

# Robotska percepcija položaja ljudske ruke

---

Grgić, Antea

Undergraduate thesis / Završni rad

2023

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:382519>

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

*Download date / Datum preuzimanja:* **2024-11-05**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**  
**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I**  
**INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

**Preddiplomski sveučilišni studij Računarstvo**

**Izborni blok Programsko inženjerstvo**

**ROBOTSKA PERCEPCIJA POLOŽAJA LJUDSKE RUKE**

**Završni rad**

**Antea Grgić**

**Osijek, 2023.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac ZIP - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 15.09.2023.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju**

<b>Ime i prezime Pristupnika:</b>	Antea Grgić
<b>Studij, smjer:</b>	Programsko inženjerstvo
<b>Mat. br. Pristupnika, godina upisa:</b>	R4490, 27.07.2020.
<b>OIB Pristupnika:</b>	87274339558
<b>Mentor:</b>	doc. dr. sc. Petra Pejić
<b>Sumentor:</b>	Valentin Šimundić, mag. ing. comp.
<b>Sumentor iz tvrtke:</b>	
<b>Naslov završnog rada:</b>	Robotska percepcija položaja ljudske ruke
<b>Znanstvena grana rada:</b>	<b>Umjetna inteligencija (zn. polje računarstvo)</b>
<b>Zadatak završnog rad:</b>	Potrebno je razviti računalni program za estimaciju parametara ljudske ruke snimljene kamerom. Odrediti nalazi li se ljudska ruka na snimljenoj slici u zadanom položaju. Na temelju detektiranog položaja ljudske ruke pokrenuti predefimiranu robotsku akciju. Tema rezervirana za: Antea Grgić Sumentor s FERIT-a: Valentin Šimundić
<b>Prijedlog ocjene završnog rada:</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina
<b>Datum prijedloga ocjene od strane mentora:</b>	15.09.2023.
<b>Datum potvrde ocjene od strane Odbora:</b>	24.09.2023.
Potvrda mentora o predaji konačne verzije rada:	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum: 29.09.2023.



**FERIT**

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

## IZJAVA O ORIGINALNOSTI RADA

Osijek, 25.09.2023.

**Ime i prezime studenta:**

Antea Grgić

**Studij:**

Programsko inženjerstvo

**Mat. br. studenta, godina upisa:**

R4490, 27.07.2020.

**Turnitin podudaranje [%]:**

8

Ovom izjavom izjavljujem da je rad pod nazivom: **Robotska percepcija položaja ljudske ruke**

izrađen pod vodstvom mentora doc. dr. sc. Petra Pejić

i sumentora Valentin Šimundić, mag. ing. comp.

mog vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

# SADRŽAJ

<b>1. UVOD</b> .....	<b>1</b>
<b>1.1. Zadatak završnog rada</b> .....	<b>2</b>
<b>2. POVEZANA ISTRAŽIVANJA</b> .....	<b>3</b>
<b>3. METODOLOGIJA</b> .....	<b>6</b>
<b>3.1. Korištene metode i alati</b> .....	<b>6</b>
3.1.1. OpenPose.....	6
3.1.2. OpenCV.....	7
3.1.3. NumPy.....	7
3.1.4. Random Forests klasifikator .....	8
3.1.5. ROS.....	10
<b>3.2. Opis izrađenog sustava</b> .....	<b>11</b>
<b>4. Eksperimentalna evaluacija</b> .....	<b>13</b>
<b>4.1. Učenje</b> .....	<b>13</b>
<b>4.2. Testiranje</b> .....	<b>15</b>
<b>4.3. Implementacija na hardver</b> .....	<b>17</b>
4.3.1. UR5 .....	17
4.3.2. Izvršavanje eksperimenta.....	18
<b>5. Zaključak</b> .....	<b>21</b>
<b>LITERATURA</b> .....	<b>22</b>
<b>SAŽETAK</b> .....	<b>24</b>
<b>ABSTRACT</b> .....	<b>25</b>
<b>ŽIVOTOPIS</b> .....	<b>26</b>
<b>PRILOG</b> .....	<b>27</b>

# 1. UVOD

Posljednjih godina, područje robotike je napredovalo, omogućujući novi stupanj interakcije između čovjeka i robota. Pri tome se razvijaju i algoritmi koji omogućavaju robotima da detektiraju različite dijelove tijela i identificiraju ih kao zasebne kategorije. Ljudska ruka u različitim položajima i pokretima predstavlja različite znakove, geste i signale koji služe za komunikaciju. Učenje detekciji i pravodobnoj reakciji na takve signale, omogućuje čovjeku zadavanje zadataka robotu na temelju simbola prikazanog rukom. Na primjer, može se robotu dati signal da započne neku radnju ili da se zaustavi u slučaju opasnosti, može se koristiti pri pomoći u kućanstvima, pomoći starijim ljudima itd.

