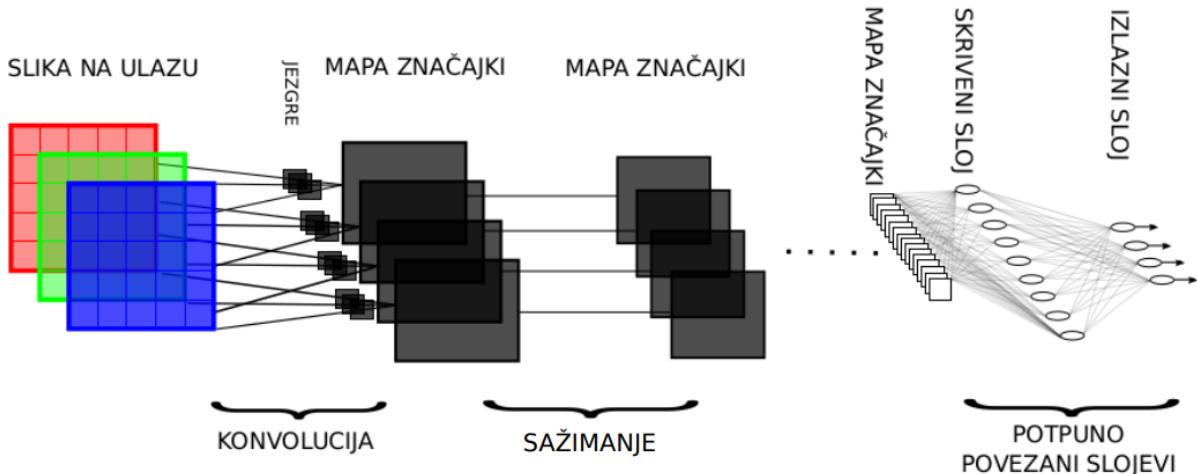
### 2.3.1. Konvolucijske neuronske mreže

Konvolucijska neuronska mreža (*CNN*, eng. *Convolutional Neural Network*) [1, str. 326] je specijalizirana vrsta neuronskih mreža za preprocesiranje nestrukturiranih podataka, posebice vezanih za slike, tekst, zvuk i govor. Primjerice, *CNN*-ovi imaju ponavljajuće slojeve neurona koji se primjenjuju u prostoru (za slike) ili vremenu (za audio signale, itd.). Za slike, ti neuronski slojevi mogu se tumačiti kao 2D konvolucijske jezgre koje se ponavljajući primjenjuju na svaki dio slike. Za govor, oni se mogu promatrati kao 1D konvolucijske jezgre.

*CNN*-ovi su posebno korisni za pronalaženje uzoraka u slikama za prepoznavanje predmeta, lica i scene. Oni također mogu biti vrlo učinkoviti u klasificiranju neslikovitih podataka kao što su audio, vremenske prognoze i signalni podaci.

Kada je riječ o pronalaženju uzoraka iz slika, konvolucijske neuronske mreže mogu imati desetke ili stotine slojeva (Slika 10 i 12), od kojih svaki uči otkrivati različite značajke slike. Filtri se primjenjuju na svaku sliku za obuku u različitoj razlučivosti, a izlaz svake modificirane slike koristi se kao ulaz za sljedeći sloj. Filtri mogu započeti s vrlo jednostavnim značajkama, kao što su svjetlina i rubovi i dodati složenost kako bi se jedinstveno definirale karakteristike objekta [6, str. 90].

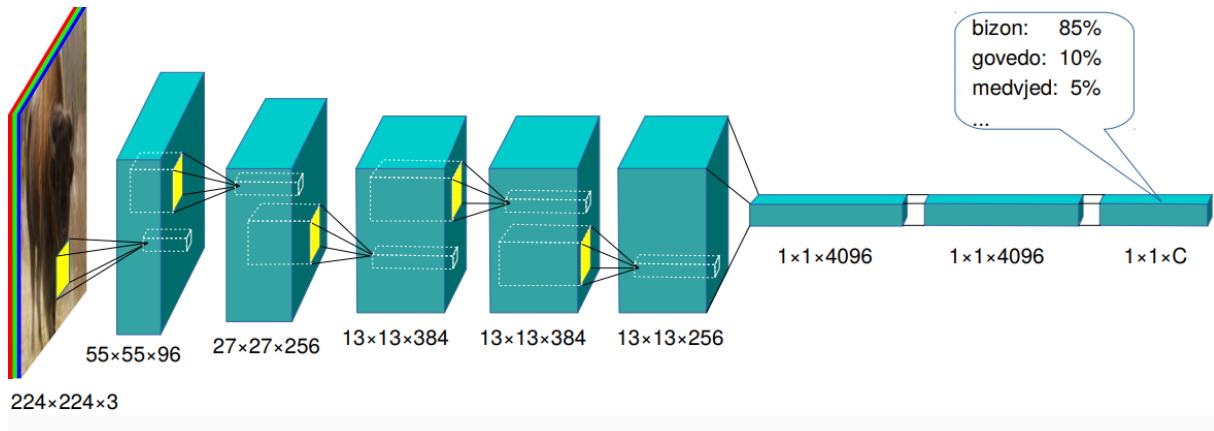
Postoje četiri glavna koraka [7] u izvođenju operacija koji mijenjaju podatke s ciljem učenja značajki specifičnih za podatke: konvolucija, *ReLU* (eng. *Rectified Linear Unit*) aktivacijska funkcija i potpuna povezanost.



Slika 10: Arhitektura konvolucijske neuronske mreže; prema: [2]

Prvi sloj koji prima ulazni signal naziva se konvolucijski filter. Konvolucija (eng. *Convolution*) [1, str. 327] je proces u kojem mreža pokušava označiti ulazni signal pozivajući se na ono što je naučila u prošlosti. Primjerice, ako ulazni signal izgleda kao slika mačke koja je već viđena, referentni signal "mačke" bit će pomiješan ili konvolviran s ulaznim signalom. Rezultirajući izlazni signal zatim se proslijeđuje sljedećem sloju. Sažimanje (eng. *pooling*) [7] pojednostavljuje izlaz izvođenjem nelinearnog smanjivanja uzorkovanja, smanjujući broj para-

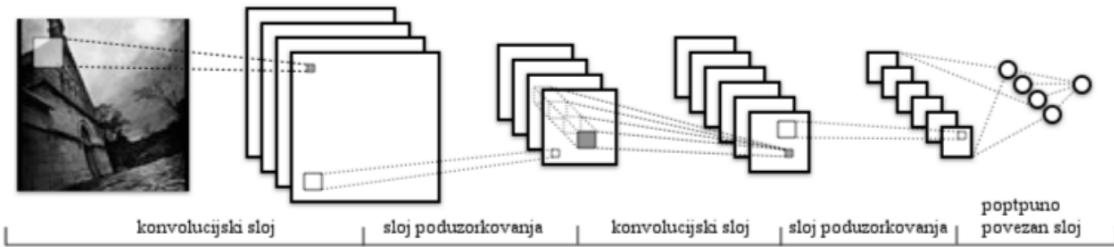
metara koje mreža mora naučiti (Slika 11).



Slika 11: Praktični primjer na temelju prepoznavanja životinje: bizona, goveda i medvjeda; prema: [7]

Potpuno povezani (eng. *Fully Connected (FC)*) [6, str. 89, 90] slojevi su slojevi gdje je svaki ulaz povezan sa svim neuronima. Ako postoji, potpuno povezani slojevi obično se nalaze blizu kraja CNN arhitekture i mogu se koristiti za optimizaciju ciljeva kao što su rezultati razreda.

Aktivacijska funkcija (*ReLU* (eng. *Rectified Linear Unit*)) [6, str. 13-15] omogućava bržu i učinkovitiju obuku preslikavanjem negativnih vrijednosti na nulu i zadržavanjem pozitivnih vrijednosti. To se ponekad naziva aktivacijom jer se samo aktivirane značajke proslijedu na sljedeći sloj.



Slika 12: Slojevi konvolucije, poduzorkovanja i potpuno povezani sloj; prema: [7]

Jedan od primjera primjene konvolucijskih mreža su aplikacije [8] za pretvaranje govora u tekst kod kojih je najbitnija stvar otkrivanje ključnih riječi. Dakle, aplikacija identificira određene ključne riječi ili fraze i može ih koristiti kao upute.

Još jedan primjer [8] korištenja CNN-ova je za semantičku segmentaciju gdje se identificira svaki piksel na slici s odgovarajućom oznakom klase. Semantička segmentacija može se koristiti u aplikacijama kao što su autonomna vožnja, industrijska inspekcija, klasifikacija terena i medicinsko snimanje. Konvolucijske neuronske mreže osnova su za izgradnju semantičkih segmentacijskih mreža.

### 3. Računalni vid

Ljudi su desetljećima sanjali o stvaranju strojeva s karakteristikama ljudske inteligencije, odnosno strojeva koji mogu razmišljati i ponašati se poput ljudi. Jedna od najfascinantnijih ideja je omogućiti računalima da "vide" i tumače svijet oko sebe (Slika 13). Tadašnja fantazija danas postaje stvarnost [9, str. 3].

Zahvaljujući napretku umjetne inteligencije i računalne snage, tehnologija računalnog vida napravila je velik korak prema integraciji u naš svakodnevni život.

Računalni vid je polje umjetne inteligencije [10] koje omogućuje računalima i sustavima da izvuku značajne informacije iz digitalnih slika, videa i drugih vizualnih *inputa* te da djeluju ili daju preporuke na temelju tih informacija. Ako umjetna inteligencija računalima omogućuje razmišljanje, računalni vid im omogućuje da vide, promatraju i razumiju.

Kao znanstvena disciplina [10], računalni vid bavi se teorijom iza umjetnih sustava (neuronских mreža) koji dohvaćaju informacije iz slika. Podaci o slikama mogu imati mnoge oblike, kao što su video sekвенце, prikazi s više kamera ili višedimenzionalni podaci iz medicinskih skenera. Kao tehnička disciplina [10], računalni vid nastoji "primijeniti svoje teorije i modele na stvarni svijet" za dobrobit čovječanstva.



