

# Detekcija ključnih točaka ruku i prepoznavanje gesti u videosekvenci

---

Čirko, Fran

Undergraduate thesis / Završni rad

2024

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:168:304032>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-15**



*Repository / Repozitorij:*

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 974

**DETEKCIJA KLJUČNIH TOČKA RUKU I PREPOZNAVANJE  
GESTI U VIDEOSEKVENCI**

Fran Čirko

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 974

**DETEKCIJA KLJUČNIH TOČKA RUKU I PREPOZNAVANJE  
GESTI U VIDEOSEKVENCI**

Fran Čirko

Zagreb, lipanj 2023.

## ZAVRŠNI ZADATAK br. 974

Pristupnik: **Fran Čirko (0036524830)**  
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo  
Modul: Računarstvo  
Mentorica: prof. dr. sc. Sonja Grgić

Zadatak: **Detekcija ključnih točaka ruku i prepoznavanje gesti u videosekvenci**

### Opis zadatka:

Detekcija pokreta ruku u videosekvencama i prepoznavanje gesti ima brojne primjene koje uključuju virtualna okruženja, robotiku, prepoznavanje znakovnog jezika i slično. Cilj ovog rada je napraviti programsko rješenje za prepoznavanje gesti ruku koje uključuje detekciju ruku, prepoznavanje lijeve i desne ruke, detekciju ključnih točaka ruke i prepoznavanje unaprijed definiranog ograničenog skupa gesti ruku. Detektirati i lokalizirati ključne točke ruku kao što su vrhovi prstiju, zglobovi na prstima, zglobovi kojima se prsti vežu sa šakom i zglob kojim je šaka povezana s podlakticom. Prepoznavanje gesti ruku provesti uz pomoć unaprijed uvježbanog modela strojnog učenja i detektiranih ključnih točaka ruku. Evaluirati programsko rješenje te prikazati i analizirati postignute rezultate.

Rok za predaju rada: 9. lipnja 2023.



# Sadržaj

Uvod .....	1
1. Prepoznavanje gestikulacija ruku .....	2
1.1. Problem prepoznavanja gestikulacija ruku.....	2
1.2. Metode prepoznavanja gesti ruku.....	3
1.3. Rješenja za prepoznavanje gesti ruku.....	4
1.4. Evaluacija prepoznavanja gesti ruku .....	5
2. Razvojno okruženje i podaci .....	7
2.1. Razvojno okruženje .....	7
2.2. Korištene biblioteke.....	7
2.3. Prikupljanje baze .....	8
2.4. Baza slika.....	8
2.5. Model.....	9
3. Implementacija prepoznavanja gestikulacija ruku .....	11
3.1. Akvizicija podataka .....	11
3.2. Detekcija ruku.....	12
3.3. Praćenje ruku .....	13
3.4. Izdvajanje značajki .....	14
3.5. Klasifikacija gesta.....	16
3.6. Intrepretacija i obrada rezultata .....	17
3.7. Rezultati.....	18
4. Primjena prepoznavanja gestikulacija ruku .....	20
4.1. Primjeri primjene .....	20
4.1.1. Robotska interakcija .....	20
4.1.2. Interakcija s računalom.....	21

4.1.3.	Virtualna i proširena stvarnost.....	21
4.1.4.	Edukacija i rehabilitacija .....	23
4.1.5.	Sigurnost i nadzor .....	24
	Zaključak .....	25
	Literatura .....	26
	Sažetak.....	27
	Summary.....	28

# Uvod

Detekcija pokreta ruku u videosekvencama i prepoznavanje gesti postali su iznimno zanimljiva područja istraživanja u računalnom vidu i obradi slika. Ovi problemi imaju široku primjenu u različitim područjima, uključujući virtualna okruženja, prepoznavanje znakovnog jezika, interakciju čovjeka i računalnih sustava te robotiku. Prepoznavanje gesti ruku omogućuje intuitivno i prirodno upravljanje raznim digitalnim sustavima, kao što su virtualna stvarnost, igre ili robotske ruke. Također, omogućuje komunikaciju osoba s oštećenim govorom putem znakovnog jezika.

Cilj ovog završnog rada je razviti programsko rješenje za prepoznavanje gesti ruku u videosekvencama. Ova se problematika sastoji od nekoliko ključnih dijelova, uključujući detekciju ruku, prepoznavanje lijeve i desne ruke, lokalizaciju ključnih točaka ruku te prepoznavanje unaprijed definiranog ograničenog skupa gesti ruku. Ključne točke ruku ili kostur ruku obuhvaćaju vrhve prstiju, zglobove na prstima, zglobove s kojima se prsti vežu sa šakom te zglob kojim je šaka povezana s podlakticom. Prepoznavanje gesti ruku temelji se na kombinaciji detektiranih ključnih točaka ruku i unaprijed uvježbanog modela strojnog učenja.

U ovom radu pristupit ćemo problemu prepoznavanja gesti ruku koristeći pristup temeljen na dubokom učenju. Želimo ostvariti preciznu i strogu detekciju ključnih točaka ruku te prepoznavanje gesti s visokom točnošću. Uz razvoj programskog rješenja, također ćemo evaluirati ostvarene rezultate te analizirati performanse provođenjem testova.



# 1. Prepoznavanje gestikulacija ruku

Prepoznavanje gestikulacija ruku je jako zanimljivo i dinamično područje istraživanja u strojnom učenju i računalnom vidu. Postupak prepoznavanja gesti ruku omogućuje računalima i drugim uređajima da interpretiraju i razumiju prirodne pokrete ljudskih ruku zbog komunikacije s korisnicima. Prepoznavanje gesti ruku ima široku primjenu u različitim područjima od kojih su neki virtualna stvarnost, interakcija s računalnim igrama, robotski sustavi, upravljanje uređajima pa čak i rehabilitacijske terapije. Za opisivanje problema, metoda i potencijalnih rješenja pozivat ćemo se na istraživanje An Overview of Hand Gestures Recognition System Techniques<sup>1</sup> [1] iz 2015. godine.

## 1.1. Problem prepoznavanja gestikulacija ruku

Problem prepoznavanja gesti ruku možemo opisati kao prepoznavanje i klasifikaciju različitih gesti koje korisnik izvodi rukama. Pravi izazov leži u analizi i interpretaciji oblika, orijentacije, pokreta i pozicija ruku kako bi se razumijela korisnikova namjera. Prepoznavanje gestikulacija ruku se suočava s skupom različitih izazova, uključujući varijabilnost izvođenja gesti, raznolikost okruženja, procjenu pokreta i potrebu za obradom u stvarnom vremenu.

Precizna detekcija i praćene ruku u različitim scenarijima predstavlja pravi izazov zbog varijacije raznih faktora kao što su pozadina, osvjetljenje, veličina ruku i skrivanje dijelova ruku. Također, razlikovanje između različitih gesti može biti teško zbog sličnosti nekih pokreta kao na primjer, približavanje prstiju i širenje prstiju mogu izgledati slično, ali imaju različite namjere.

---

<sup>1</sup> <https://iopscience.iop.org/article/10.1088/1757-899X/99/1/012012>

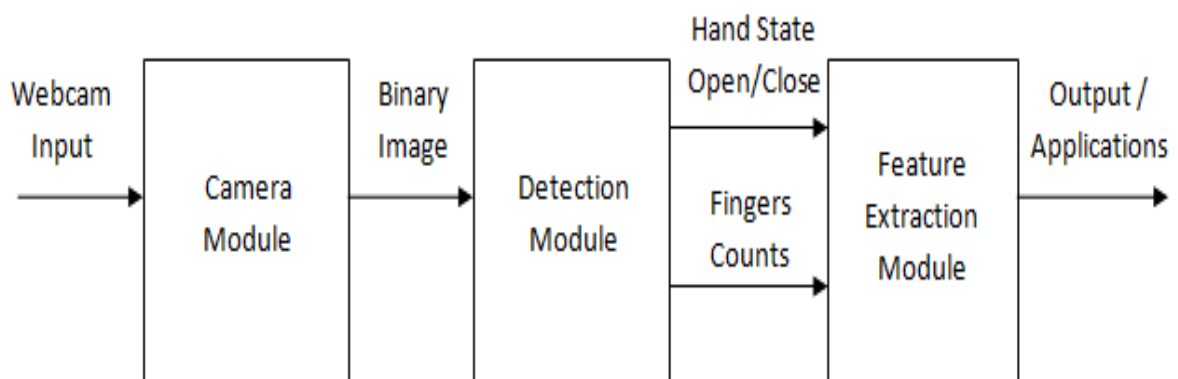
## 1.2. Metode prepoznavanja gesti ruku

Postoje nekoliko različitih metoda za prepoznavanje gesti ruku, a razvoj dubokog učenja doveo je do značajnih napredaka u ovom području. U radu dostupnom na linku<sup>2</sup> [1] predstavljen je pristup koji koristi duboko učenje za prepoznavanje gesti ruku iz videa. Ova metoda kombinira duboku konvolucijsku neuronsku mrežu (eng. Convolutional Neural Network, CNN) s arhitekturom Long Short-Term Memory (LSTM) rekurentne neuronske mreže kako bi se naučio kontekstualni uzorak i dinamičke informacije gesti.