Robotska percepcija ljudske ruke podrazumijeva prepoznavanje ljudske ruke na slici i njezinu lokalizaciju te razumijevanje znaka ili pokreta kojeg ruka demonstrira. Cilj ovog rada jest razvoj sustava koji, na temelju slike snimljene kamerom, prepoznaje ljudsku ruku i provodi zadanu robotsku akciju za unaprijed određeni znak predstavljen različitim pozama prstiju i dlana. U ovom radu koristit će se termin poza za dani znak. No, potrebno je razlikovati termine poza i položaj. Položaj u robotici podrazumijeva mjesto na kojemu se što nalazi s obzirom na nešto drugo te je određeno pozicijom i orijentacijom. S druge strane, poza određuje držanje, stav ljudskog tijela ili dijelova tijela.

Sustav je razvijen na sljedeći način. Snimljen je skup podataka u obliku RGB slika na kojima se nalazi ljudska ruka u nekoj od predefiniranih poza. Na slikama je pomoću programske biblioteke OpenPose detektirana ljudska ruka i značajke predstavljene zglobovima prstiju i šake. Međusobni položaj detektiranih značajki na slici u odnosu na referentni koordinatni sustav koji je smješten u zglob zapešća predstavlja ulaz u klasifikator. Na temelju predviđene klase, tj. simbola predstavljenog položajem ljudske ruke, odvija se unaprijed zadana robotska manipulacija. Robotska manipulacija izvršava se na stvarnom robotu UR5 kojim se upravlja iz ROS-a. Sav programski kod pisan je u programskom jeziku Python.

Rad je strukturiran na sljedeći način. U poglavlju 2 dani su primjeri povezanih istraživanja i uspoređene su tehnologije pristupa sličnom problemu s tehnologijama primijenjenima u ovom radu. Potom, u poglavlju 3 opisane su korištene tehnologije te principi sustava rada. U poglavlju 4 opisana je eksperimentalna evaluacija. Ona se sastojala od procesa učenja, testiranja izrađenog programskog rješenja te implementacije istoga na robotsku ruku i kameru unutar ROS okvira. Provedeni eksperimenti su opisani kako bi se pokazala uspješnost rada ovakvog sustava. Rad je zaključen u poglavlju 5.

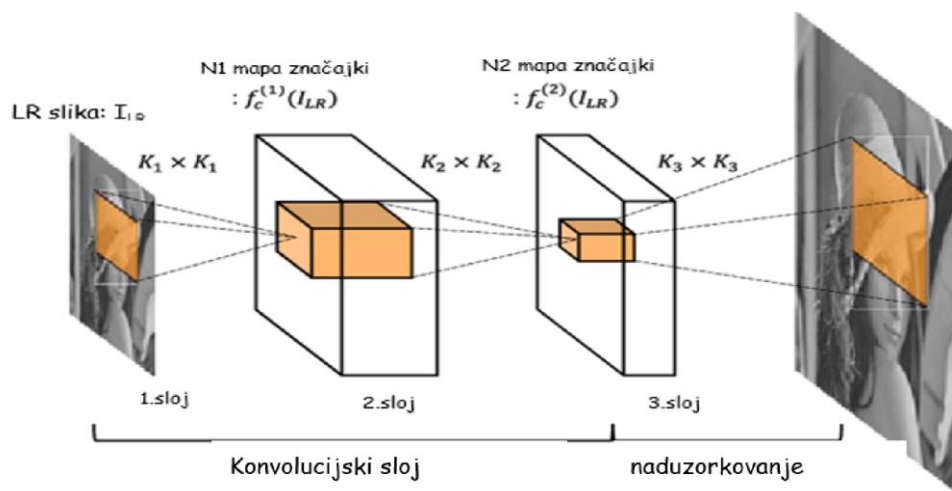
## **1.1. Zadatak završnog rada**

Zadatak ovog završnog rada jest razviti program za upravljanje robotom na temelju klasificiranih simbola šake na slikama. Prvi korak je snimiti skup podataka slika na kojima se ljudska šaka nalazi u jednoj od tri zadane poze. Potom se određuju pozicije zglobova šake na slikama koristeći OpenPose. Pozicije zglobova šake se transformiraju u koordinatni sustav zapešća te se predaju klasifikatoru. Na temelju estimirane klase, robotski manipulator odrađuje predefiniranu kretnju.