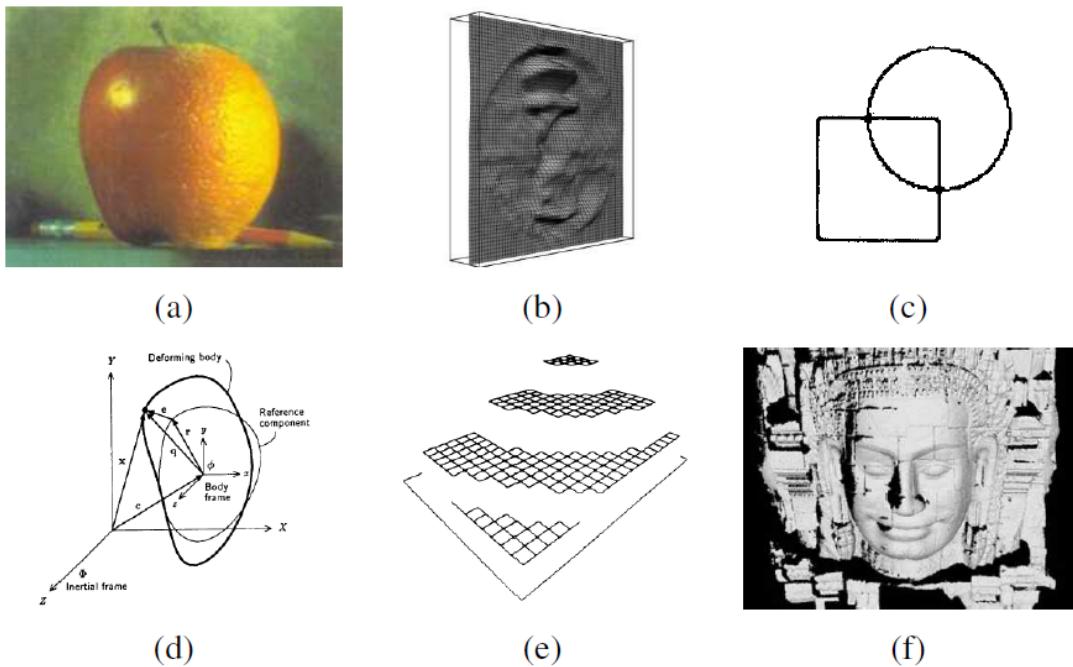
Slika 13: Primjer korištenja računalnog vida u stvarnom životu; prema: [9]

### 3.1. Pregled povijesti

U ovom odjeljku sadržan je kratak pregled glavnih razdoblja razvoja računalnog vida tijekom posljednjih 50 godina (Slika 14.) s naglaskom na napredak u razvoju algoritama i obradu podataka.



Slika 14: Vremenski raspored odabralih tema istraživanja u računalnom vidu; prema: [9]

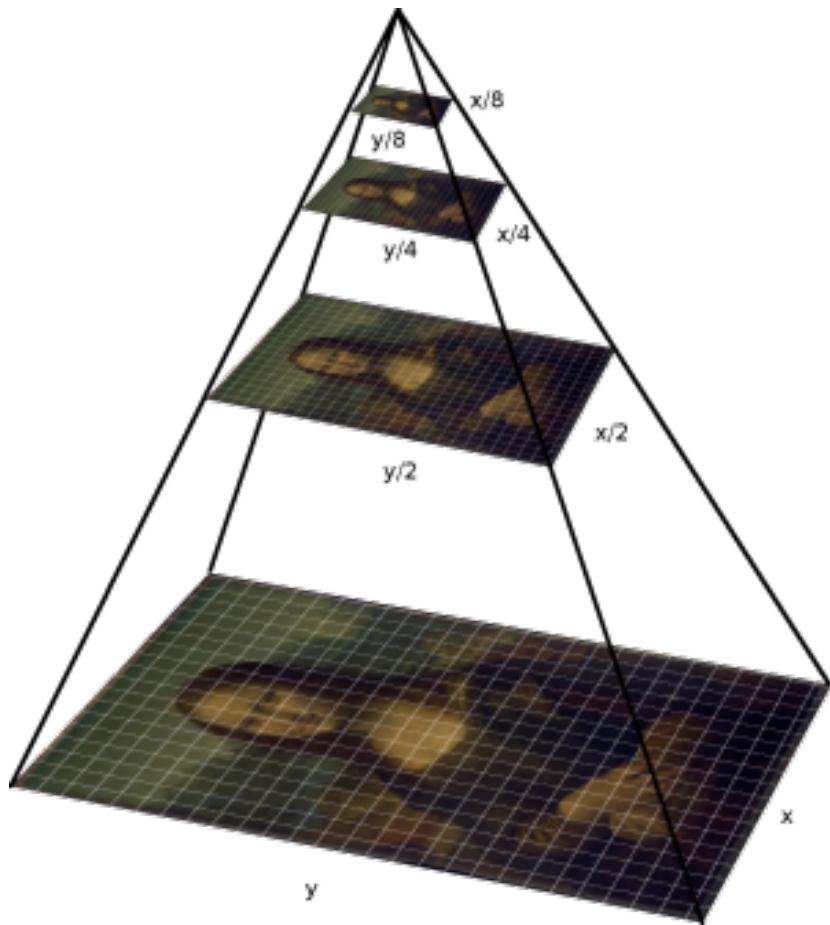


Slika 15: Primjeri algoritama računalnog vida iz 1980-ih: (a) piramidalno spajanje, (b) oblik iz sjenčanja, (c) otkrivanje rubova, (d) fizički bazirani modeli, (e) površinska rekonstrukcija temeljena na regularizaciji, (f) prikupljanje podataka o rasponu; prema: [9]

Pojava računalnog vida 70-ih godina 20. stoljeća [9, str. 11] smatrala se jednostavnim rješenjem za znatni napredak same inteligencije stroja. Nastajući kao metoda oponašanja ljudske inteligencije [9, str. 11], računalni vid se u početku znatno podcjenjivao. Smatralo se kako za vid nije potrebna inteligencija, implementiranje softvera za vid bi trebalo biti intuitivno.

Među prvim znanstvenicima koji je shvatio samu težinu izazova bio je i Marvin Minsky. Naime, u 1966. godini, Minsky je zajedno s bistrim studentom prve godine - Geraldom Sussmanom, proveo cijelo ljetno pokušavajući postići nekakav napredak [11, str. 781]. Generalna ideja bila je povezati računala s kamerama te ih navesti da opišu ono što vide. Iako su obojica bili vrlo optimistični, cijeli projekt je bio veliki neuspjeh, pokazalo se kako je razvoj takvog softvera iznimno komplikiran i problematičan. Samo polje računalnog vida i danas nakon nekoliko značajnih napredaka kaska za ljudskim vidom [12]. Sposobnost čovjeka da prepoznaje mnogo lica, lica različite životne dobe u vrlo širokom spektru nepoželjnih uvjeta, teško da će i u skorom vremenu biti sustignuta.

Unutar narednog desetljeća pozornost se okrenula druge tipove problema. Glavni fokus 80-ih godina bile su složenije matematičke tehnike korištenih u analizi slika i scena [9, str. 15]. Jedna od novih metoda je tzv. piramida slika (eng. *image pyramid*) ili jednostavnije samo piramida. Piramida je metoda unutar koje su slika (ili signal) podložni izglađivanju (eng. *smoothing*) i poduzorkovanju *subsampling* [13]. Do samog kraja desetljeća, obrada podataka nad 3D prostorima (priključivanje, spajanje, modeliranje i identifikacija) bila je aktivno istraživana.

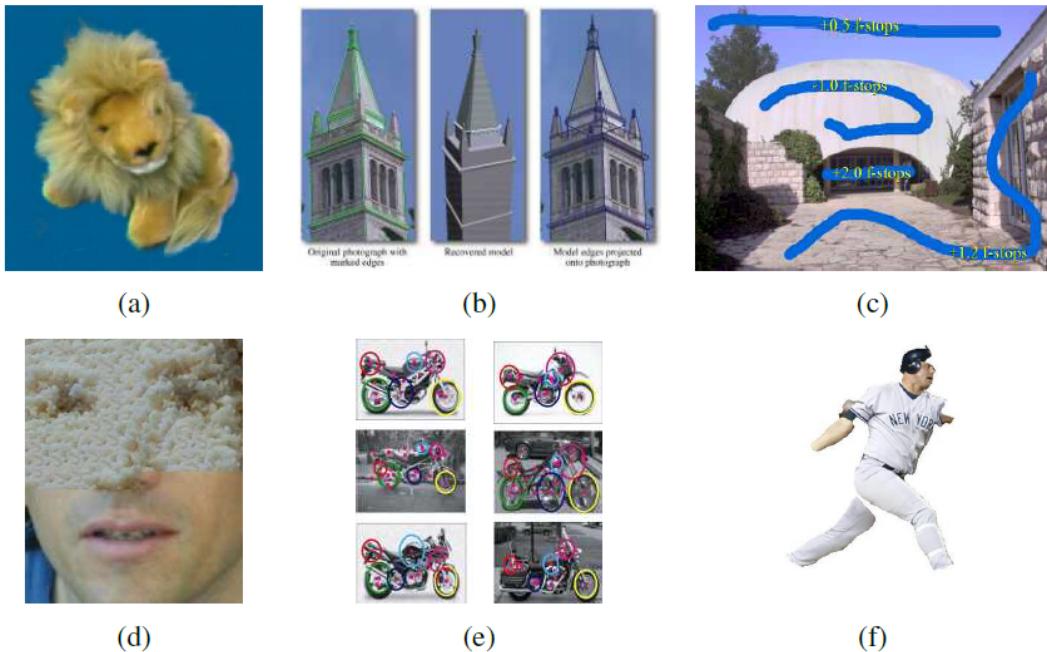


Slika 16: Piramida slika; prema: [9]

Ovisno o vremenu, neke od spomenutih tema nastavljaju se naširoko razvijati 1990-ih godina. Jedan od problema koji su se nastojali riješiti bio je i problem pokreta (eng. *motion problem*), odnosno procjene kretanja [9, str. 15]. Ovaj problem središnji je dio mnogih projekata i radova i danas. U tom razvoju pokušane su tehnikе poput projektivnih invarijanti

(eng. *projective invariants*), rekonstrukcija projekcije (eng. *projective reconstructions*), tehnika faktorizacije (eng. *factorization techniques*) i dr., no na kraju je uvedena uporaba potpune globalne optimizacije (eng. *full global optimization*) [9, str. 15-18]. Pomoću ove tehnike izgrađeni su potpuno automatizirani sustavi 3D modeliranja.

Može se reći kako razvoj računalnog vida prati i sami razvoj računala, pametnih uređaja, odnosno generalno tehnologije. Tako je početkom 21. stoljeća (Slika 17), točnije u prvom desetljeću 21. stoljeća, ubrzani razvoj grafike i vida. Izniman doprinos napretku je i prisutnost velikih skupova podataka (eng. *big data*), no iako su oni prisutni, njihova prisutnost se kreće u potpunosti iskorištavati u narednom desetljeću. Najvažniji trend ovog desetljeća je primjena složenih tehnika strojnog učenja na probleme računalnog vida, kao što su neuronske mreže, nadzirano i nenadzirano učenje itd., koje su u potpunosti preuzele vizualno prepoznavanje i većinu drugih aspekata računalnog vida [9, str. 19].