Duboko učenje omogućava automatsko učenje značajki iz sirovih, neobrađenih podataka, što ga čini pogodnim za prepoznavanje gesti ruku, Konvolucijske neuronske mreže mogu naučiti reprezentacije gesti iz slika ili videozapisa, dok LSTM mreže mogu modelirati kontekstualne veze između različitih vremenskih koraka u gestama.

Osim dubokog učenja postoje i druge metode za prepoznavanje gestikulacija ruku, kao što su temeljene na izvlačenju značajki ili pravila. Primjeri takvih metoda uključuje izvlačenje oblika i položaja ruku, praćenje pokreta ili korištenje skupa pravila temeljenih na heuristikama. Međutim, te metode često zahtijevaju ručno definirane značajke i pravila te nisu uvijek dovoljno robusne za raznolike scenarije.

Primjer izgleda modula sustava za prepoznavanje ruku možemo vidjeti na slici (Slika 1.1).



Slika 1.1 Sustav prepoznavanja gesti ruku<sup>3</sup> [1]

---

<sup>2</sup> <https://iopscience.iop.org/article/10.1088/1757-899X/99/1/012012>

<sup>3</sup> <https://iopscience.iop.org/article/10.1088/1757-899X/99/1/012012/pdf>

### 1.3. Rješenja za prepoznavanje gesti ruku

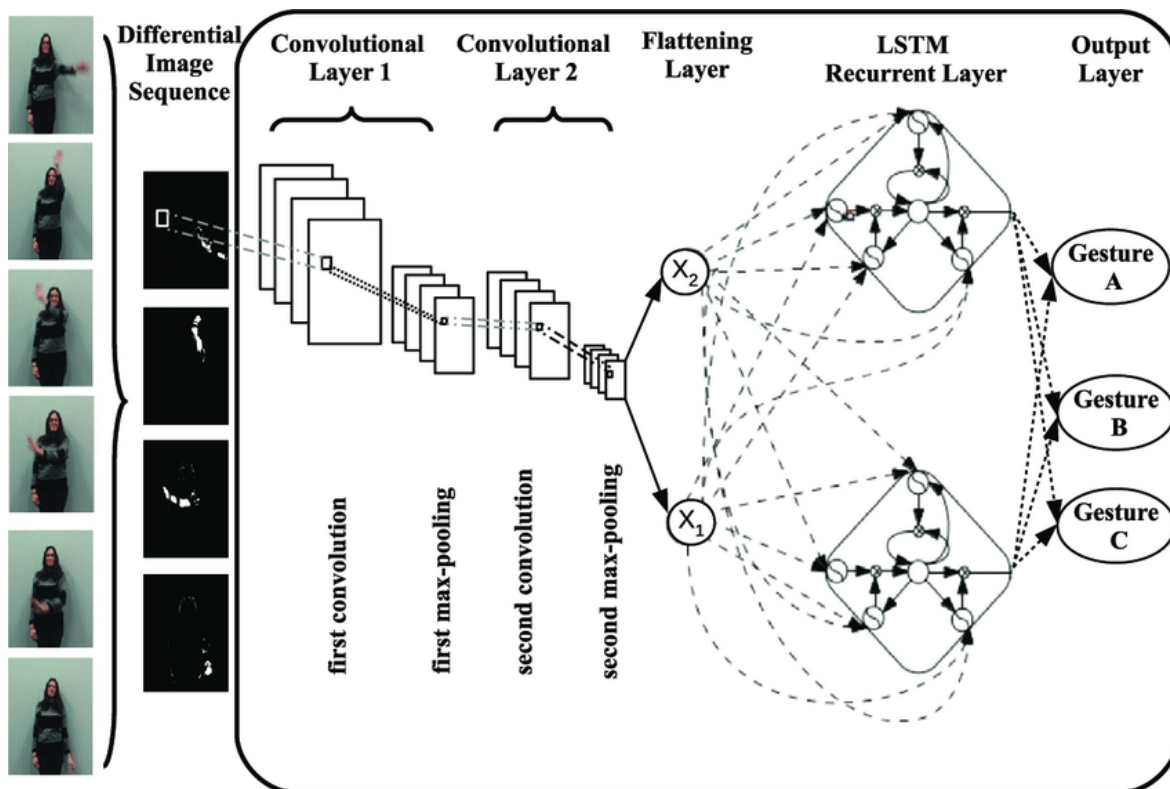
Postoji nekoliko rješenja za prepoznavanje gesti ruku temeljenih na dubokom učenju. U radu<sup>4</sup> [1] autori predstavljaju CNN-LSTM arhitekturu koja se pokazala vrlo uspješnom u prepoznavanju gesti ruku iz videa. Ova arhitektura kombinira snagu konvolucijskih neuronskih mreža u izvlačenju lokalnih značajki s mogućnošću LSTM mreže da modelira temporalne veze između gesti.

CNN-LSTM arhitektura (Slika 1.2) sastoji se od nekoliko slojeva konvolucijskih i LSTM slojeva. Konvolucijski slojevi koriste se za izlučivanje značajki iz slika ili videozapisa gesti, dok LSTM slojevi modeliraju temporalne veze između tih značajki. Duboko učenje omogućuje automatsko učenje reprezentacija značajki iz podataka i prilagodbu modela različitim gestama.

Druga rješenja za prepoznavanje gesti ruku uključuju korištenje drugih dubokih arhitektura, poput rekurentnih neuronskih mreža (RNN), transformera ili kombinacija više različitih modela. Također, postoji istraživanje primjene generativnih modela, kao što su generativno suparničke mreže (eng. Generative Adversarial Networks, GAN), za stvaranje i prepoznavanje gesti ruku.

---

<sup>4</sup> <https://iopscience.iop.org/article/10.1088/1757-899X/99/1/012012>



Slika 1.2 CNN-LSTM arhitektura<sup>5</sup>

## 1.4. Evaluacija prepoznavanja gesti ruku

Evaluacija prepoznavanja gesti ruku ključna je za utvrđivanje točnosti (Slika 1.3) i učinkovitosti sustava. U radu<sup>6</sup> [1] autori koriste različite metrike evaluacije, kao što su preciznost, osjetljivost, specifičnost i F-mjera. Preciznost mjeri postotak točno prepoznatih gesti, osjetljivost i specifičnost mjere sposobnost sustava da ispravno prepozna pozitivne i negativne geste, a F-mjera kombinira te dvije metrike.

Evaluacija se obično provodi na različitim skupovima podataka, prikupljenim u stvarnim ili simuliranim scenarijima. Skupovi podataka sadrže snimke gesti ruku izvedene od strane različitih korisnika i u različitim okruženjima. Pomoću tih skupova podataka, mogu se izračunati performanse sustava prema definiranim metrikama evaluacije.

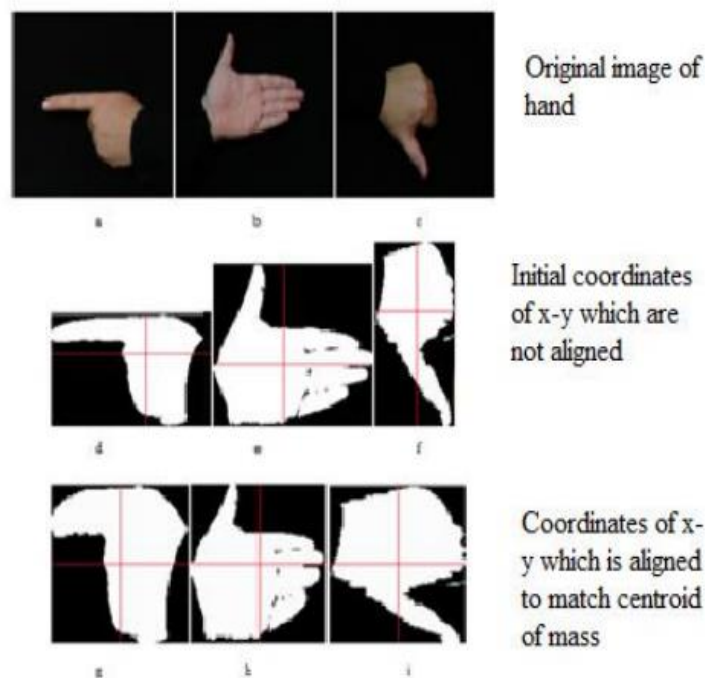
<sup>5</sup> [https://www.researchgate.net/figure/The-CNNLSTM-architecture-simplified-for-visualization-purposes\\_fig1\\_316654191](https://www.researchgate.net/figure/The-CNNLSTM-architecture-simplified-for-visualization-purposes_fig1_316654191)

<sup>6</sup> <https://iopscience.iop.org/article/10.1088/1757-899X/99/1/012012>

Dodatno, evaluacija uključuje usporedbu s drugim metodama prepoznavanja gesti ruku iz literature. To omogućava procjenu prednosti i nedostataka pristupa koji je predložen u radu [1]. Evaluacija također može uključivati analizu vremena izvođenja i resursnih zahtjeva algoritma prepoznavanja gesti ruku kako bi se procijenila njegova primjenjivost u stvarnom vremenu.