## 2. POVEZANA ISTRAŽIVANJA

Ovo poglavlje opisuje druga znanstvena istraživanja na temu prepoznavanja ljudske ruke, s naglaskom na različite metodološke pristupe istraživanju. U sklopu ovoga rada korisno je analizirati različite pristupe razmatranome problemu, njihove prednosti i mane te ostvarene rezultate kako bi se uspješno izabrale, unaprijedile i primijenile postojeće metode.

Prvi takav rad jest „Hand Gesture in Video Sequences Using Deep Convolutional and Recurrent Neural Network“ [1]. U ovome radu autori su za rješenje problema detekcije ljudske ruke preporučili metodu dubokih neuronskih konvolucijskih (CNN) i rekurentnih neuronskih mreža (RNN). One služe za identifikaciju gesta koje ruka pokazuje na ulaznim podacima, tj. videu. Glavni cilj jest kreirati sustav koji ima sposobnost prepoznavanja i klasificiranja gesta ruke u stvarnom vremenu, čineći interakciju između korisnika i sustava jednostavnijom. Prikaz rada CNN-a prikazan je slikom 2.1.



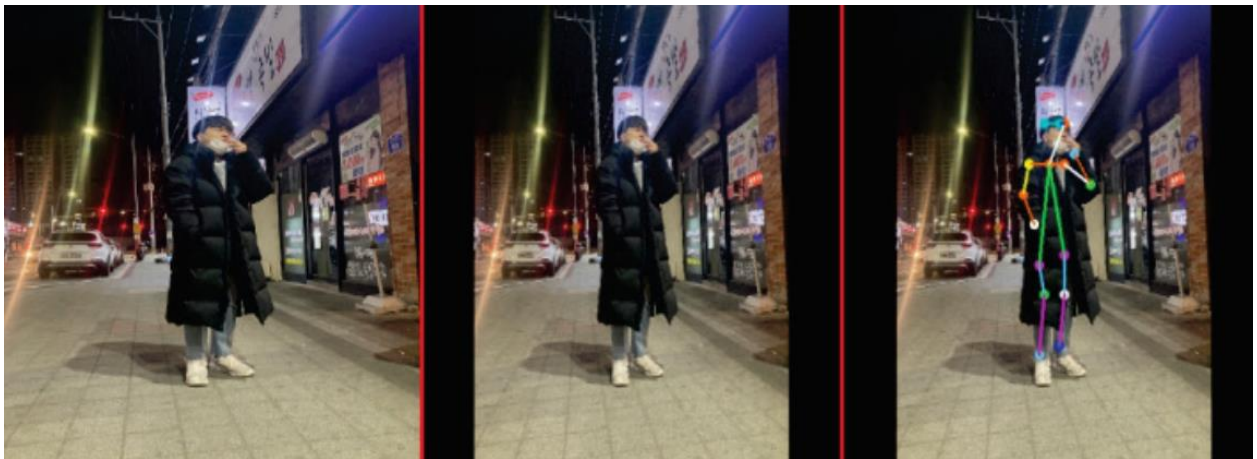
Slika 2.1. Rad konvolucijskih mreža

Na slici 2.1. prvi sloj se primjenjuje na odabranu sliku, na način da filter prolazi slikom i pritom primjenjuje matematičku operaciju konvolucije nad njom. Izlaz je u obliku aktivacijskih mapi, poznatih i kao mape značajki. One prikazuju reakciju filtera na značajke slike te svaka aktivacijska vrijednost od koje se sastoji može biti pozitivna ili negativna. Dobivene aktivacijske mape iz prvog sloja predstavljaju ulaz u drugi sloj. Nad kojima se onda ponovno primjenjuje konvolucija. Nakon konvolucijskih slojeva dolazi do primjene sloja naduzorkovanja (engl. *Upsampling*). Njegova svrha je povećati prostornu dimenzionalnost podataka, koja se vrši nad svim podacima koji izađu iz zadnjeg konvolucijskog sloja. CNN se primjenjuje kako bi se