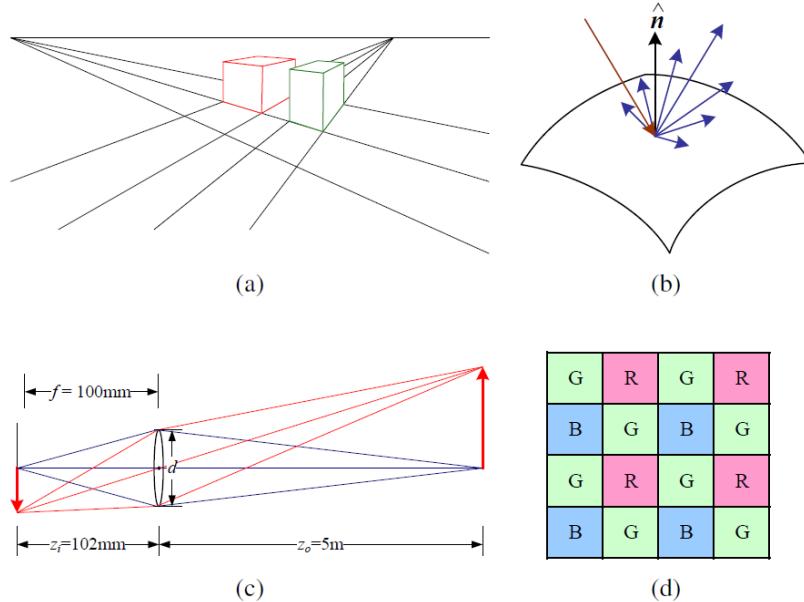
Slika 17: Primjeri algoritama računalnog vida iz 2000-ih: (a) renderiranje, (b) modeliranje, (c) interaktivno preslikavanje tonova, (d) sinteza teksture, (e) prepoznavanje, (f) regionalno prepoznavanje; prema: [9]

Zahvaljujući ogromnim napretcima unutar cijelog polja umjetne inteligencije, ali i susjednih polja poput znanosti o podacima, razvoj računalnog vida dosegnuo je nove visine. Po prvi puta u povijesti moguće je prikupiti, organizirati i spremiti ogromne količine skupova podataka čime se revolucionarizirao razvoj algoritama za prepoznavanje slika [9, str. 20]. SLAM (simultana lokalizacija i mapiranje, eng. *Simultaneous Localization And Mapping*) i VIO (vizualna inercijalna odometrija, eng. *Visual Inertial Odometry*) [9, str. 20, 21, 22] poznati su algoritmi računalnog vida koji se koriste unutar naprednijih platforma poput dronova i autonomnih vozila [9, str. 22]. Koristeći te algoritme moguće je izraditi točne 3D karte u stvarnom vremenu za, npr. autonomni let u izazovnim scenama kao što su šume ili urbana okolina.

Ukratko rečeno, računalni vid razvijao se kroz zadnjih 50 godina uz razne uspone i padove. Početni zanos, entuzijazam i optimizam vrlo je brzo zamijenjen realnošću - računalni vid vrlo je složeno područje čija kompleksnost znatno sprječava veći napredak. Najveći napredak ostvaren je u prethodnom desetljeću zbog generalnog razvoja strojnog učenja, neuronskih mreža i drugih dijelova umjetne inteligencije. Zahvaljujući tome, računalni vid danas ima primjenu u mnogim dijelovima našeg života - od kamera u pametnim mobitelima, preko autonomnih vozila i prometa pa sve do medicine. Jednostavno rečeno, primjena je mnogo, a nastavkom razvoja mogućnosti će biti mnogo veće.

## 3.2. Digitalna obrada slike

Prije analize i obrade slike potrebno je naglasiti određene pojmove koji opisuju obradu slike. Pojnta je u tom da je potrebno razumjeti proces formiranja slike (Slika 18) koji proizvodi određenu sliku s obzirom na skup uvjeta osvjetljenja, prostornu geometriju, svojstva površine i optiku kamere (Slika 19).



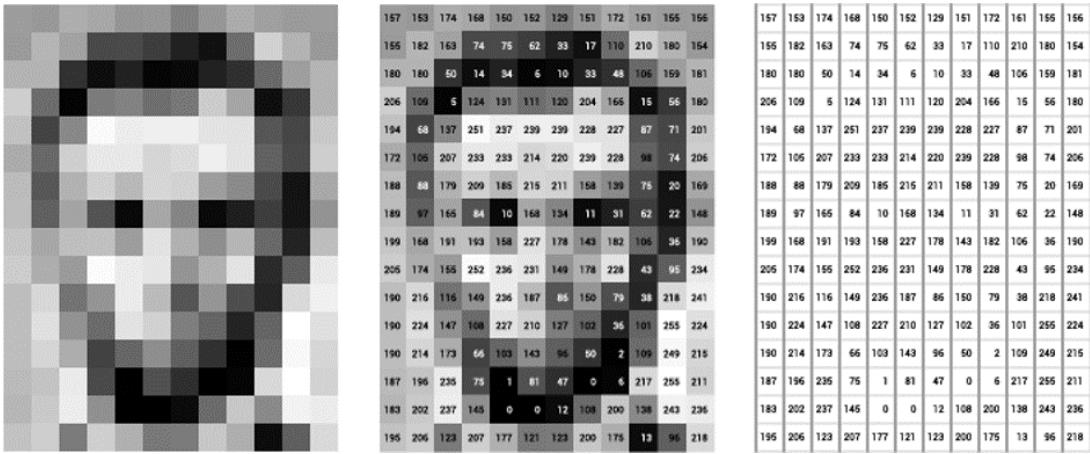
Slika 18: Komponente procesa oblikovanja slike: (a) perspektivna projekcija, (b) raspršenje svjetlosti pri udaru o površinu, (c) optika leća, (d) Bayerov niz filtera u boji; prema: [9]

„Slika, definirana matematičkim rječnikom, može se prikazati kao dvodimenzionalna funkcija,  $F(x, y)$ , gdje su  $x$  i  $y$  prostorne koordinate, a veličina  $F$  je bilo koji par koordinata  $(x, y)$  naziva se intenzitet slike u tom trenutku. Drugim riječima, slike se mogu definirati 2D nizovima posebno raspoređenim u retke i stupce. Digitalna slika sastoji se od konačnog broja elemenata, od kojih svaki ima određenu vrijednost na određenom mjestu. Ti se elementi nazivaju slikovni elementi, objektni elementi u slikama i pikseli. Od nabrojanih, pikseli se najčešće koriste za predstavljanje elemenata digitalnih slika.” [14]

Slike su, matematičkim zapisom, predstavljene kao matrice te zbog toga postoji sljedeći zapis [14]:

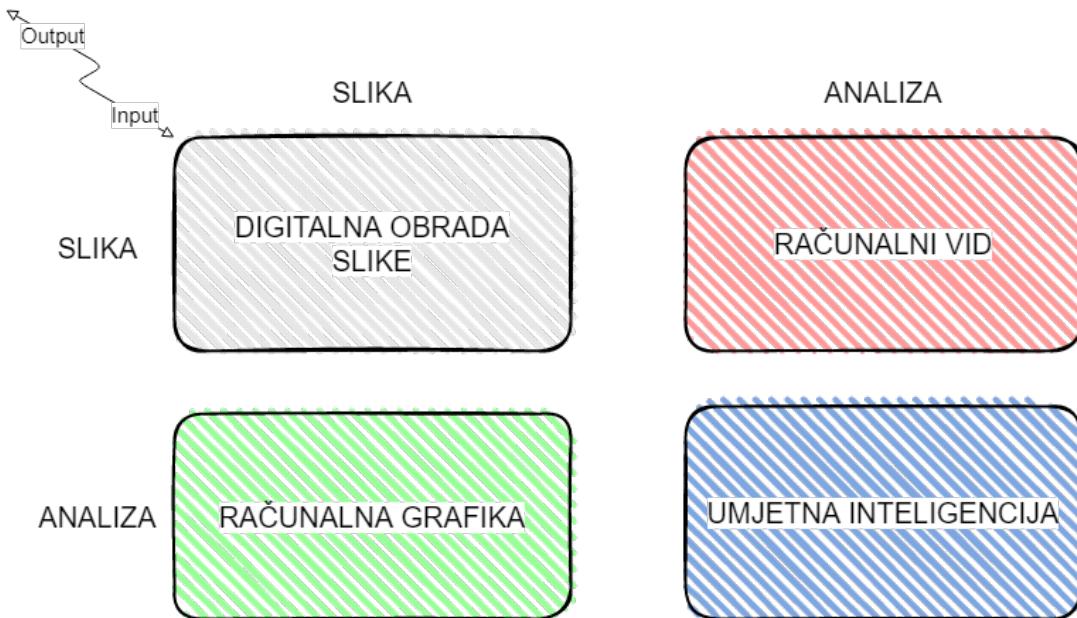
$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & f(0, 2) & \dots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & f(1, 2) & \dots & f(1, N - 1) \\ \vdots & \vdots & \vdots & & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & f(M - 1, 2) & \dots & f(M - 1, N - 1) \end{bmatrix}.$$

Kao što se može vidjeti na slici 20 [14], obrada slike podijeljena je na nekoliko znans-



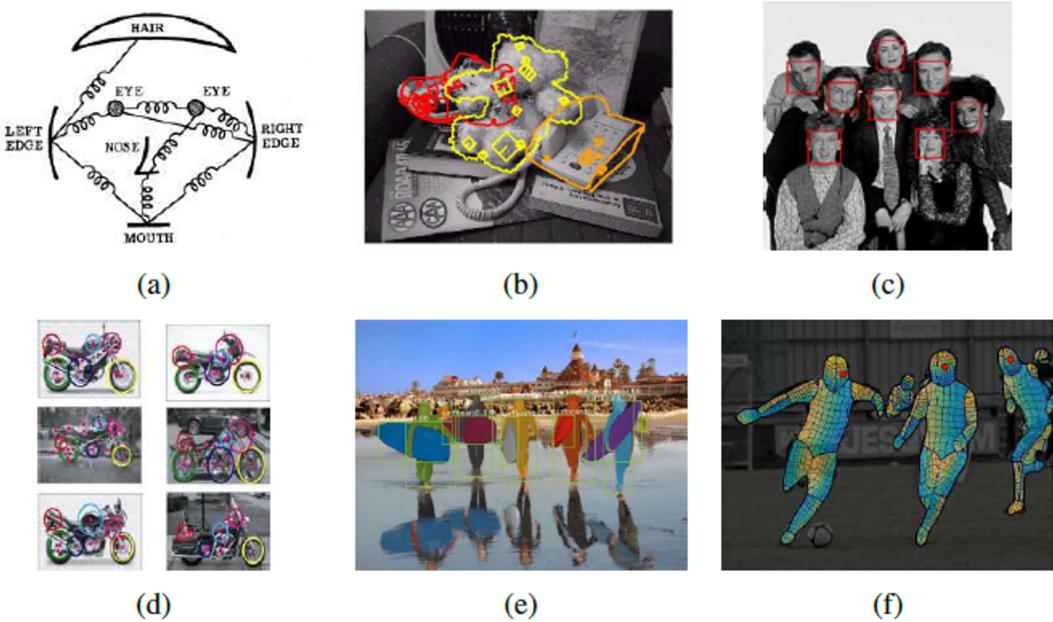
Slika 19: Vrijednosti boja pojedinačnih piksela; prema: [15]

tvenih područja na temelju slika (kao predmeta obrade) ili informacija kao ulaznih podataka [14]. Shodno tomu, može se reći da ako ulaz u proces obrade slike predstavlja sliku, tada je izlaz također slika jer se očekuje da taj proces nju obradi. Primjerice, poboljša njenu kvalitetu ili ju filtrira. Druga stvar, na primjer, ako za taj isti ulaz se opet stavi slika, a dobije se nekakva njena analiza, opis ili neki drugi tekstualni rezultat, tada se taj proces odvija pomoću računalnog vida. To znači da računalni vid analizira i daje rješenje u obliku opisa te slike. Primjerice, ulaz je bio slika psa, a proces je dao rezultat u tekstualnom obliku da je to pas. Nadalje, ako je sada obrnut proces, tj. ulaz je opis u tekstualnom obliku ili kod, krajni rezultat slika, to se može nazvati računalnom grafikom jer je tu proces stvaranja odnosno prikaza slike iz date ulazne vrijednosti. I na kraju, ako se ulaz opet sastoji od nekakvog opisa ili programskog koda, a u ovom slučaju krajnji rezultat predstavlja opis ili analizu, onda se to može nazvati procesom koji je bio obrađivan uz pomoć umjetne inteligencije.