Navedeni rad pruža samo jedan primjer pristupa prepoznavanju gesti ruku. U literaturi postoji velik broj drugih radova koji istražuju različite metode, arhitekture i algoritme za rješavanje ovog problema. Stalni napredak u području dubokog učenja i računalnog vida pruža nove mogućnosti i otvara put za daljnje istraživanje i razvoj u prepoznavanju gesti ruku.

$$Recognition (\%) = \frac{\text{number of matched features}}{\text{total number of blocks}} \times 100\% \quad (1)$$



Slika 1.3 Računanje točnosti <sup>7</sup>[1]: a, b, c: Originalna fotografija, d, e, f: fotografija nakon segmentacije, g, h, i: fotografija nakon skaliranja na normaliziranu veličinu

<sup>7</sup> <https://iopscience.iop.org/article/10.1088/1757-899X/99/1/012012/pdf>

## 2. Razvojno okruženje i podaci

### 2.1. Razvojno okruženje

Program ovog završnog rada napisan je u programskom jeziku Python unutar razvojnog okruženja PyCharm<sup>8</sup>. PyCharm je razvojno okruženje tvrtke JetBrains te je dostupan u nekoliko verzija: besplatna zajednička verzija (Community Edition) i profesionalna verzija (Professional Edition) koja ima dodatne značajke. Za izradu programa korišten je PyCharm Community Edition. Za izradu rada korišten je PyCharm koji pruža niz alata i funkcionalnosti za olakšavanje procesa razvoja softvera, ponajprije jednostavna instalacija i deinstalacija biblioteka te testiranje. Radi toga, isječci koda koji se pojave u nastavku rada bit će isječci napisani u aplikaciji PyCharm.

### 2.2. Korištene biblioteke

Zbog praktičnosti u izradi rada korišteno je nekoliko različitih biblioteka u Pythonu. *Cvzone* koji služi za detekciju, praćenje i prepoznavanje ruku. *OpenCV* biblioteka sa kraticom *cv2* ima mnogo funkcionalnosti i algoritama za manipulaciju slikama i videom. *Mediapipe*, korisna za obradu multimedijjskih podataka, uključujući video i sliku. *TensorFlow*, koji se koristi za izgradnju i treniranje različitih modela umjetne inteligencije. *NumPy* kratice *np* koja pomaže u radu s višedimenzionalnim nizovima i matricama, meni pomaže oko različitih matematičkih operacija vezanih uz veličine i oblike slika. *Math* koja pripada standardnoj biblioteci Pythona, korisna za izračunavanja omjera visine i širine slike te zaokruživanje brojeva. Te na kraju *time*, također standardna biblioteka Pythona koja pruža funkcionalnosti vezane uz vrijeme, u ovom kodu iskorištena prilikom automatskog imenovanja slika spremanih u bazu.

---

<sup>8</sup> <https://www.jetbrains.com/pycharm/download/#section=windows>

## 2.3. Prikupljanje baze

Za izradu modela za prepoznavanje određenih gesti ruku koristili su se podatci koji su prikupljeni uz pomoć koda. U kod je implementirana mogućnost uzimanja slika pritiskom na gumb za uzimanje i spremanje slika (Slika 2.1).

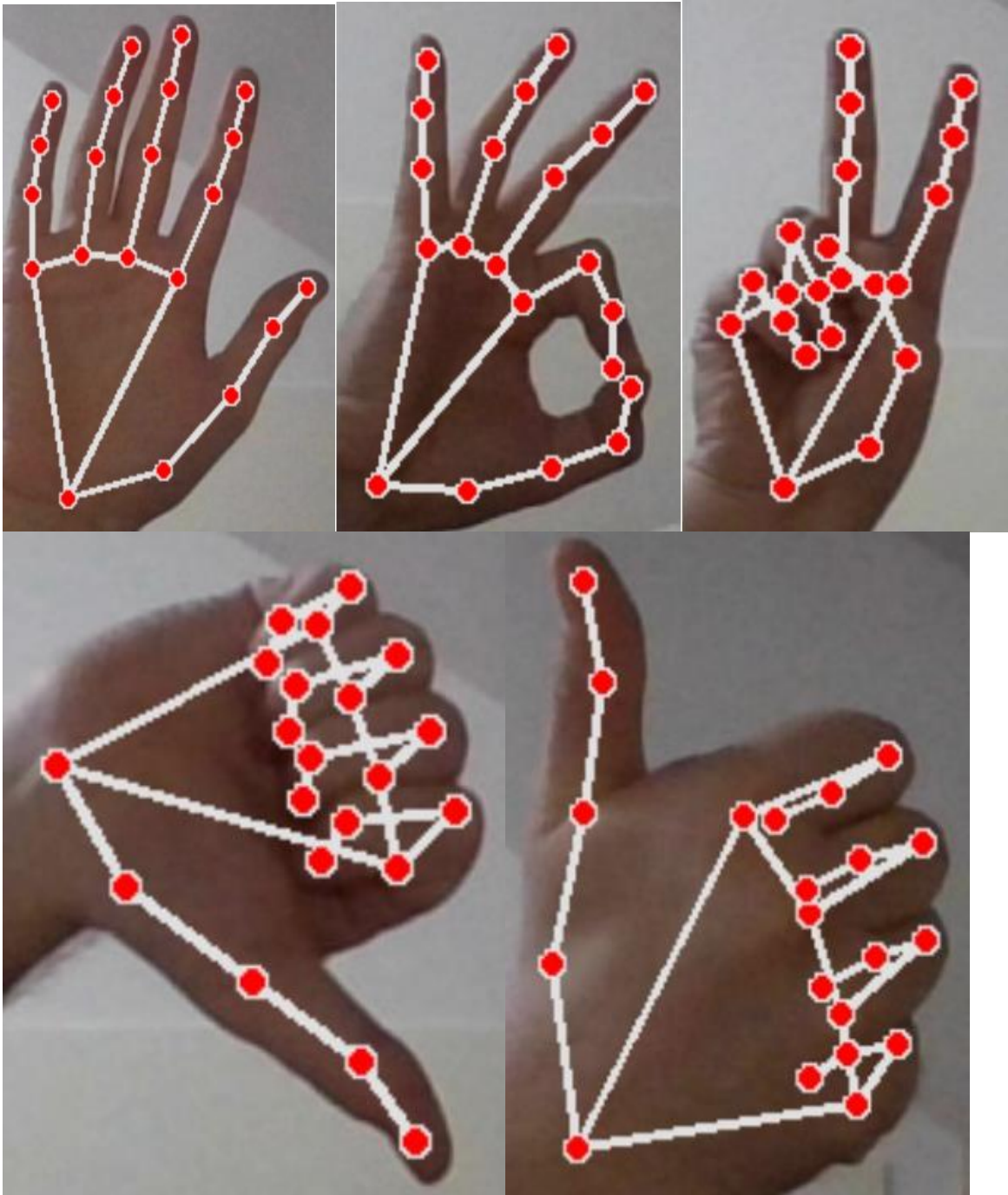
```
key = cv2.waitKey(1)
if key == ord("s"):
    counter += 1
    cv2.imwrite(f'{folder}/Image_{time.time()}.jpg', img_white)
    print(counter)
```

Slika 2.1 Uzimanje i spremanje slika

Spremanjem slika stvaramo bazu koja će kasnije biti korištena u izradi modela za treniranje za prepoznavanje gestikulacija ruku.

## 2.4. Baza slika

Sve prikupljene slike spremaju se u bazu slika. Svaka slika sprema se u određeni direktorij kojeg odredimo prije pokretanja programa. Osobno sam prikupio preko 2500 (dvije tisuće i petsto) slika. U radu je implementirano prepoznavanje za pet različitih gestikulacija redom na slikama 'HelloSign', 'OkSign', 'PeaceSign', 'ThumbsDown', 'ThumbsUp' (Slika 2.2). Program je napisan tako da se prilikom svakog pokretanja može nadodati još slike te se mogu istrenirati nove gestikulacije, a ne samo koje su trenutno implementirane u radu.