dobile prostorne karakteristike svakog okvira u videu i RNN kako bi se modelirale vremenske udaljenosti između okvira. Kombinacija ove dvije mreže omogućila im je sve potrebne prostorne i vremenske podatke za prepoznavanje gesta ljudske ruke. Mogućnost prepoznavanja ruke promjenom pokreta je jedna od prednosti ovog tipa pristupa. Međutim, kompleksnost neuronskih mreža može činiti prepreku, pogotovo kada dolazi do njihove primjene u stvarnom vremenu na uređajima s ograničenim resursima. Zahvaljujući vlastitom velikom skupu podataka, uspješno je provedeno testiranje s preciznošću od 93% na posljednjoj verziji modela.

Rad „OpenPose based Smoking gesture Recognition System using Artificial Neural Network“ [2] bazira se na stvaranju sustava koji prepoznaje geste koje izvodi ruka tijekom pušenja pomoću OpenPose [6] modela uz Softmax klasifikator. Cilj je na precizan način identificirati i klasificirati geste u stvarnom vremenu. Dakle, nakon što je sakupljen opširan skup podataka u obliku slika i videa ljudi u pušačkoj pozi, podatci su provučeni kroz OpenPose. S dobivenim izlaznim podacima uspješno je, uz pomoć Softmaxa, ostvarena klasifikacija podataka. U ovom završnom radu je na sličan način prikupljen skup podataka u obliku slika koje su uspješno klasificirane. OpenPose omogućuje preciznu identifikaciju položaja tijela i ruke u obliku  $x$  i  $y$  koordinata svakog zgloba tijela. Slika [2]. prikazuje identifikaciju poze pušača pomoću mreže OpenPose.



*Slika 2.2. Prikaz detekcije poze pušača putem kamere [2]*

Ovakav sustav ipak može imati neke poteškoće. U slučaju da OpenPose krivo detektira neku od točaka, može doći do krive estimacije ili klasifikacije poza. Dakle, od velike je važnosti pri korištenju takvog alata konstantna kontrola ulaznih podataka i kontrola izlaznog polja točaka. U radu je evaluirano nekoliko različitih modela. U slučajevima kada se na slici nalazi osoba koja puši, najbolja postignuta preciznost je 88%, a u slučajevima kada na slici nema osobe koja puši, najbolja postignuta preciznost je 83%.

OpenPose, osim detekcije cijelog tijela, u mogućnosti je detektirati zasebno točke lica i točke ruke. Cilj sljedećeg rada pod nazivom „Detecting Fingerspelling in Sign Language Videos Using Pose Information“ [3] jest identifikacija znakovnog jezika unutar videa koristeći informacije o pozici ruke. Autor je putem OpenPose mreže dobio informaciju o gesti prikazanoj ljudskom rukom (dlanom i prstima). Nakon toga, koristeći CNN uspješno je klasificirao poze ruku i prstiju. OpenPose se i u ovom radu još jednom pokazuje kao koristan alat zbog svoje mogućnosti identifikacije zglobova ruke i prstiju. Upravo ta mogućnost je razlog zašto se OpenPose koristio za rješavanje problema ovog završnog rada.

Još jedan od načina raspoznavanja gesta ruku za interakciju čovjek-robot predstavljena je radom „A Real-time Hand Gesture Recognition System for Human-Computer and Human-Robot Interaction“ [4]. Predstavljeni sustav raspoznaje i klasificira poze ruku koristeći kombinaciju algoritama strojnog učenja i obrade slike. Identifikacija poza bazira se na segmentaciji ruku i ekstrakciji relevantnih podataka. Koristeći duboke neuronske mreže (DCNN), klasificiraju se poze koristeći egzaktne karakteristike svake poze. Takva konvolucijska mreža s velikom preciznošću može raspoznati veliki broj znakova. Prednost ovakvog pristupa jest što omogućuje sustavu poboljšanje predviđanje poza koristeći podatke iz prethodnih testiranja. Opet, problem može predstavljati kriva segmentacija ruku ili loša kvaliteta pruženih podataka. Također, ovaj tip neuronske mreže zahtijeva velike količine resursa. Provođenjem testiranja preciznosti postignuti su rezultati od 93.8% kroz 15 epoha.