Slika 20: Područja koja djeluju ovisno o slici ili vrsti informacije

### 3.3. Prepoznavanje

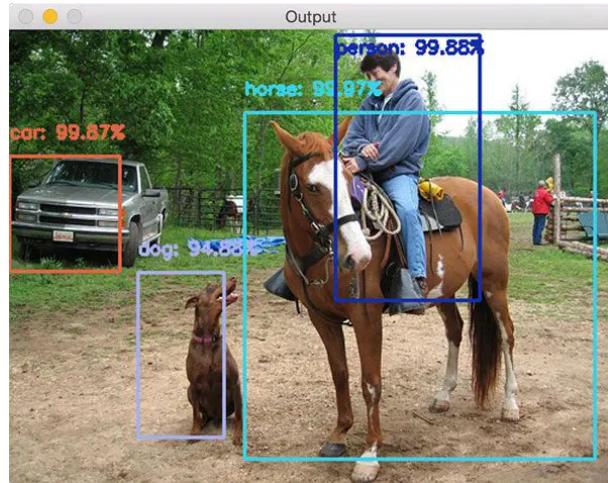


Slika 21: Razne vrste prepoznavanja: (a) prepoznavanje lica sa slikovnim strukturama, (b) prepoznavanje instanci (poznatog objekta), (c) prepoznavanje lica u stvarnom vremenu, (d) prepoznavanje temeljeno na značajkama, (e) segmentacija instance pomoću *Mask R-CNN*, (f) procjena poze; prema: [9]

#### 3.3.1. Prepoznavanje objekata

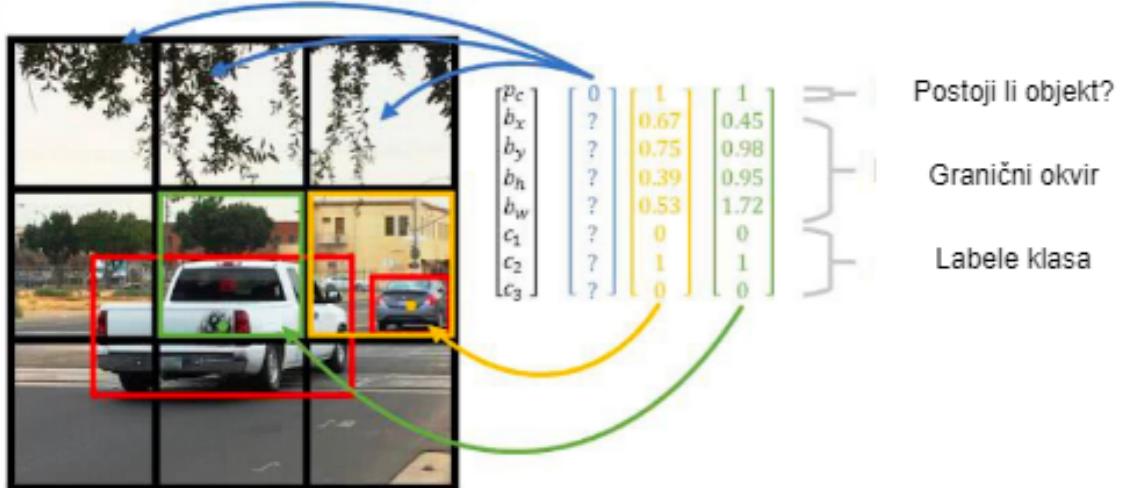
Prepoznavanje objekata opći je izraz koji se koristi za opisivanje povezanih zadataka računalnog vida koji uključuju prepoznavanje objekata na digitalnim fotografijama [9, str. 349, 350]. Kada osoba govori o "prepoznavanju predmeta" obično misli na "otkrivanje predmeta" (Slika 22). Klasifikacija slike uključuje predviđanje klase jednog objekta na slici. Lokalizacija objekta odnosi se na identificiranje lokacije jednog ili više objekata na slici i crtanje okvira oko njegovog opsega. Detekcija objekata kombinira ova dva zadatka za lociranje i klasificiranje jednog ili više objekata na slici.

Samo po sebi [9, str. 346, 347, 349], prepoznavanje objekata se može podijeliti na prepoznavanje instanci i prepoznavanje klase te kombinirati četiri tehnike: prepoznavanje slike, lokalizaciju objekta, detekciju objekta i segmentaciju slike. Prepoznavanje objekata analizira značajke slike i predviđa njezinu klasu ili kategorije putem klasifikatora, kao što je nadzirani model strojnog učenja primjerice, stroj potpornog vektora, *Adaboost* algoritam, stabla odlučivanja. Prepoznavanje klasa kao područje istraživanja u prepoznavanju objekata znatno se brže razvija za razliku od tehnika prepoznavanja instanci koje svoju svrhu i dalje imaju u komercijalnim aplikacijama kao što je npr. prepoznavanje prometnih znakova, pješaka, semafora itd.



Slika 22: Detekcija okoline videokamerom; prema: [9]

Regionalne konvolucijske neuronske mreže (*R-CNN*, eng. *Region-Based Convolutional Neural Network*) obitelj su tehnika za rješavanje zadataka lokalizacije i prepoznavanja objekata, dizajniranih za izvedbu modela [9, str. 383, 385]. Metoda *YOLO* (skraćeno od eng. *You Only Look Once*) (Slika 23) je klasa tehnika prepoznavanja objekata dizajniranih za brzu upotrebu u stvarnom vremenu. *YOLO* [9, str. 385] uključuje jednu neuronsku mrežu obučenu od kraja do kraja te uzima fotografije kao ulazne podatke i predviđa granične okvire i oznake klase za svaki granični okvir. Ova tehnika inače pruža manju točnost predviđanja (npr. više pogrešaka lokalizacije), a iako ima brzinu promjene okvira u sekundi od 45 (*fps*, eng. *frames per second*), verzija modela koja je optimizirana, može postići brzinu i do 155 *fps*-a.

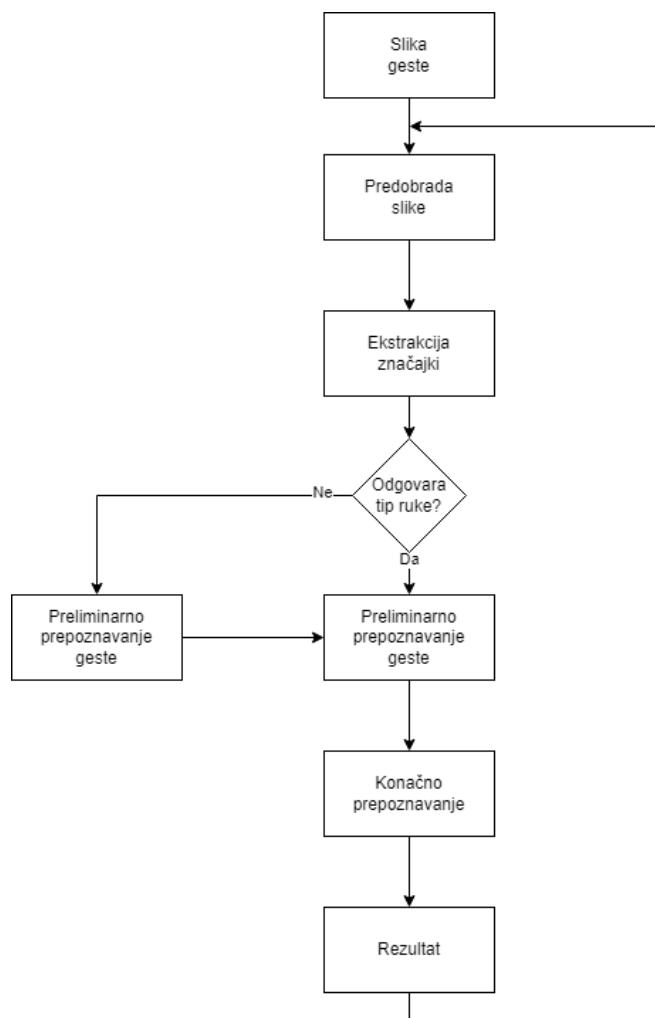


Slika 23: *YOLO*, algoritam za otkrivanje objekata; prema: [9]

### 3.3.2. Prepoznavanje geste ruke

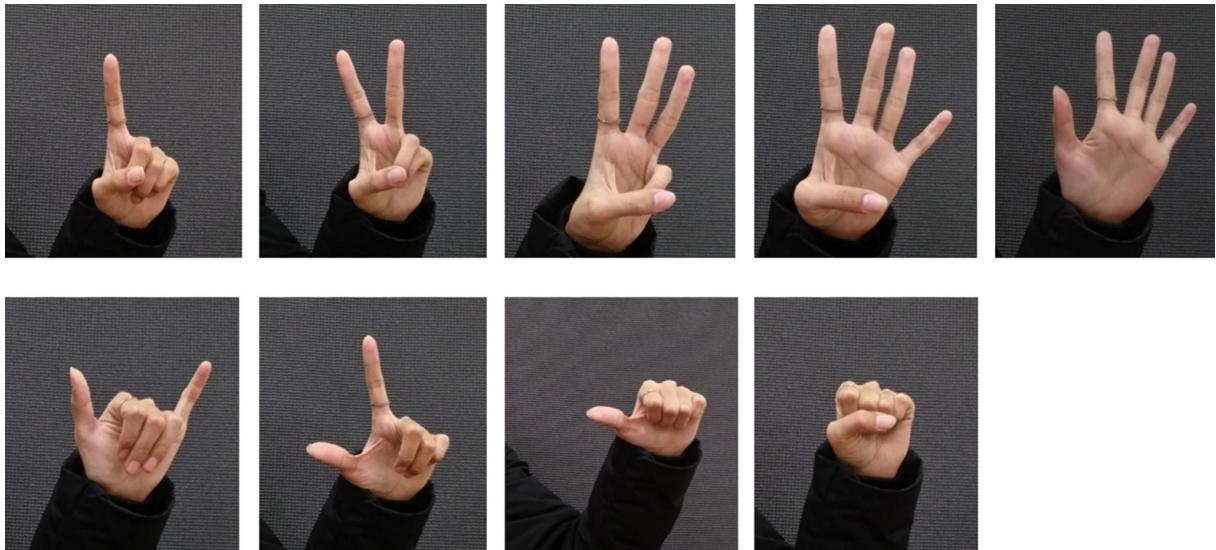
Ljudska svakodnevica ovih godina ne bi bila potpuna da nije napredovao razvoj računalnog vida [16], [17]. Primjerice, otključavanje mobilnih uređaja koristeći lice ili otisak prsta i sl. Među raznim tehnikama interakcije, korištenje ruku kao unosa je atraktivna metoda za uspostavljanje prirodne interakcije između čovjeka i računala. Korištenjem gesta korisnici npr. mobilnih uređaja mogu prenijeti više informacija u kraćem vremenu. Stoga, kako bi se poboljšalo sučelje između korisnika i računala, tehnologija interakcije između čovjeka i računala (*HCI*, eng. *Human-computer interaction*) ima veliku praktičnost [16], [17].