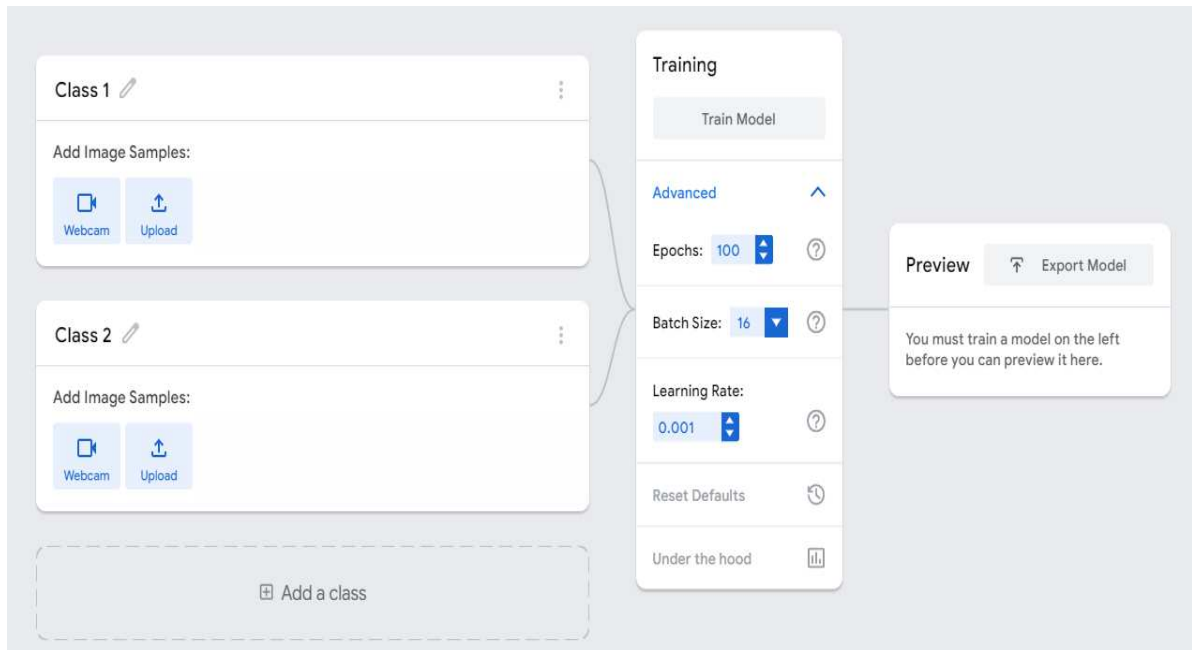
Slika 2.2 Primjer slika iz baze

## 2.5. Model

Za izradu modela korištene su usluge postojećeg web programa, Googleovog Teachable Machine. Google Teachable Machine (Slika 2.3) jest alat na web-u koji omogućuje brzo i jednostavno kreiranje modela strojnog učenja te je dostupan svima. Program funkcionira na način da se za 'klase' koje program treba prepoznati učitaju fotografije iz baze. U sklopu



ovog završnog rada učitano je otprilike 500 slika za svaku klasu. Kada su slike učitane, onda se model treba istrenirati, a za to je postavljeno da svaka slika prođe kroz model za treniranje barem 100 puta kako bi se dobili što točniji rezultati. Na kraju se dobiveni keras model izvozi te se u kodu kasnije to implementira.



Slika 2.3 Google Teachable Machine<sup>9</sup>

---

<sup>9</sup> <https://teachablemachine.withgoogle.com/train/image>

## 3. Implementacija prepoznavanja gestikulacija ruku

Implementacija prepoznavanja gestikulacija ruku uključuje primjenu računalnog vida i strojnog učenja kako bi se identificirale i interpretirale geste iz snimljenih podataka ili stvarno vremenskih podataka o rukama. U ovom poglavlju opisujem glavne korake i tehnike koje se koriste u implementaciji.

### 3.1. Akvizicija podataka

Prvi bitni korak jest prikupljanje podataka o rukama, a to može uključivati korištenje RGB ili dubinskih kamera, senzora pokreta ili drugih uređaja za snimanje ruku. Podaci se mogu snimati u stvarnom vremenu ili se može koristiti skup unaprijed snimljenih podataka.

U sklopu ovog rada, akvizicija podataka odvija se korištenjem `cv2.VideoCapture` (Slika 3.1) funkcije za otvaranje videozapisa s kamere. To omogućuje snimanje videa u stvarnom vremenu. Također implementirana je opciju prikupljanja podataka u kojoj korisnik može unijeti naziv mape u koju će se spremati snimljeni podaci. Isječak koda u kojem se vidi kako je implementirano prikupljanje podataka može se vidjeti na slici (Slika 3.2).

```
import cv2  
  
cap = cv2.VideoCapture(0)
```

Slika 3.1 Otvaranje videozapisa kamere

```

choice = input("Choose an option (test/collect): ")

if choice == "collect":
    folder = input("Enter the folder name: ")
    counter = 0

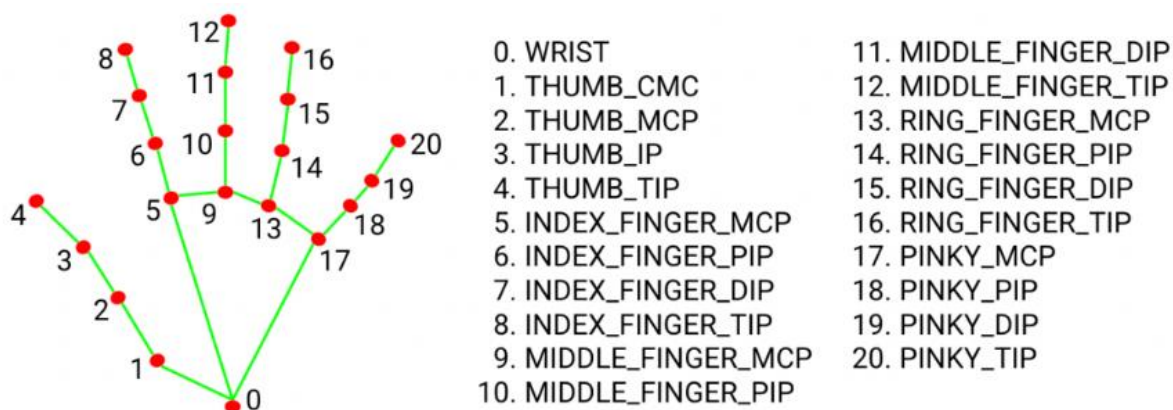
key = cv2.waitKey(1)
if key == ord("s"):
    counter += 1
    cv2.imwrite(f'{folder}/Image_{time.time()}.jpg', img_white)
    print(counter)

```

Slika 3.2 Prikupljanje podataka

## 3.2. Detekcija ruku

Nakon akvizicije podataka primjenjuje se detekcija ruku kako bi se identificirale regije koje sadrže ruke. Za to se koriste tehnike poput detekcije objekata, detekcija rubova ili duboko učenje za lokalizaciju i izdvajanje ruku iz slike. Na slici (Slika 3.3) su prikazane ključne točke ruke koje se prepoznaju prilikom pokretanja programa.



Slika 3.3 Ključne točke ruke<sup>10</sup>

Za detekciju ruku na slici koristile su se funkcije *findHands* (Slika 3.5) i *HandDetector* (Slika 3.4) iz *cvzone* biblioteke. Ova tehnika detektiranja ruku može se opisati kao kombinacija detekcije objekata i dubokog učenja za izdvajanje ruku iz slike i lokalizaciju.

<sup>10</sup> [https://developers.google.com/mediapipe/solutions/vision/gesture\\_recognizer](https://developers.google.com/mediapipe/solutions/vision/gesture_recognizer)

Detektor ruku prima ulazni kadar videozapisa i koristi unaprijed obučeni model za pronalaženje regija koje sadrže ruke.

```
from cvzone.HandTrackingModule import HandDetector  
  
cap = cv2.VideoCapture(0)  
detector = HandDetector(maxHands=1)
```

Slika 3.4 Implementacija HandDetectora

```
while True:  
    success, img = cap.read()  
    hands, img = detector.findHands(img)  
    img_output = img.copy()
```

Slika 3.5 Korištenje funkcije za pronalazak ruku

### 3.3. Praćenje ruku

Kada se sigurno mogu detektirati ruke koristi se korak praćenja ruku kako bi se pratila pozicija kretanja ruku tijekom vremena. Ono nam omogućuje kontinuirano praćenje ruku i u onim trenucima kada su ruke djelomično sakrivene ili kada mijenjaju položaj.

U kodu praćenje ruku implementirano je unutar *while* petlje. U petlji se koristi metoda *findHands()* kako bi se dobila informacija o trenutnom položaju i obliku ruke. Kada se dobiju rezultati detekcije ruke ti se podatci spremaju u objekt *hand*. Iz njega se kasnije dohvaćaju koordinate *okvira (bbox)* ruke kao *x*, *y*, *w* i *h* gdje su *x* i *y* koordinate gornjeg lijevog kuta, a *w* i *h* su širina i visina okvira. Pomoću toga i dimenzija ruke dalje je moguće obraditi sliku ruke i izdvojiti značajke i izvršiti klasifikaciju gesti. Prikaz implementacije nalazi se na slici (Slika 3.6).

```

while True:
    success, img = cap.read()
    hands, img = detector.findHands(img)
    img_output = img.copy()

    if hands:
        hand = hands[0]
        x, y, w, h = hand['bbox']