Rad „Skeleton-based Dynamic hand gesture recognition“ [5] bazira se na nešto kompleksnijem načinu izvlačenja poza ljudske ruke. Koriste se: histogram rotacije zgloba (HoWR), histogram smjera ruke (HoHD) i oblik povezanih zglobova (SoCJ). SoCJ je zadužen za ekstrakciju prostornih odnosa između različitih dijelova ruke, predstavljajući tako oblik zglobova koji su spojeni. HoWR prikuplja podatke o kretanju zglobova, dok HoHD vraća podatke o orijentaciji ruke u vremenu. Sustavu je dan ulazni skup slika iz kojih izvlači navedene podatke i na samome kraju se ti podatci provlače kroz SVM klasifikator. Ostvarena je preciznost od 83%, čak i na kompleksnim bazama podataka. Mana je što je za ovakav postupak potrebna velika količina resursa.

Prepoznavanje ljudske ruke je aktivno područje istraživanja zadnjih godina. Razni znanstveni radovi prikazuju različite pristupe tim problemima. Iz gore opisanih radova može se zaključiti kako je OpenPose poznat i precizan pristup za detekcije točaka ljudske ruke, potrebnih za treniranje odabranog klasifikatora. Stoga se OpenPose koristi i u ovom završnom radu.

### 3. METODOLOGIJA

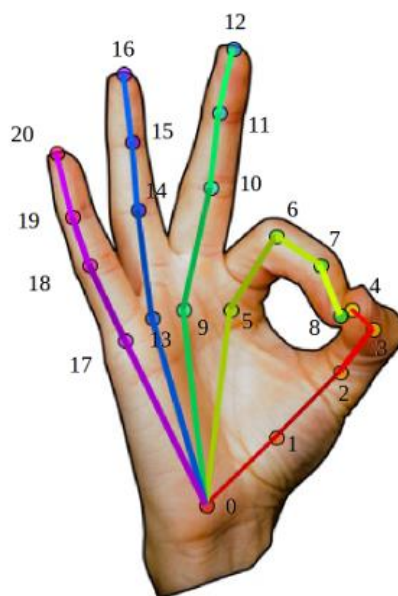
Ovo poglavlje opisuje korištene metode i alate te njihovu primjenu u ovom radu.

#### 3.1. Korištene metode i alati

##### 3.1.1. OpenPose

OpenPose [6] je vrsta modela koja identificira i analizira poze ljudskoga tijela putem konvolucijskih neuronskih mreža. Program je razvio CMU Perceptual Computing Lab i ima široku uporabu u raznim područjima poput robotike, interakcije između čovjeka i robota te kod analize pokreta ljudskog tijela. OpenPose je model istreniran na temelju velike baze podataka označenih slika. Koristi duboku konvolucijsku mrežu koja je zadužena za detekciju ključnih točaka različitih dijelova tijela poput cijelog ljudskog kostura, ljudskog lica ili ljudske ruke na temelju ulaznih podataka u obliku slike ili videa. Izlaz iz mreže su koordinate značajki na slici. U ovom radu OpenPose se koristi za detekciju značajki ljudske ruke na RGB slikama.

OpenPose se sastoji od konvolucije, glasovanja i potpuno povezanih slojeva. Prvih nekoliko slojeva mreže zaduženo je za analizu podataka na slici. Nakon toga, podatci su predani u dva paralelna odjeljka konvolucijskih mrežnih slojeva. Ključne točke definirane su brojem slojeva ekvivalentnog broju točaka. Kod detekcije ruke, na svakoj slici detektirana je 21 točka u obliku mapa vjerojatnosti, od kojih svaka predstavlja zglob zapešća, dlana ili prstiju kao što je prikazano na slici 3.1.



Slika 3.1. Prikaz zglobova ruke označenih točkom

Svaku točku određuje zasebna mapa vjerojatnosti koja predstavlja vjerojatnost pojavljivanja svake od tih točaka u prostoru putem  $x$  i  $y$  koordinata. Dakle, izlaz iz mreže je skup od 22 aktivacijske mape gdje 21 mapa predstavlja po jednu ključnu točku, a zadnja pozadinski sloj.