Sustave za prepoznavanje specifičnih ljudskih gesti može se koristiti za prijenos bilo kakvih informacija ili kontrolu bilo kojeg uređaja ili robota u uredskim i kućnim aplikacijama. Statičko stanje ruke može se definirati kao poza dok dinamično se može definirati kao gesta koja predstavlja fizički pokret nekog od tjelesnih organa kao što su šake, ruke, lice itd. kako bi se prenijele značajne informacije, kao što su primjerice "lajk" ili "dislajk" [17], [18]. Za uspješnu komunikaciju, pošiljatelj i primatelj moraju imati slične informacije za određenu gestu. Postoje dvije uobičajene metode tumačenja gesta u interakciji između čovjeka i računala, a to su metoda temeljena na rukavicama baziranim na podacima i metoda temeljena na viziji [18].



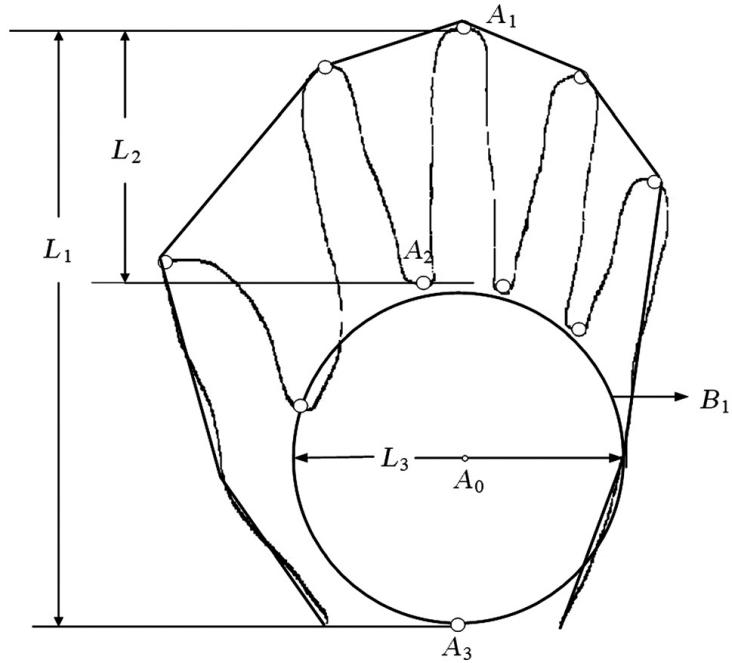
Slika 24: Dijagram toka prepoznavanja

Tehnike detekcije ruku temeljene na značajkama koriste algoritme poput Viola-Jones algoritma i algoritma temeljenog na transformacijama značajki bez skaliranja *SIFT* (eng. *Scale-Invariant Feature Transform*) [18]. Ovi algoritmi daju rezultate visoke točnosti, ali su vrlo osjetljivi na pozadinu. Jedan od ostalih pristupa za prepoznavanje gesta je algoritam *AdaBoost* (eng. *Adaptive Boosting*) koji integrira informacije iz iste klase objekata i obučava mrežu kombiniranjem svih slabih klasifikatora u jedan jak klasifikator [17]. *AdaBoost* algoritam učenja odabire najbolji slab klasifikator iz skupa pozitivnih i negativnih uzoraka slike. Osim toga, algoritam daje rezultate s većom preciznošću i većom brzinom, ali ponekad zahtijeva više vremena za obuku mreže.



Slika 25: Geste ruke; prema: [19]

Koraci obrade klasifikacije gesta [18] uključuju prikupljanje gesta (Slika 25), segmentaciju, filtriranje, predstavljanje kontura i klasifikaciju korištenjem različitih tehnika. Prema dijagramu toka prepoznavanja ruke (Slika 24) prelazi se na primjenjeni dio gdje se analizira dlan ruke te uz pomoć točki označenih na rubovima prstiju se pokušava detektirati ruka. Najupečatljivija linija je ona gdje se duljina dlana, označena kao  $L_1$ , uzima od vrha najdužeg prsta do samog dna dlana. Zatim se mjeri duljina najdužeg prsta  $L_2$  od vrha pa do njegovog dna. I za kraj se mjeri širina dlana koja je označena s  $L_3$ .



Slika 26: Dijagram mjerena duljine dlana; prema: [19]

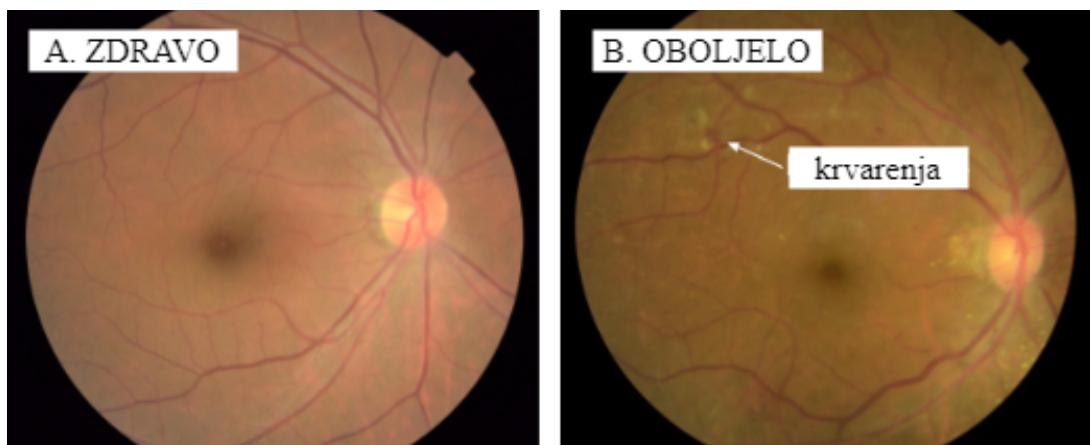
Na temelju prethodnih mjerena (Slika 26), određuju se relativni položaji dlana i prstiju.  $A_1$  predstavlja ordinatu vrha srednjeg prsta,  $A_2$  najnižu točku srednjeg prsta zatim  $A_3$  točku dna dlana te  $A_0$  točku u središtu dlana.  $L_1$ ,  $L_2$  i  $L_3$  izračunavaju se na sljedeći način [19]:

$$\left\{ \begin{array}{l} L_1 = A_1 - A_3 \\ L_2 = A_1 - A_2 \\ L_3 = 2 \times (A_0 - A_3) \end{array} \right.$$

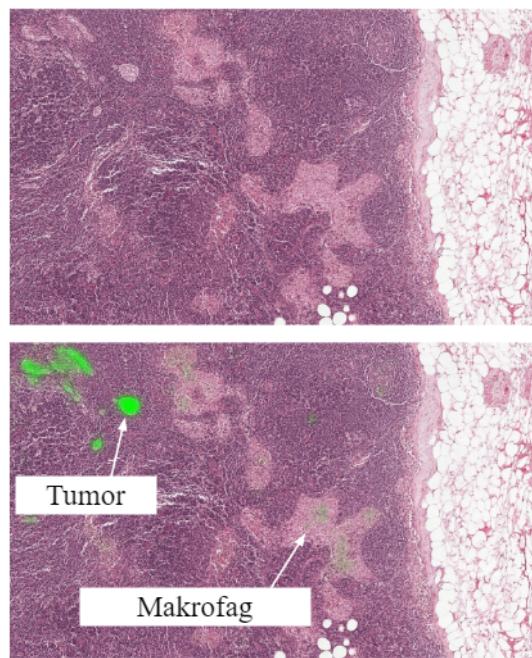
U ovoj sekciji obrađivane su samo slike gesta dlana dok neki ostali dijelovi, npr. šaka ruke, nisu obrađeni. Slike gesta se filtriraju u fazi predobrade slike, kao što je prikazano na grafu 24, a zatim se pretvaraju iz RGB prostora boja u YCbCr prostor boja. S obzirom na to da tonovi boje kože imaju dobro grupiranje u YCbCr prostoru boja, može se postići segmentacija gesta [19]. I na kraju se postiže sami rezultat, odnosno *output* koji je dobiven postupkom konačnim prepoznavanjem.

### 3.4. Primjena

Slikovite informacije imaju ključnu ulogu u medicinskoj dijagnozi jer čine gotovo većinu medicinskih i zdravstvenih informacija [15]. Upravo zato, veliki broj dijagnoza temelji se na obradama medicinskih slika poput rendgenskih snimaka, magnetskih rezonanci i mamografija [15]. Segmentacija slike je također pokazala svoju učinkovitost u analizi medicinskih skeniranja. Na primjer [15], algoritmi računalnog vida mogu otkriti dijabetičku retinopatiju, najbrže rastući uzrok sljepote. Računalni vid može obraditi slike stražnjeg dijela oka (Slika 27) i procijeniti prisutnost i ozbiljnost bolesti.



Slika 27: Primjeri fotografija retinalnog fundusa. Slika s lijeve strane je zdrava mrežnica (A), dok je slika s desne strane mrežnica s dijabetičkom retinopatijom (B) zbog brojnih krvarenja (crvenih mrlja); prema: [20]



Slika 28: Krupni plan biopsije limfnog čvora; prema: [21]

Može se istaknuti kako je preciznost u dijagnosticiranju različitih oblika raka je ključna.

Alati za računalni vid mogu otkriti metastaze raka ponekad točnije i od lječnika, sudeći prema [21]. Na slici 28 prikazan je krupni plan biopsije limfnog čvora čije tkivo sadrži metastaze raka dojke i područja koja izgledaju poput tumora, ali su dobroćudna. [21] Algoritam računalnog vida uspješno je identificirao područja tumora (svijetlozeleno) i nisu ga zbunila normalna područja koja su izgledala kao tumori.

Osim medicine, računalni vid omogućuje i automobilima razumijevanje okoline. Parametno vozilo [15] u svom sustavu ima popriličan broj kamera koje mogu pratiti iz različitih kutova i slati sliku kao ulaz softveru za računalni vid. Sustav obraduje sliku u stvarnom vremenu i otkriva oznake na cesti, objekte u blizini automobila (kao što su pješaci ili drugi automobili), semafore i još mnogo toga. Jedan od najvažnijih primjera primjene ove tehnologije je Teslin automobil. Prema [22], autopilot predstavlja napredni sustav pomoći vozaču koji povećava sigurnost i udobnost vožnje. Kada (i ako) se pravilno koristi, autopilot može smanjiti opterećenje vozača. Svako novo Teslino vozilo (Slika 29) opremljeno je s osam vanjskih kamera i snažnom vizualnom obradom za dodatni sloj sigurnosti. Sva vozila većinom napravljena za sjeverno-američko tržište sada koriste "Tesla Vision" temeljen na kamери za pružanje mogućnosti gotovo samostalne vožnje.



Slika 29: Unutrašnjost Tesla automobila te detekcija vanjskih objekata u prometu; prema: [22]

Sve spomenuto o računalnom vidu također omogućuje poljoprivrednicima da se učinkovito bave većim površinama. Tehnologije poput dronova, senzora i satelitskih slika uz strojno učenje omogućavaju prikupljanje i analiziranje golemih količina podataka, što za rezultat ima, ne samo znatan napredak u količini i kvaliteti generiranih poljoprivrednih proizvoda, nego i u mnogim slučajevima spas poljoprivredne industrije [23]. Zahvaljujući prethodno spomenutim tehnologijama, mnogi poljoprivredni aspekti cjelokupnog procesa mogu biti i automatizirani. Osim toga, moguće ih je koristiti i u predviđanju prinosa te predviđanju i/ili identificiranju gubitaka uzrokovanih insektima, bolestima ili prirodnim uvjetima [24]. Strojno učenje, odnosno računalni vid, danas moguće je koristiti i za rano otkrivanje i zaustavljanje biljnih bolesti [23]. Time se štedi vrijeme i novac, no najviše od svega stvara sigurnija i zdravija poljoprivredna praksa koja je iznimno podložna na vanjske, nepredvidive čimbenike.