```

Slika 3.6 Praćenje ruku

### 3.4. Izdvajanje značajki

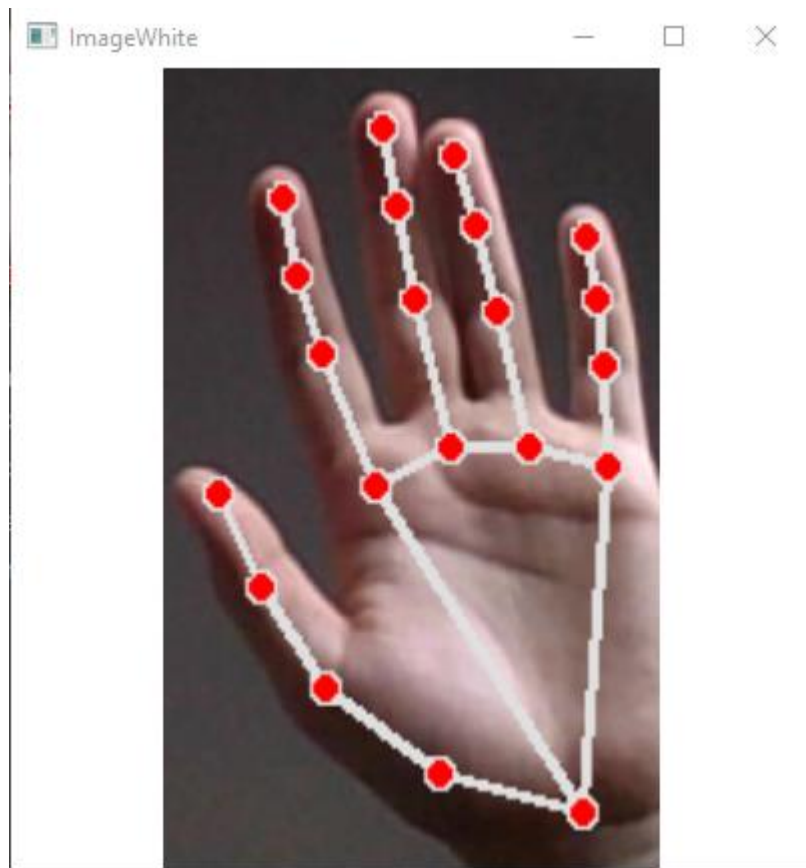
Sljedeći korak je izdvajanje značajki iz praćenih ruku. To može uključivati izdvajanje oblika, položaja prstiju, orijentaciju ili nekakve dinamičke informacije o kretanju. Značajke se koriste za opisivanje gesta i stvaranje reprezentacije podataka nad kojima se može izvršiti klasifikacija.

Pri implementaciji koristila se *cv2.resize* funkcija za promjenu veličine izrezane slike ruke kako bi se izdvojio oblik i veličina ruke kao značajke. Ovisno o odnosu visine i širine ruke, primjenjuje se odgovarajuća strategija za izdvajanje značajki.

Ako je odnos visine i širine manji ili jednak jedan, koristilo se skaliranje slike ruke na širinu od *imgSize* i odgovarajuću visinu koja održava omjer strana. Ova promjena veličine omogućuje konzistentnu veličinu ulazne slike za *klasifikator*.

Ako je odnos visine i širine veći od jedan, tada se koristilo skaliranje slike ruke na visinu od *imgSize* i odgovarajuću širinu koja održava omjer strana. Na taj se način, osigurava konzistentna veličinu ulazne slike za *klasifikator* bez iskrivljavanja aspekta ruke.

Nakon promjene veličine, ažurira se slika *img\_white* s izdvojenom rukom na bijeloj pozadini (Slika 3.7). Ta slika se potom koristi kao ulaz za klasifikator kako bi se izvršila klasifikacija geste. Na slici (Slika 3.8) prikazano je kako se koriste funkcije *resize* i kako se koristi *img\_white* za dobivanje željenih rezultata.



Slika 3.7 img\_white

```

if hands:
    hand = hands[0]
    x, y, w, h = hand['bbox']

    img_white = np.ones((imgSize, imgSize, 3), np.uint8) * 255
    img_crop = img[y - offset:y + h + offset, x - offset:x + w + offset]

    img_crop_shape = img_crop.shape
    aspect_ratio = h / w

    if aspect_ratio <= 1:
        k = imgSize / w
        h_cal = math.ceil(k * h)
        img_resize = cv2.resize(img_crop, (imgSize, h_cal))
        img_resize_shape = img_resize.shape
        h_gap = math.ceil((imgSize - h_cal) / 2)
        img_white[h_gap:h_cal + h_gap, :] = img_resize
        prediction, index = classifier.getPrediction(img_white, draw=False)

    else:
        k = imgSize / h
        w_cal = math.ceil(k * w)
        img_resize = cv2.resize(img_crop, (w_cal, imgSize))
        img_resize_shape = img_resize.shape
        w_gap = math.ceil((imgSize - w_cal) / 2)
        img_white[:, w_gap:w_cal + w_gap] = img_resize
        prediction, index = classifier.getPrediction(img_white, draw=False)

```

Slika 3.8 Izdvajanje značajki

### 3.5. Klasifikacija gesta

Nakon izdvajanja značajki, primjenjuje se klasifikacijski model koji koristi strojno učenje. Model se trenira s označenim podacima o gestama kako bi naučio prepoznati i klasificirati različite geste. Klasifikacijski modeli mogu uključivati algoritme poput neuronskih mreža, potpornih vektora ili slučajnih šuma.

Za klasifikaciju u ovom kodu ovog rada koristila se *Classifier* metoda iz *cvzone* biblioteke. Ovaj klasifikator koristi strojno učenje kako bi naučio prepoznati i klasificirati različite geste na temelju značajki koje se izdvajaju iz praćenih ruku. Učitava se prethodno obučeni

model *Model/keras\_model.h5*. Te se nakon toga koristi *classifier.getPrediction()* metoda klasifikatora za dobivanje predikcije i indeksa klase za trenutnu gestu. Kao ulaz za klasifikator koristi se izrezana slika ruke sa bijelom pozadinom. Klasifikator daje predikciju geste. Predikcija geste može se koristiti za prikaz rezultata na izlaznoj slici ili za daljnju obradu. *Classifier* metoda i *keras\_model* iskorišten je na način prikazan na slici (Slika 3.9).

```
classifier = Classifier("Model/keras_model.h5", "Model/labels.txt")
labels = ["HelloSign", "OkSign", "PeaceSign", "ThumbsDown", "ThumbsUp"]
prediction, index = classifier.getPrediction(img_white, draw=False)
```

Slika 3.9 Klasifikacija i predikcija

### 3.6. Intrepretacija i obrada rezultata

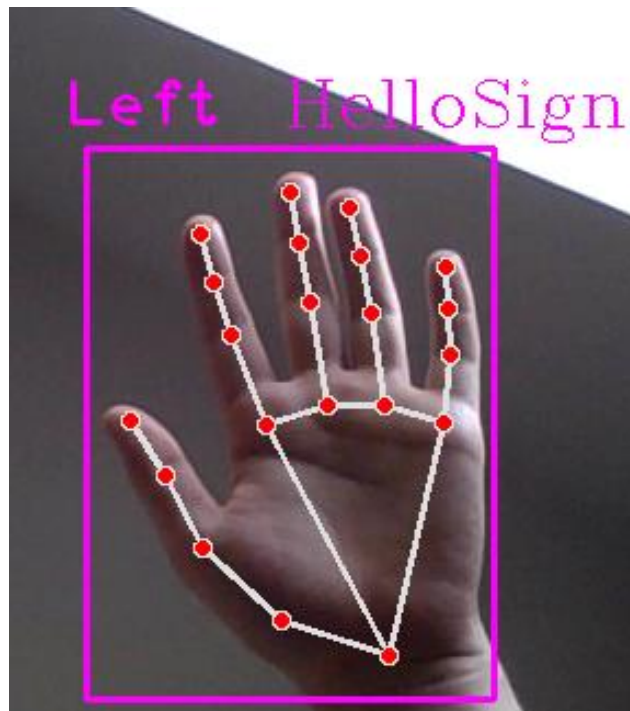
Posljednji korak u implementaciji sustava prepoznavanja gesta je interpretacija rezultata klasifikacije gesta. Nakon što su ulazni podaci procesirani kroz razvijeni algoritam prepoznavanja, slijedi važan korak koji omogućava konkretnu upotrebu prepoznatih gesti. Ovisno o primjeni sustava, interpretacija rezultata može uključivati izvršavanje odgovarajuće radnje ili interakciju s računalom, robotom, virtualnom stvarnošću ili nekim drugim sličnim sustavima.

Za interpretaciju i obradu rezultata u radu se koristila funkcija *putText* kako bi rezultat klasifikacije prikazao naziv geste izlaznoj slici. S tom funkcijom ispisuje se naziv prepoznate geste. Također koristi se varijablu *labels[index]* kako bi se dobio naziv prepoznate geste na temelju indeksa koji je doiven iz klasifikatora. Taj ispis naziva geste pojaljuje se samo u slučaju kada korisnik odabere testni način rada, to jest kada je u načinu rada prikupljanja podataka i snimanja podataka neće se ispisivati koja je gesta prikazana.

Dodatno, ovisno o primjeni sustava, mogu se implementirati daljnje radnje ili interakcije na temelju prepoznate geste. Na primjer ako je prepoznata gesta „*ThumbsUp*“, sustav može odgovarajuću radnju kao što je potvrдна povratna informacija, no u ovom radu to nije implementirano.

Ukoliko je korisnik pri pokretanju koda odlučio testirati kako radi program, iznad *bounding boxa* ispisat će se ime prepoznate geste (Slika 3.10), a implementacija toga vidi se na slici (Slika 3.11).