U OpenPose-u koristi se arhitektura „Convolutional Pose Machine“ (CPM). Ovakav oblik neuronske mreže dobro detektira strukturirane objekte. Strukturirani objekti su objekti sastavljeni od različitih dijelova ili objekata. CPM je sastavljen od više stadija izvođenja. Svaki stadij kreira svoju mapu vjerojatnosti za svaku ključnu točku. Kao ulaz, svaki stadij prima karakteristike slike i podatke vjerojatnosti iz prethodnog. Njihov broj nije ekvivalentan broju ključnih točaka, a rezultiraju brojem mapa koliko je i detektiranih točaka, i to za svaku točku pojedinačno. Stvori se 128 kanala za karakteristike točaka i onoliko mapa vjerojatnosti koliko je i točaka, nadodajući još jednu pozadinsku mapu. U prvom stadiju visina i širina slike smanjuju se s faktorom 8, što rezultira time da je mapa vjerojatnosti manja od početne slike. Mreža se potom trenira i tako se dobiva kontekst podataka. U svakom stadiju dodaje se funkcija gubitka koja se definira svakom pojedinačno kako bi se izbjegao „problem nestajućih gradijenata“. Problem nestajućih gradijenata nastaje kada se produkt derivata smanjuje sve dok se parcijalna derivacija ne krene približavati 0. Problem nestajućih gradijenata dogodi se kada se jedan od gradijenata propagira kroz mrežu unatrag tijekom učenja i postane progresivno jako mal. Zbog ovoga optimizacija neuronskih mreža i njihovo efikasno učenje postaju zahtjevniji.

### **3.1.2. OpenCV**

OpenCV [7] je vrsta biblioteke otvorenog koda za računalni vid. Projektirana je za formiranje seta algoritama i struktura za obradu slika i videa u stvarnom vremenu. Najpoznatija je zbog velikog broja funkcionalnosti i algoritama za promjenu i analizu slika. Podržava razne formate slika i videa kao što su: JPEG, GIF, PNG, BMP, PGM i brojni drugi. Neka od područja primjene su robotika i računalni vid, a koristi se i za prepoznavanje lica, praćenje objekata, analizu pokreta, itd. Bazirana je na programskom jeziku C++, no ima i sučelje koje podržava programske jezike Python, JAVA i MATLAB.

### **3.1.3. NumPy**

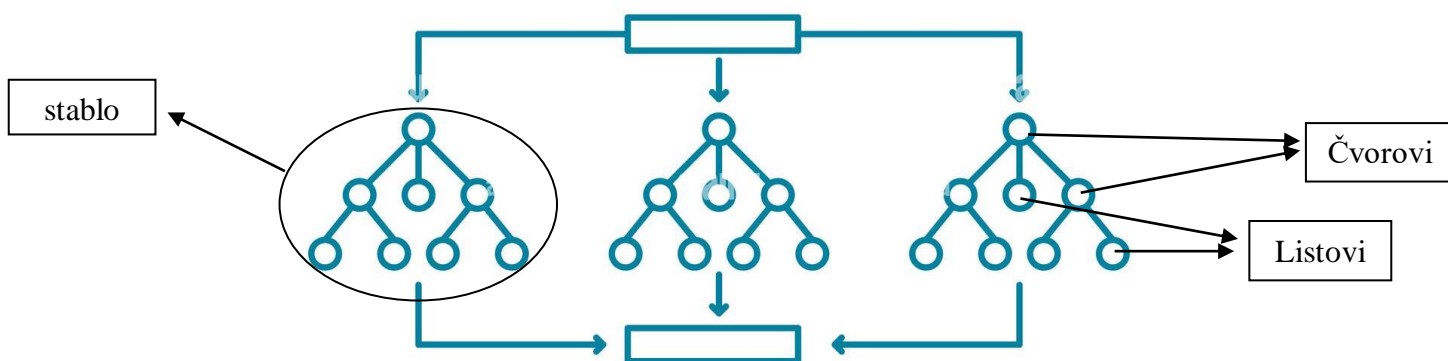
Numerical Python (NumPy) [8] vrsta je biblioteke u programskom jeziku Python koja omogućava uporabu raznih struktura podataka, algoritama i korištenje biblioteka koje se nalaze pod njom. Sadrži *ndarray*, brzo i efikasno višedimenzionalno polje objekata te funkcije i

matematičke operacije za rad s poljima, alate za čitanje i pisanje baza podataka sastavljenih od polja, operacije iz linearne algebre, Fourierove transformacije itd.