## 4. Implementacije praktičnog dijela

### 4.1. Opis i svrha

Prepoznavanje gesta aktivno je područje istraživanja tehnologije interakcije čovjeka i računala. Ima mnoge primjene u kontroli virtualnog okruženja i prevodenju znakovnog jezika, kontroli robota, stvaranju glazbe itd. U ovom praktičnom dijelu dubokog učenja i računalnog vida o prepoznavanju gestikulacije rukama izraditi će se implementacija prepoznavanja gestikulacija u stvarnom vremenu koristeći *framework-e* odnosno biblioteke MediaPipe, Tensorflow, OpenCV i Python.

Nadalje, u drugoj implementaciji ovog praktičnog dijela rada, koristi se ruka kao virtualni miš za računalo koji može učiniti sve što i obični (fizički) miš za računalo, a da se ne dodirne sustav (tipkovnica, miš, *touchpad*). Koristi se web kamera računala za otkrivanje ruku. Zatim će stvoriti okvir oko ruke i fokusirati se na dva prsta: kažiprst i srednji prst. Kažiprst će djelovati kao pokazivač i pomicati ga uokolo, pomicat će se pokazivač uokolo po zaslonu. Sada, kako bi se uspješno kliknulo pomoću detekcije ruke, bitna je udaljenost između kažiprsta i srednjeg prsta. Ako su spojeni zajedno ili na jako maloj udaljenosti, tada će se izvesti klik. Nadalje, dodan je faktor izglađivanja jer je pokret bio stvarno neusklađen.

Za ispravan rad potrebno je instalirati sljedeće biblioteke:

- OpenCV,
- MediaPipe,
- Tensorflow,
- AutoPy.

Ukratko su opisane sve od potrebnih biblioteka, a o OpenCV će biti govora malo kasnije. MediaPipe je biblioteka za izgradnju multimodalnih (npr. video, audio), međuplatformskih (tj. Android, iOS, web, perifernih uređaja) primijenjenih metoda strojnog učenja [25].

Može se instalirati pomoću "pip install mediapipe."

Tensorflow je biblioteka za strojno učenje i duboko učenje kojeg je razvio tim Googlea. Može se koristiti u nizu zadataka, ali ima poseban fokus na duboke neuronske mreže [25].

Može se instalirati pomoću "pip install tensorflow."

AutoPy je jednostavna biblioteka *GUI* automatizacije za Python. Uključuje funkcije za upravljanje tipkovnicom i mišem, pronalaženje boja i bitmapa na zaslonu i prikaz upozorenja [25].

Može se instalirati pomoću "pip install autopsy."

Programski dio rada, rađen je *Microsoft Visual Studio Code* okruženju.

## 4.2. OpenCV

OpenCV (eng. *Open Source Computer Vision Library*) [26] biblioteka je softvera za računalni vid i strojno učenje otvorenog koda. OpenCV (Slika 30.) je napravljen kako bi osigurao zajedničku infrastrukturu za aplikacije računalnog vida i ubrzao korištenje strojne percepcije u komercijalnim proizvodima. Budući da je proizvod s *BSD* licencom, OpenCV tvrtkama olakšava korištenje i izmjenu koda.

Biblioteka ima više od 2500 optimizacijskih algoritama, uključujući velik broj klasičnih i najsuvremenijih algoritama računalnog vida i strojnog učenja. Ovi se algoritmi mogu koristiti [26] za otkrivanje i prepoznavanje lica, identifikaciju objekata, klasificiranje ljudskog ponašanja u videozapisima, praćenje kretanja kamere, praćenje pokretnih objekata, izdvajanje 3D modela objekata, generiranje 3D oblaka točaka iz stereo kamera, spajanje slika za generiranje visoka rezolucija slike, pronalaženje sličnih slika iz baza podataka slika, uklanjanje crvenih očiju iz Flash slika, praćenje pokreta očiju, prepoznavanje krajolika i postavljanje markera za preklapanje proširene stvarnosti i još mnogo toga. OpenCV ima više od 47 tisuća korisnika zajednice i procjenjuje se da ima više od 18 milijuna preuzimanja. Knjižnicu naširoko koriste tvrtke, istraživačke grupe i vladine agencije.

Uz dobro etablirane tvrtke kao što su *Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota* koje koriste biblioteku, postoji mnogo *startupova* kao što su *Applied Minds, VideoSurf i Zeitera*, koji u velikoj mjeri koriste OpenCV [26]. Upotreba OpenCV-a pri implementaciji uključuje spajanje slika uličnog prikaza, otkrivanje upada ili provala putem videonadzora, praćenje vozila, pomoć robotima u navigaciji, otkrivanje utapanja u bazenima, izvođenje interaktivne umjetnosti, pregled etiketa proizvoda u tvornicama diljem svijeta i brzo prepoznavanje lica.

Biblioteka sadrži C++, Python, Java i MATLAB sučelja i podržava Windows, Linux, Android i Mac OS. OpenCV se uglavnom oslanja na aplikacije za viziju u stvarnom vremenu i koristi MMX i SSE upute kada su dostupne [26]. Trenutno se aktivno razvijaju potpuno opremljena CUDA i OpenCL sučelja. Postoji preko 500 algoritama i oko 10 puta više funkcija koje podržavaju te algoritme [26]. OpenCV je izvorno napisan u C++ i ima predložak sučelja koji besprijekorno radi sa STL datotekama.

Može se instalirati pomoću "pip install opencv-python."



Slika 30: OpenCV i Python; prema: [26]

## 4.3. Analiza korištenih metoda

U obje implementacije koje slijede u nastavku, potrebno je uključiti biblioteke (kod 1.) koje su bile u prethodnim poglavljima opisane, a koje su neophodne za rad.

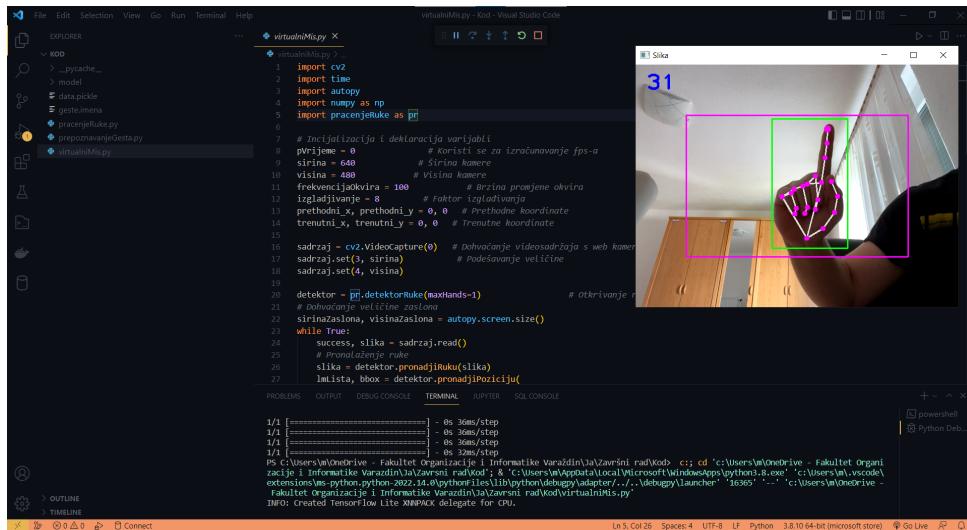
Isječak kôda 1: Biblioteke

```
1 import cv2
2 import time
3 import math
4 import autopy
5 import numpy as np
6 import mediapipe as mp
7 import tensorflow as tf
8 from keras.models import load_model
```

Osim dosad spomenutih biblioteka, tu se nalaze i biblioteke `time`, `math` te `numpy` koje za svrhu ovog rada predstavljaju pomoćne biblioteke. Bitno je spomenuti kako za datoteku `virtualniMis.py` je potrebno uključiti modul `pracenjeRuke.py`. Upravo taj modul (kod 2.) je jedan od ključnih elemenata izrade virtualnog miša jer se u njemu nalazi klasa (`detektorRuke`) koja sadrži neke od najbitnijih metoda za samu detekciju ruke prilikom prikaza u stvarnom vremenu.

Isječak kôda 2: Klasa detektorRuke

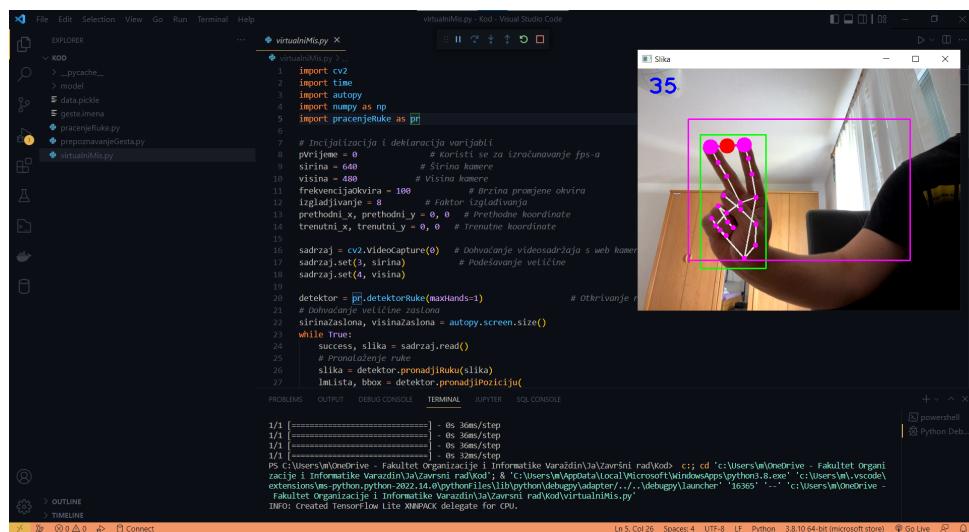
```
1 # uključivanje biblioteke zaglavlja za datoteku virtualniMis.py
2 import pracenjeRuke as pr
3 # klasa detektorRuke koja sadrži metode za detekciju ruke
4 class detektorRuke():
```



Slika 31: Pomicanje pokazivača po zaslonu

Nadalje, metode klase `detektorRuke` koje je bitno spomenuti su:

- `pronadjiRuku()` služi kako bi se detektirala ruka u određenom okviru te kako bi joj se dodijelili orijentiri koji bi se prikazali na ekranu.
- `pronadjiPoziciju()` služi za dohvaćanje položaja ruke, odnosno njene pozicije na slici.
- `prstPodignut()` služi za provjeravanje podignutog prsta, posebno za palac te posebno za ostale prste ruke.
- `pronadjiUdaljenost()` služi za pronalazak udaljenosti između kažiprsta i srednjeg prsta pomoću kojih se obavlja instrukcija "klika" na računalu.



Slika 32: Klik

Datoteka `virtualniMis.py` predstavlja primarnu datoteku u kojoj se odvijaju operacije poput kretnje pokazivača po zaslonu (slika 31.) te instrukcije klika gestikulacijom kažiprsta i srednjeg prsta (slika 32). Pomoću biblioteke `autopy` i njenih metoda `autopy.mouse.move()` i `autopy.mouse.click()` uspješno se izvršavaju instrukcije pomicanja pokazivača te klik (kod 3.) dohvaćajući trenutne informacije stečene gestikulacijom ruke.