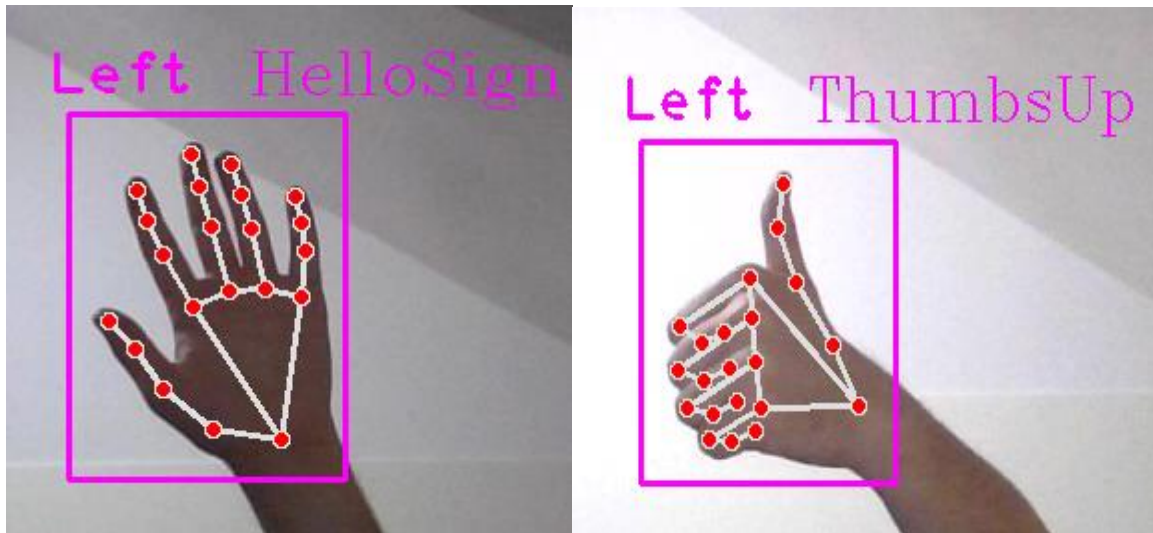
Slika 3.10 Primjer u kojem se vidi ispis naziva geste

```
if choice == "test":  
    cv2.putText(img_output, labels[index], (x + 70, y - 30), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 0, 255), 1)
```

Slika 3.11 Ispis teksta

### 3.7. Rezultati

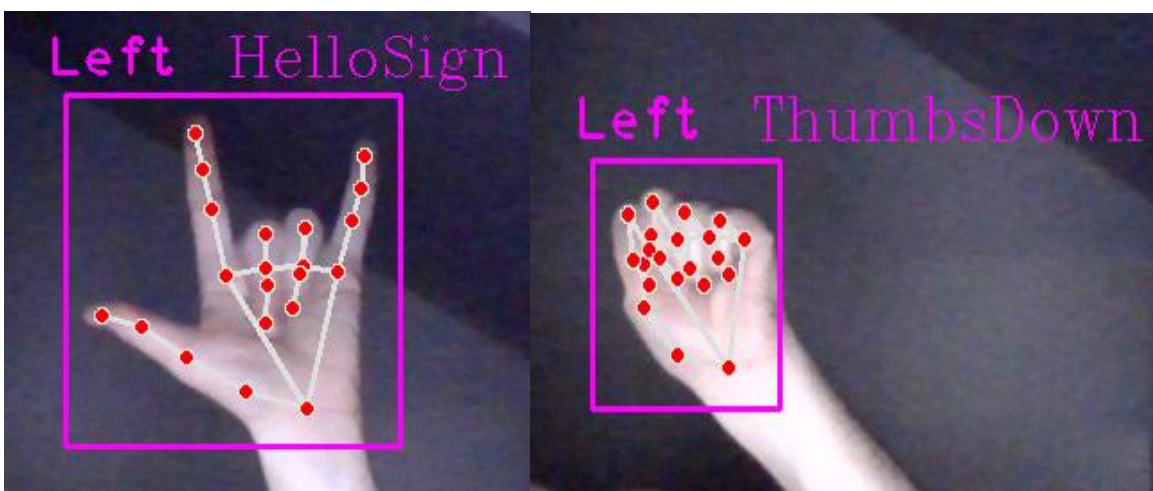
Rezultati ovoga rada su izrazito zadovoljavajući. Program je sposoban konzistentno prepoznati o kojoj se ruci radi i označiti ključne točke šake te postavlja okvir dobre veličine i poprilično dobro prepoznaje o kojoj se gesti radi (Slika 3.12). No postoje situacije u kojima program nije sposoban prepoznati o kojoj je gesti riječ. Do toga dolazi u situacijama kada se šaka nalazi pod čudnim kutom ili kada osvjetljenje smeta u procjeni. Program uvijek ispisuje naziv geste, čak i u situaciji kada se ne prikazuje nikakva gesta i takav slučaj prikazan je na slici (Slika 3.14). A nekada jednostavno pretpostavi da se radi o gesti koja to nije. Kao na primjer u ovom slučaju, zbog nepovoljnog položaja ruke u obziru na kameru program smatra da je riječ o znaku 'PeaceSign' kada se ustvari radi o znaku 'OkSign' (Slika 3.13) ili iz istog razloga iako šaka pokazuje 'OkSign' zbog neobičnog položaja program zaključuje da se radi o 'ThumbsDown' znaku (Slika 3.13).



Slika 3.12 Primjeri dobrih rezultata



Slika 3.13 Primjeri loših rezultata, krivi znak



Slika 3.14 Primjer loših rezultata, nepostojeci znak

## 4. Primjena prepoznavanja gestikulacija ruku

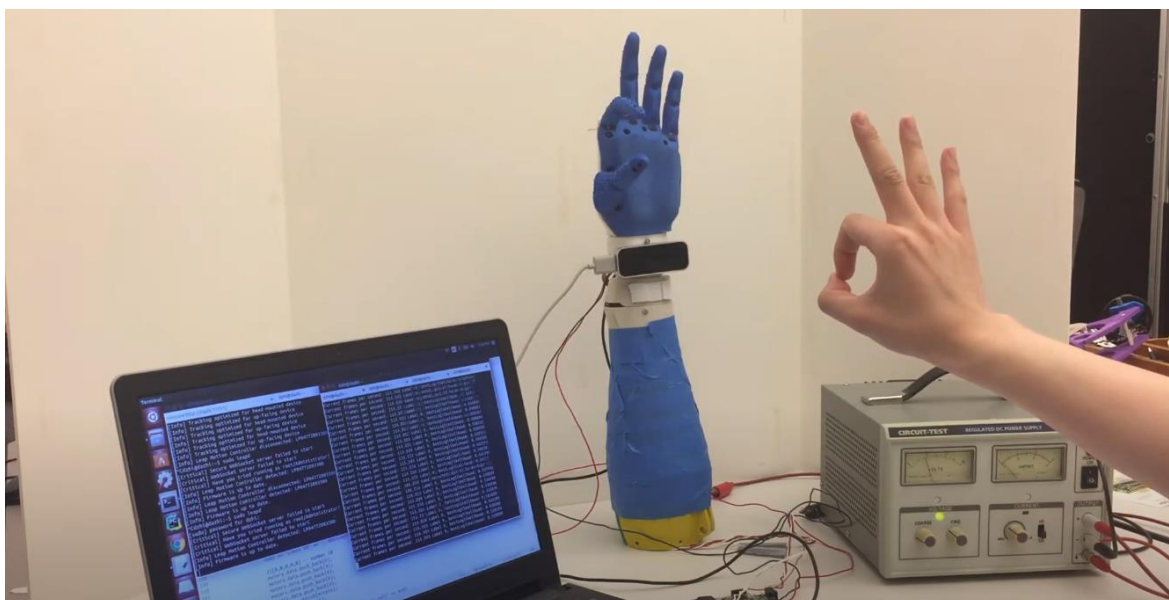
Preoznavanje gesti ruku ima široki spektar primjene u različitim područjima.

### 4.1. Primjeri primjene

Neki od mnogobrojnih načina na koje se prepoznavanje gestikulacija ruku može primjeniti su: robotska interakcija, interakcija s računalom, virtualna i proširena stvarnost, edukacija i rehabilitacija te sigurnost i nadzor. Ova je tehnologija dinamična i otvara puno mogućnosti za inovativne i praktične primjene u raznim područjima.

#### 4.1.1. Robotska interakcija

To može biti primjena u robotskoj interakciji, gdje se prepoznavanje gestikulacija ruku koristi za interakciju između ljudi i robota (Slika 4.1). Korisnicima se omogućuje upravljanje robotima putem gesta, komunikacija s robotima na intuitivan način ili demonstriranje raznih zadataka.



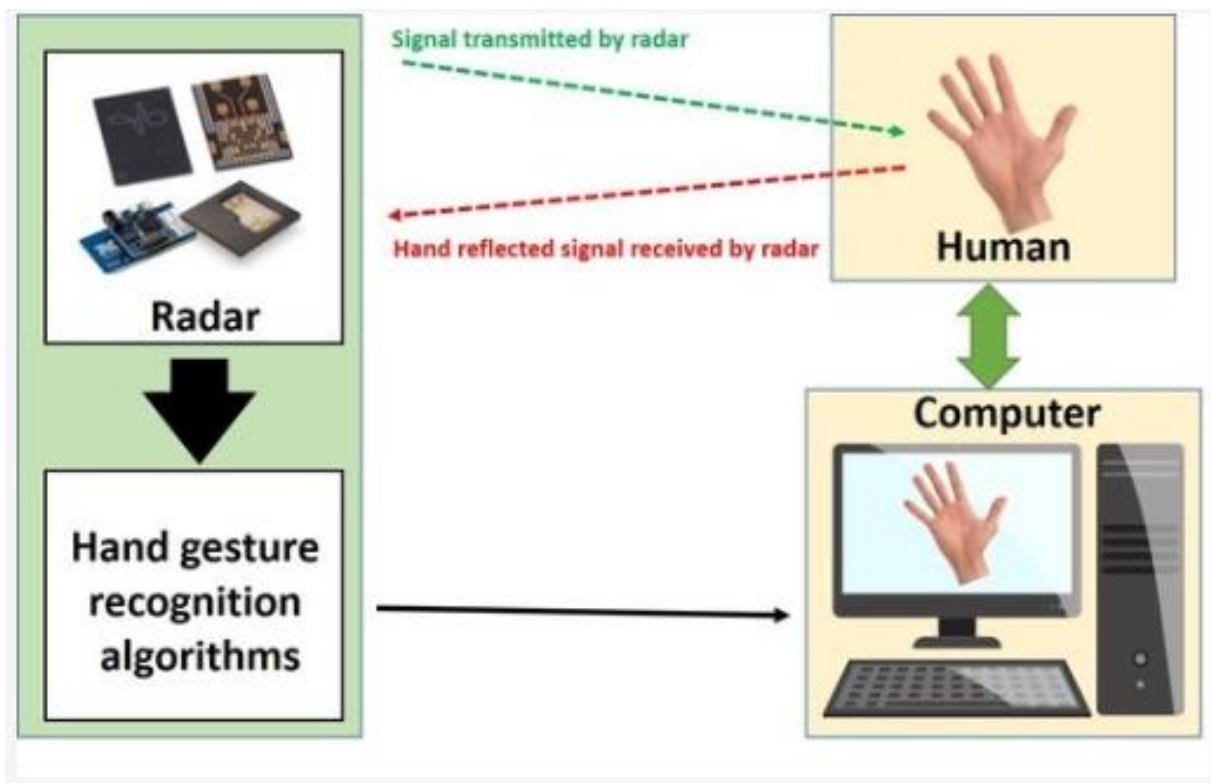
Slika 4.1 Robotska ruka imitira pokrete čovjekove ruke<sup>11</sup> [3]

---

<sup>11</sup> <https://www.youtube.com/@bioinroboticslabatuottawa7666>

### 4.1.2. Interakcija s računalom

Također se može koristiti u interakciji s računalom (Slika 4.2). Prepoznavanje gestikulacija ruku omogućuje korisnicima interakciju s računalima putem prirodnih gesti i pokreta, a postoje dvije vrste prepoznavanja gestikulacija ruku: statička i dinamička. U statičkom prepoznavanju, prepoznavanje se temelji na obliku ruke, dok dinamičko prepoznavanje uključuje kretanje ruku i temelji se na trajektoriji kretanja ruke u prostoru kako bi se izvršilo prepoznavanje. Može se koristiti za kontrolu programa, unos teksta, crtanje, navigaciju, manipulaciju objektima na zaslonu i slično.



Slika 4.2 Proces interakcije ruke s računalom<sup>12</sup>

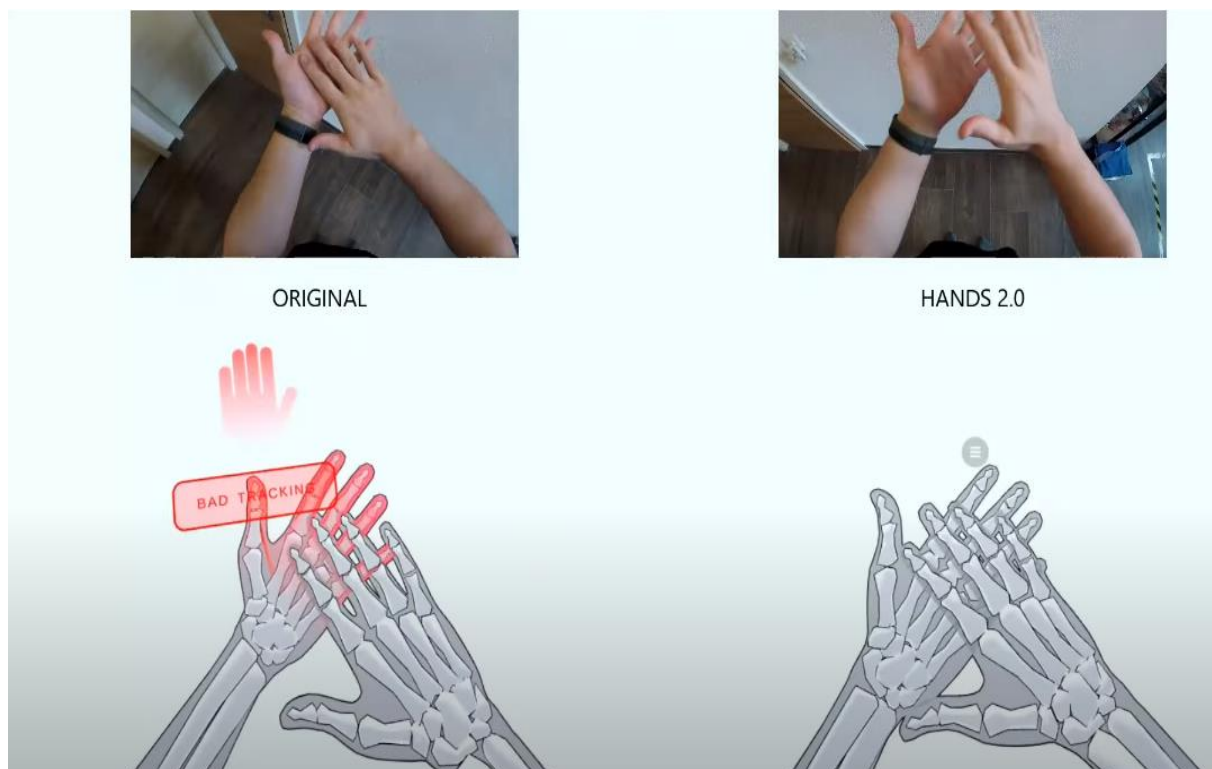
### 4.1.3. Virtualna i proširena stvarnost

Osim toga prepoznavanje gestikulacija ruku može se primjeniti i u području virtualne i proširene stvarnosti (Slika 4.3). U tom području prepoznavanje gestikulacija ruku omogućuje korisnicima interakciju s virtualnim objektima i okolinom putem gesta. Može

---

<sup>12</sup> <https://www.mdpi.com/2072-4292/13/3/527>

se koristiti za manipulaciju virtualnim objektima (Slika 4.4), za stvaranje imerzivih iskustava ili za nešto jednostavno poput odabira opcija.