### Isječak kôda 3: Pomicanje pokazivača i klik

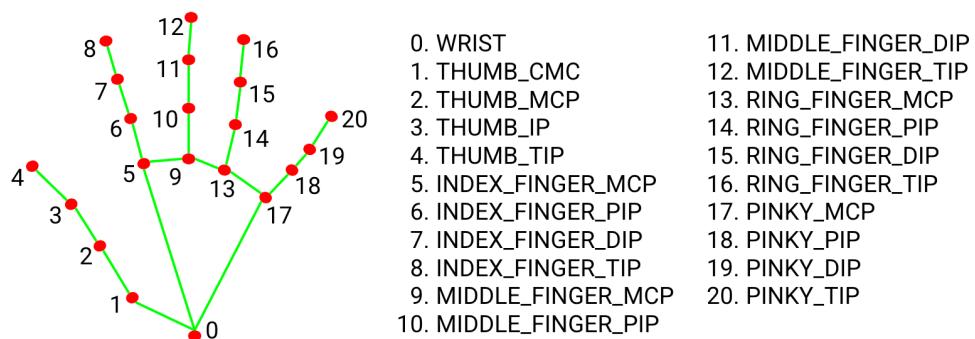
```

1 # Pomicanje pokazivača
2 autopsy.mouse.move(sirinaZaslona - trenutni_x, trenutni_y)
3 # Ako su kažiprst i srednji prst gore
4 if prsti[1] == 1 and prsti[2] == 1:
5     duljina, slika, linijaInfo = detektor.pronadjiUdaljenost(8, 12, slika)
6 # Ako su oba prsta jako blizu jedan drugome
7 if duljina < 40:
8     cv2.circle(slika, (linijaInfo[4], linijaInfo[5]), 15, (0, 255, 0), cv2.FILLED)
9 # klik
10 autopsy.mouse.click()

```

Kod implementacije prepoznavanja gesti (`prepoznavanjeGeste.py`), najbitniju ulogu preuzima model. Model izvodi preciznu lokalizaciju svih ključnih 21 točaka 3D koordinate šake kao na slici 33. i zglobo unutar detektiranih područja šake putem regresije, odnosno izravnog predviđanja koordinate. Model uči dosljedan interni prikaz položaja ruku i prikladan je čak i za djelomično vidljive ruke i samookluzije.

Kako bi se dobili istiniti podatci, ručno je označen veliki broj slika stvarnog svijeta s 3D koordinatama preko svih 21 označenih točaka, kao što je prikazano u nastavku (uzima se Z-vrijednost iz karte dubine slike, ako postoji po odgovarajućoj koordinati). Kako bi se bolje prikazali mogući položaji ruku i pružili dodatni provjeru nad prirodom geometrije ruku, renderira se visokokvalitetni sintetički model ruke preko različitih pozadina i mapira ga se na odgovarajuće 3D koordinate (Slika 34).



Slika 33: 21 točka šake; prema: [26]

Jedna od najbitnijih stvari za ovu implementaciju je inicijalizacija MediaPipe-a. (kod 4). MediaPipeov modul `mp.solution.hands` sadrži algoritam za prepoznavanje ruku. Stoga se prvo stvori objekt i pohrani ga u `mpHands`. Pomoću metode `mpHands.Hands` se konfigurira model. Prvi parametar je `max_num_hands`, što znači da će model detektirati maksimalan broj ruku u okviru. MediaPipe može detektirati više ruku u jednom kadru, ali za sada je stavljen da se detektira samo jedna po jedna ruku. `Mp.solutions.drawing_utils` odraduje posao otkrivanja ključnih točki, tako da ih ne moramo crtati ručno.

#### Isječak kôda 4: Inicijalizacija MediaPipea

```

1 # Inicijalizacija MediaPipe-a.
2 mpHands = mp.solutions.hands
3 hands = mpHands.Hands(max_num_hands=1, min_detection_confidence=0.7)
4 mpDraw = mp.solutions.drawing_utils

```

Pomoću funkcije `load_model` se učitava unaprijed obučeni TensorFlow model (kod 5). Datoteka `geste.imena` sadrži nazine klasa gesta. Prvo se ta datoteka otvoru pomoću Python ugrađene funkcije `open()`, a nakon toga se koristi funkciju `read()` za čitanje datoteke.

### Isječak kôda 5: Inicijalizacija TensorFlowa

```

1 # Učitavanje modela koji prepoznaće gestikulacije.
2 model = load_model('model')
3 # Učitavanje biblioteke naziva gestikulacija.
4 f = open('geste.imena', 'r')
5 imenaKlase = f.read().split('\n')
6 f.close()

```

MediaPipe obrađuje RGB slike, ali OpenCV čita slike u BGR formatu. Pomoću funkcije `cv2.cvtColor()` se sadržaj pretvara se u RGB format kako bi se vidjelo na zaslonu.

Zatim se koristi metoda `rezultat.multi_hand_landmarks` da provjeri jesu li prikazane ruke. Nakon toga, ponavlja se svaka detekcija i spremaju se koordinate na popis koji se zove orijentir (eng. *landmark*) (kod 6.). Ovdje se visina slike (y) i širina slike (x) množe s rezultatom jer model vraća normalizirani rezultat. To znači da je svaka vrijednost u rezultatu između 0 i 1. Pomoću funkcije `mpDraw.draw_landmarks()` prikazivaju se svi orijentiri u okviru. I na samom kraju, nakon pokazivanja ruke te detekcije geste, ostaje još samo prikazati dobiveni rezultat na zaslon pomoću metode `cv2.putText()`.

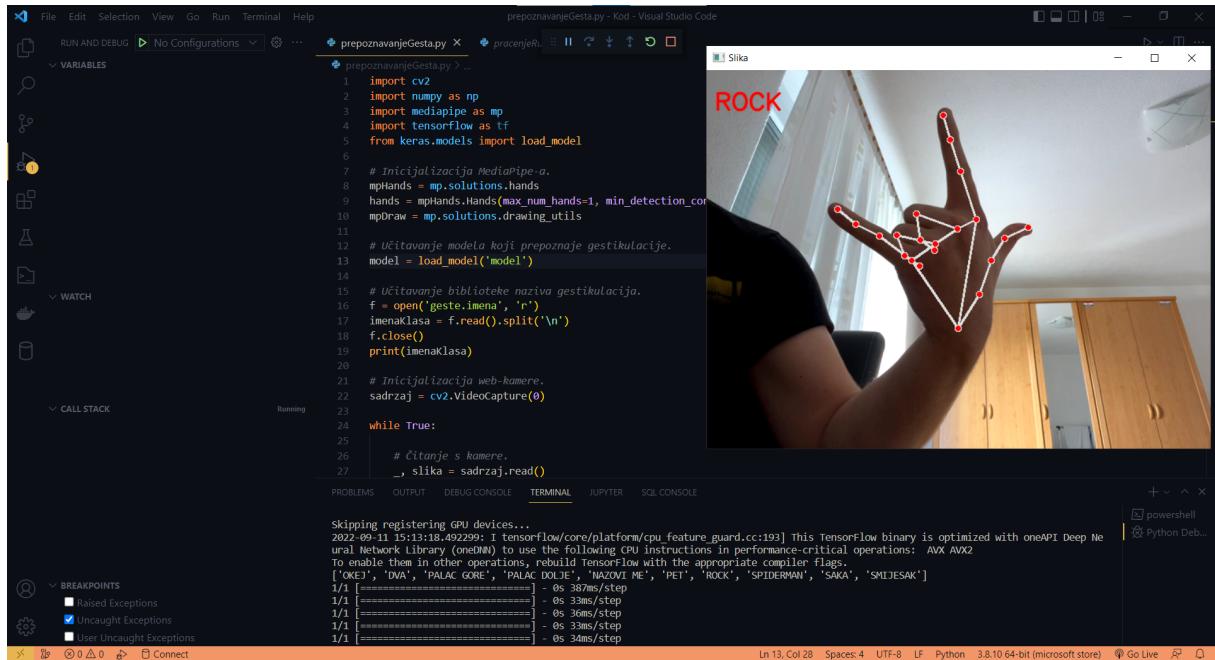
### Isječak kôda 6: Prepoznavanje geste

```

1 if rezultat.multi_hand_landmarks:
2     landmarks = []
3     for handslms in rezultat.multi_hand_landmarks:
4         for lm in handslms.landmark:
5             #print(id, lm)
6             lmx = int(lm.x * x)
7             lmy = int(lm.y * y)
8
9             landmarks.append([lmx, lmy])
10
11     # Crtanje orijentira na prikazu.
12     mpDraw.draw_landmarks(slika, handslms, mpHands.HAND_CONNECTIONS)
13
14     # Predikcija gestikulacije.
15     predikcija = model.predict([landmarks])
16     # print(predikcija)
17     klasaID = np.argmax(predikcija)
18     imeKlase = imenaKlase[klasaID]
19
20 # Prikazivanje predikcije.
21 cv2.putText(slika, imeKlase, (10, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2,
22 cv2.LINE_AA)
23 cv2.imshow("Slika", slika)

```

U ovim implementacijama prepoznavanja geste i virtualnog miša, pokazano je kako se OpenCV i Python koriste kako bi se detektirale neke od gestikulacija ruke koristeći modele koji se obučavaju da bi mogli davati ispravan rezultat. Pokazano je i korištenje biblioteka poput MediaPipe i TensorFlow pomoću kojih ne bi bilo izvedivo napraviti efikasan algoritam za ovakve vrste implementacija.



Slika 34: Prepoznavanje geste

## 5. Zaključak

Računalni vid je popularna tema u radovima o novoj tehnologiji. Drugačiji pristup korištenju podataka je ono što ovu tehnologiju čini drugačijom. Ogromne količine podataka koje se svakodnevno stvaraju, za koje neki ljudi misle da su prokletstvo današnjih generacija, zapravo se koriste za našu dobrobit – podaci mogu naučiti računala da vide i razumiju objekte. Ova tehnologija također pokazuje važan korak koji naša civilizacija čini prema stvaranju umjetne inteligencije koja može biti sofisticirana poput ljudske.

U radu je prikazano kako u teoriji računalo može prepoznati postoji li određena vrsta objekta na slici, poput prometnog znaka, ljudskog lica, tumora na medicinskoj slici itd. Počevši od samih začetaka ove grane znanosti, obrade slike, razvoja isto tako nove grane umjetne inteligencije dubokog učenja pa sve do primjena njegovih koncepata u računalnom vidu. Primjerice, kada se putem sigurnosnih kamera traži određena osoba, trenutno je moguće suziti izbor na najmanje 10 posto osoba koje kamere detektiraju, što ljudima olakšava posao. I u autonomnoj vožnji samo je pitanje vremena kada će se u potpunosti moći prepustiti vožnja autonomnim vozilima bez ugrožavanja ljudskih života i okoliša.

Upravo ta tehnologija (rezolucija kamere, brzina računala), sve je bliža proizvodnji potpuno sigurnih vozila ili prijevoznih sredstava koji ne ugrožavaju živote civila, već samo otklanjaju prijetnje (opasnosti). Daljnji napredak može se postići primjenom kombiniranih tehnologija poput infracrvenih kamera te analizom ljudskih pokreta poput hodanja.