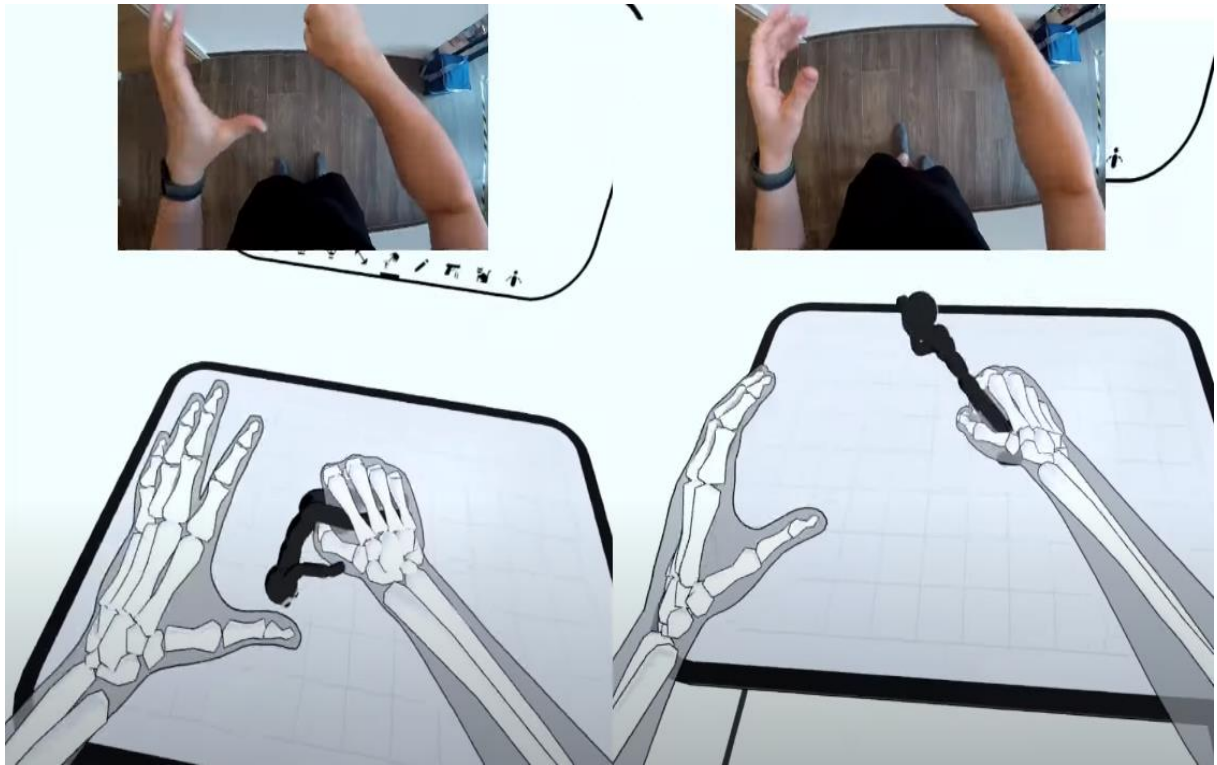


Slika 4.3 Primjer igrice u virtualnoj stvarnosti<sup>13</sup>

---

<sup>13</sup> <https://www.youtube.com/watch?v=8oMukwt7VDs>





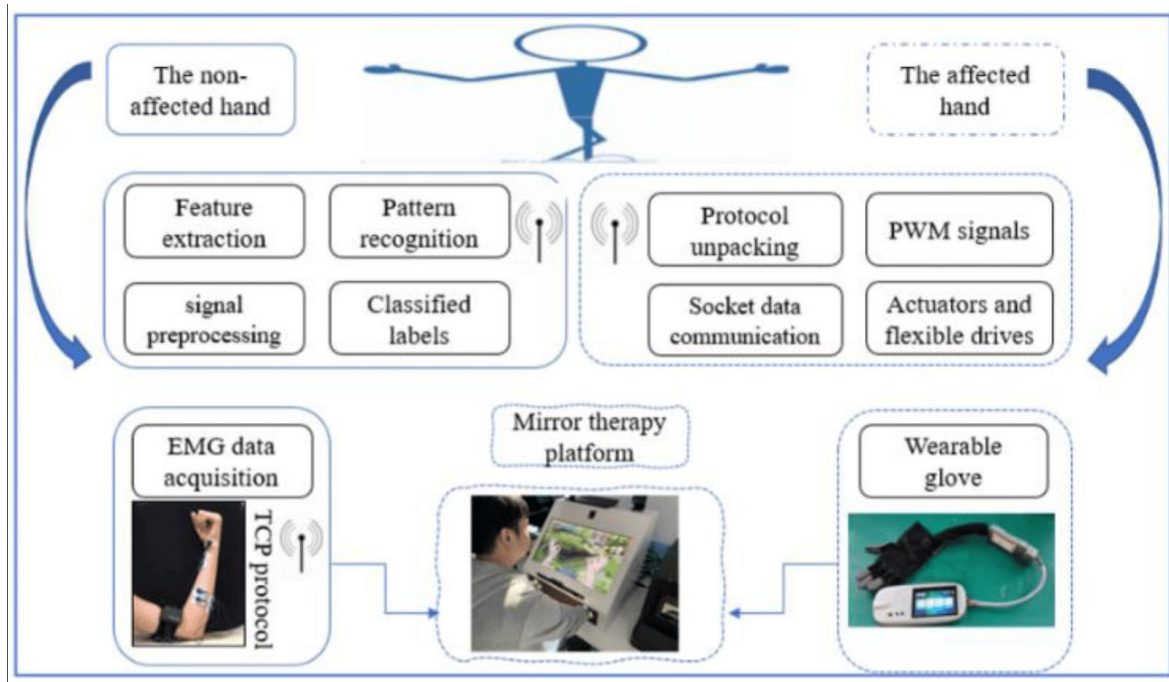
Slika 4.4 Primjer igrice u virtualnoj stvarnosti<sup>14</sup>

#### 4.1.4. Edukacija i rehabilitacija

Uz sve to prepoznavanje gestikulacija ruku može biti korisno i u nekim područjima koji su utjecajni u stvarnom svijetu kao na primjer u edukacija i rehabilitacija. Može se koristiti u obrazovnim i rehabilitacijskim programima za poticanje interaktivnog učenja ili rehabilitacije motoričkih funkcija (Slika 4.5). Može se koristiti za podučavanje znakovnog jezika, rehabilitaciju nakon ozljeda ili poticanje motoričkih vještina.

---

<sup>14</sup> <https://www.youtube.com/watch?v=8oMukwt7VDs>



Slika 4.5 Rehabilitacija za osnaživanje ruke pomoću umjetne inteligencije sa znanjem prepoznavanja gesti<sup>15</sup>

#### 4.1.5. Sigurnost i nadzor

I na kraju, prepoznavanje gesti ruku, može se iskoristiti i u nadzoru i sigurnosti. U sigurnosnim i nadzornim sustavima, prepoznavanje gestikulacija ruku može se iskoristiti za detekciju određenih gesta ili signala za identifikaciju prijetnji, upozorenje ili provjeru pristupa.

<sup>15</sup> <https://www.mdpi.com/2306-5354/10/5/557>

## Zaključak

Nakon razvoja algoritma, prikupljanja podatka te razvoja modela primjetio sam da nismo dobili u potpunosti željeni rezultat, no isto tako program ne daje niti katastrofalne rezultate. Sigurno možemo reći da se s većim brojem slika u bazi za svaku klasu i s još detaljnijim modelom mogu postići bolji rezultati, no ni nikada ne bi bilo savršeno.

Možemo zaključiti da prepoznavanje gestikulacija ruku može itekako pomoći ljudima u raznim područjima kao što su sigurnost i nadzor, edukacija i rehabilitacija, interakcija čovjeka i računala, virtualna stvarnost. Za očekivati je da će se kroz idućih nekoliko godina ovo područje još više razviti te će to dovesti do još boljih rezultata i do još šire primjene prepoznavanje gestikulacija ruku u videosekvencama.



## Literatura

- [1] Farah Farhana Mod Ma'asum, Suhana Sulaiman and Azilah Saparon, An Overview of Hand Gestures Recognition System Techniques, IOP Conference Series: Materials Science and Engineering, Volume 99, 4th International Conference on Electronic Devices, Systems and Applications 2015 (ICEDSA) 14–15 September 2015, Kuala Lumpur, Malaysia
- [2] Aashni Hariaa, Archanasri Subramaniana, Nivedhitha Asokkumara, Shristi Poddara, Jyothi S Nayaka, Hand Gesture Recognition for Human Computer Interaction, 7th International Conference on Advances in Computing & Communications, ICACC-2017, 22-24 August 2017, Cochin, India
- [3] BioIn Robotics Lab at uOttawa. Poveznica:  
<https://www.youtube.com/@bioinroboticslabatuottawa7666>
- [4] Murtaza's Workshop - Robotics and AI. Poveznica:  
<https://www.youtube.com/@murtazasworkshop>
- [5] Osama Mazhar, Sofiane Ramdani, Benjamin Navarro, Robin Passama, Andrea Cherubini. A Framework for Real-Time Physical Human-Robot Interaction using Hand Gestures. ARSO: Advanced Robotics and its Social Impacts, Sep 2018, Genova, Italy.

## Sažetak

### DETEKCIJA KLJUČNIH TOČAKA RUKU I PREPOZNAVANJE GESTI U VIDEOSEKVENCI

U ovom završnom radu razvijao sam programsko rješenje za prepoznavanje gestikulacija ruku u videosekvencama. Cilj rada je bio ostvariti preciznu detekciju ključnih točaka ljudske šake/dlana i prepoznavanje unaprijed definiranih gesti koristeći model dubokog učenja. Uz pomoć već postojećih biblioteka u pythonu napravio sam program pomoću kojeg korisnik sam može uzimati podatke za treniranje modela te koji je sposoban, ovisno o kvaliteti istreniranog modela, prepoznati te unaprijed određene gestikulacije ruku.

Zbog korisnosti prepoznavanja gestikulacija ruku u raznim područjima, možemo reći da se nadamo da će se u idućih nekoliko godina poboljšati modeli za prepoznavanje gestikulacija i da će se prepoznavanje gesti ruku početi koristiti u još više svrha.

**Ključne riječi:** ruke, prepoznavanje gesti, praćenje pokreta, slika, video; Python.

## Summary

### DETECTION OF KEY HAND POINTS AND GESTURE RECOGNITION IN VIDEO SEQUENCES

In this final paper, I developed a software solution for gesture recognition of hands in video sequences. The aim of the paper was to achieve precise detection of key points on the human hand and recognize predefined gestures using deep learning models. Using existing Python libraries, I created a program that allows users to collect data for training the model and is capable of recognizing the predetermined hand gestures based on the quality of the trained model.

Due to the usefulness of hand gesture recognition in various fields, we hope that in the upcoming years, models for gesture recognition will improve, and the utilization of hand gesture recognition will expand to even more applications.

**Keywords:** hands, gesture recognition, movement tracking, images, video, Python.