Predstavljene su i dvije implementacije, od kojih su kao dio praktičnog dijela rada, jedna implementacija predstavlja detekciju ruke te prikaz njenih gesti u riječi kao rezultat te druge na kojoj se pomoću također gestikulacije ruke, provodi postupak kretanje pokazivača po zaslonu te lijevog klika miša pokazivajući dva podignuta prsta.

Unatoč svom tom nedavnom napretku, koji je bio impresivan, još uvijek se ne nadzire konačno rješenje problema računalnog vida. Međutim, već postoji više zdravstvenih ustanova i poduzeća koja su pronašla načine za primjenu sustava računalnog vida, koje pokreće CNN, na probleme u stvarnom svijetu. A ovaj trend vjerojatno neće uskoro prestati.

# Popis literature

- [1] I. Goodfellow, Y. Bengio i A. Courville, *Deep Learning*. MIT Press, 2016., <http://www.deeplearningbook.org>.
- [2] A. Oppermann. „What is deep learning and how does it work?” Built In. (2.5.2022.), adresa: <https://builtin.com/machine-learning/what-is-deep-learning> (pogledano 20.8.2022.).
- [3] D. Fhel. „What is machine learning?” Towards Data Science. (24.10.2019.), adresa: <https://medium.com/@FhelDimaano/what-is-machine-learning-885aa35db58b> (pogledano 5.9.2022.).
- [4] A. Bonner. „The complete beginners guide to deep learning,” Towards Data Science. (13.1.2019.), adresa: <https://towardsdatascience.com/intro-to-deep-learning-c025efd92535> (pogledano 1.9.2022.).
- [5] A. Zayegh i N. Al Bassam, *Neural Network Principles and Applications*. IntechOpen, 2018.
- [6] N. Buduma i N. Lacascio, *Fundamentals of Deep Learning*, First Edition. United States of America: O'Reilly Media, Inc., 2017., 298 str., ISBN: 978-1-491-92561-4.
- [7] S. Šegvić i J. Krapac. „Konvolucijski modeli,” Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu. (), adresa: <http://www.zemris.fer.hr/~ssegvic/du/du2convnet.pdf> (pogledano 8.9.2022.).
- [8] A. Kumar. „Real-world applications of convolutional neural networks,” Data Analytics. (), adresa: <https://vitalflux.com/real-world-applications-of-convolutional-neural-networks/> (pogledano 4.9.2022.).
- [9] R. Szeliski, *Computer Vision - Algorithms and Applications, Second Edition*, 2nd ed. London; New York: Springer, 2022., 1232 str., ISBN: 9783030343712.
- [10] Wikipedia. „Computer vision,” Wikimedia Foundation, Inc. (2022.), adresa: [https://en.wikipedia.org/wiki/Computer\\_vision](https://en.wikipedia.org/wiki/Computer_vision) (pogledano 17.8.2022.).
- [11] M. A. Boden, *Mind As Machine: A History of Cognitive Science Two-Volume Set*, engleski. Oxford University Press, USA, 2006., 1680 str., ISBN: 9780199241446.
- [12] T. S. Huang, „Computer Vision: Evolution And Promise,” 1996.
- [13] Wikipedia. „Pyramid (image processing),” Wikimedia Foundation, Inc. (2022.), adresa: [https://en.wikipedia.org/wiki/Pyramid\\_\(image\\_processing\)](https://en.wikipedia.org/wiki/Pyramid_(image_processing)) (pogledano 13.9.2022.).

- [14] N. Kumar. „Digital image processing basics,” GeeksForGeeks. (2021.), adresa: <https://www.geeksforgeeks.org/digital-image-processing-basics/> (pogledano 23.8.2022.).
- [15] N. Babich. „What is computer vision and how does it work? an introduction,” Adobe. (28.7.2020.), adresa: <https://xd.adobe.com/ideas/principles/emerging-technology/what-is-computer-vision-how-does-it-work/> (pogledano 20.8.2022.).
- [16] E. Ceolini, C. Frenkel, S. B. Shrestha i dr., „Hand-Gesture Recognition Based on EMG and Event-Based Camera Sensor Fusion: A Benchmark in Neuromorphic Computing,” *Frontiers in Neuroscience*, sv. 14, 2020., ISSN: 1662-453X. DOI: 10.3389/fnins.2020.00637.
- [17] M. Oudah, A. Al-Naji i J. Chahl. „Hand gesture recognition based on computer vision: A review of techniques,” National Library of Medicine. (23.7.2020.), adresa: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8321080/> (pogledano 23.8.2022.).
- [18] P. Nguyen Huu i T. Phung Ngoc. „Hand gesture recognition algorithm using svm and hog model for control of robotic system,” Hindawi. (17.6.2021.), adresa: <https://doi.org/10.1155/2021/3986497> (pogledano 6.9.2022.).
- [19] Q. Zhang, S. Xiao, Z. Yu, H. Zheng i P. Wang. „Hand gesture recognition algorithm combining hand-type adaptive algorithm and effective-area ratio for efficient edge computing,” Sun Yat-sen University, Harbin University of Science i Technology. (23.12.2021.), adresa: <https://ui.adsabs.harvard.edu/abs/2021JEI....30f3026Z/abstract> (pogledano 6.9.2022.).
- [20] L. Peng i V. Gulshan. „Deep learning for detection of diabetic eye disease,” Google LLC. (29.11.2016.), adresa: <https://ai.googleblog.com/2016/11/deep-learning-for-detection-of-diabetic.html> (pogledano 26.8.2022.).
- [21] L. Peng i M. Stumpe. „Assisting pathologists in detecting cancer with deep learning,” Google LLC. (3.3.2017.), adresa: <https://ai.googleblog.com/2017/03/assisting-pathologists-in-detecting.html> (pogledano 3.9.2022.).
- [22] N.N. „Autopilot and full self-driving capability,” Tesla, Inc. (2019.), adresa: <https://www.tesla.com/support/autopilot> (pogledano 14.8.2022.).
- [23] saurabh409. „Everything happening in computer vision that you should know,” Analytics Vidhya. (30.6.2021.), adresa: <https://www.analyticsvidhya.com/blog/2021/06/everything-happening-in-computer-vision-that-you-should-know/> (pogledano 26.8.2022.).
- [24] N.N. „Computer vision in agriculture: Introduction to smart farming,” EasyFlow. (2021.), adresa: <https://easyflow.tech/computer-vision-in-agriculture-introduction-to-smart-farming/> (pogledano 12.9.2022.).
- [25] ——, „Installation,” Google LLC. (), adresa: [https://google.github.io/mediapipe/getting\\_started/install.html](https://google.github.io/mediapipe/getting_started/install.html) (pogledano 7.8.2022.).

- [26] J. Howse i J. Minichino, *Learning OpenCV 4 Computer Vision with Python 3 Third Edition*, 3rd ed. Birmingham: Packt Publishing, 2022., 495 str., ISBN: 978-1-78953-161-9.

# Popis slika

1.	Neuronska mreža i polutka mozga; prema: [1] . . . . .	2
2.	Potpodručja umjetne inteligencije; prema: [2] . . . . .	3
3.	Razlika dubokog i strojnog učenja; prema: [2] . . . . .	4
4.	Nadzirano učenje (lijevo) i nenadzirano učenje (desno); prema: [1] . . . . .	5
5.	Biološka neuronska mreža; prema: [2] . . . . .	6
6.	Umjetni neuron; prema: [5] . . . . .	6
7.	Arhitektura neuronske mreže; prema: [2] . . . . .	7
8.	Veze slojeva; prema: [2] . . . . .	8
9.	Širenje; prema: [2] . . . . .	8
10.	Arhitektura konvolucijske neuronske mreže; prema: [2] . . . . .	9
11.	Praktični primjer na temelju prepoznavanja životinje: bizona, goveda i medvjeda; prema: [7] . . . . .	10
12.	Slojevi konvolucije, poduzorkovanja i potpuno povezani sloj; prema: [7] . . . . .	10
13.	Primjer korištenja računalnog vida u stvarnom životu; prema: [9] . . . . .	11
14.	Vremenski raspored odabranih tema istraživanja u računalnom vidu; prema: [9] . . . . .	12
15.	Primjeri algoritama računalnog vida iz 1980-ih: (a) piramidalno spajanje, (b) oblik iz sjenčanja, (c) otkrivanje rubova, (d) fizički bazirani modeli, (e) površinska rekonstrukcija temeljena na regularizaciji, (f) prikupljanje podataka o rasponu; prema: [9] . . . . .	12
16.	Piramida slika; prema: [9] . . . . .	13
17.	Primjeri algoritama računalnog vida iz 2000-ih: (a) renderiranje, (b) modeliranje, (c) interaktivno preslikavanje tonova, (d) sinteza teksture, (e) prepoznavanje, (f) regionalno prepoznavanje; prema: [9] . . . . .	14
18.	Komponente procesa oblikovanja slike: (a) perspektivna projekcija, (b) raspršenje svjetlosti pri udaru o površinu, (c) optika leća, (d) Bayerov niz filtera u boji; prema: [9] . . . . .	16

19.	Vrijednosti boja pojedinačnih piksela; prema: [15] . . . . .	17
20.	Područja koja djeluju ovisno o slici ili vrsti informacije . . . . .	17
21.	Razne vrste prepoznavanja: (a) prepoznavanje lica sa slikovnim strukturama, (b) prepoznavanje instanci (poznatog objekta), (c) prepoznavanje lica u stvarnom vremenu, (d) prepoznavanje temeljeno na značajkama, (e) segmentacija instance pomoću <i>Mask R-CNN</i> , (f) procjena poze; prema: [9] . . . . .	18
22.	Detekcija okoline videokamerom; prema: [9] . . . . .	19
23.	<i>YOLO</i> , algoritam za otkrivanje objekata; prema: [9] . . . . .	19
24.	Dijagram toka prepoznavanja . . . . .	20
25.	Geste ruke; prema: [19] . . . . .	21
26.	Dijagram mjerenja duljine dlana; prema: [19] . . . . .	22
27.	Primjeri fotografija retinalnog fundusa. Slika s lijeve strane je zdrava mrežnica (A), dok je slika s desne strane mrežnica s dijabetičkom retinopatijom (B) zbog brojnih krvarenja (crvenih mrlja); prema: [20] . . . . .	23
28.	Krupni plan biopsije limfnog čvora; prema: [21] . . . . .	23
29.	Unutrašnjost Tesla automobila te detekcija vanjskih objekata u prometu; prema: [22] . . . . .	24
30.	OpenCV i Python; prema: [26] . . . . .	26
31.	Pomicanje pokazivača po zaslonu . . . . .	27
32.	Klik . . . . .	28
33.	21 točka šake; prema: [26] . . . . .	29
34.	Prepoznavanje geste . . . . .	31

# Popis isječaka koda

1.	Biblioteke . . . . .	27
2.	Klasa detektorRuke . . . . .	27
3.	Pomicanje pokazivača i klik . . . . .	28
4.	Inicijalizacija MediaPipea . . . . .	29
5.	Inicijalizacija TensorFlowa . . . . .	30
6.	Prepoznavanje geste . . . . .	30

## **Prilozi**

# 1. Programski kôd

Poveznica na *GitHub* platformu na kojoj se nalaze programski kôdovi. (Potrebno je kliknuti na ovaj tekst.)