Osim što NumPy ima široku primjenu i mogućnost brzog procesiranja polja, jedna od njenih glavnih svrha jest u području analize podataka. Korisna je i kao skladište podataka koji se razmjenjuju među algoritmima i bibliotekama, omogućava lakšu manipulaciju i sortiranje numeričkih podataka. Također i biblioteke koje su izvorno napisane u jezicima niže razine, kao što je C, mogu direktno raditi s podacima koji se nalaze unutar NumPy polja. Često se koristi kao osnovna biblioteka za predobradbu podataka prije nego li se koriste naprednije biblioteke poput biblioteka Pandas (Python Data Analysis Library) i Matplotlib. Pandas je biblioteka pomoću koje možemo kreirati DataFrame, tablične strukture podataka s nazivima kolona i redova. Također, korisna je i za stvaranje serija koje su jednodimenzionalni objekti polja. Koristi se za pripremu, manipulaciju i čišćenje podataka te je zato često korištena pri njihovoj analizi. Matplotlib je najpopularnija Python biblioteka za vizualizaciju podataka, prikaz grafova i drugih dvodimenzionalnih podataka. Dakle, biblioteka NumPy ima širok spektar primjene te je zbog svoje lake uporabe jako popularna.

#### 3.1.4. Random Forests klasifikator

Random Forests klasifikator [9] jest algoritam strojnog učenja korišten za klasifikaciju danih podataka. Kako i samo ime nalaže, sastoji se od velikog broja zasebnih stabala odluke koji rade kao cjelina. Svako od tih zasebnih stabala daje svoju predikciju. Kada svako stablo iznese svoj „stav“, odabire se ono rješenje s najvećim brojem glasova. Na ovaj način zasebna stabla štite jedno drugo od pogreške jer će ipak većina stabala, koja imaju točnu predikciju, odrediti konačno rješenje.



Slika 3.2. Prikaz konstrukcije Random Forests algoritma.

Glavna cjelina algoritma Random Forests je binarno stablo, sagrađeno na temelju rekurzivne particije (RPART) [10]. Dakle, podatci se rekurzivno dijele na manje podskupove podataka kako bi onda dobili homogene ili skoro homogene listove. Homogeni listovi su oni listovi kojima su svi podatci iz samo jedne klase. U skoro homogenim listovima, većina podataka pripada jednoj klasi. Metoda kojom se na taj način grade stabla binarnom podjelom naziva se CART. Random Forests često je sastavljen od tisuća različitih stabla, gdje svako od tih stabla raste pomoću uzoraka originalnog podatka. Takva stabla koriste dva stadija nasumičnog odabira. Prvi stadij je kreiranje stabla, putem uzoraka iz glavnih podataka, a drugi stadij pojavljuje se kod svakog čvora zasebno pri rastu stabla. Umjesto proširivanja čvorova korištenjem svih varijabli pri svakoj iteraciji, Random Forests pri kreiranju svakog čvora odabire novi podskup varijabli i samo se te varijable koriste za daljnje odluke o proširenju. Nasumični odabir u dva koraka ima svrhu poništavanja korelacije među zasebno nastalim stablima. Dokazano je da potpuna čistoća ili polučistoća stabla ima bolje performanse na manjim bazama podataka, što se pokazalo i u ovom radu na skupu podataka od nešto više od 3000 slika.

Konstrukcija Random Forests algoritma može se objasniti u 4 koraka [11]. Prvi korak je crtanje  $n$ -stabala uzoraka iz originalnog stabla. Drugi je kreiranje stabla za skup podataka. Pri kreiranju svakog od čvorova potrebno je zadati  $m$  varijabli za dijeljenje. Treće je agregacija informacija svih novonastalih stabala kako bi se onda putem „glasanja“ izglasala predikcija s najviše glasova. Četvrto su provedena testiranja nad podacima koji nisu bili dani pri treniranju modela.

Postoje četiri mjere mjerenja važnosti varijabli. Dvije se baziraju na GINI indeksu čvora i točnost klasifikacije podataka. Recimo da je dan čvor  $t$  i vjerojatnost klase  $p(k|t)$ , za  $k=1, \dots, Q$ , GINI je definiran na sljedeći način:

$$G(t) = 1 - \sum_{k=1}^Q p^2(k|t), \quad (3-1)$$

gdje  $Q$  predstavlja broj klasa.

Random Forests otporan je na prenaučenosť (engl. *overfitting*), jer je svako drvo sastavljeno od podskupa danih podataka te je završna odluka o rješenju temeljena na većinskom glasanju. Može manipulirati i kategoričkim, numeričkim podacima i nedostajućim vrijednostima unutar skupa podataka. Problem koji se može pojaviti kod korištenja ovog klasifikatora jest što njegova izvedba može zahtijevati veliki broj resursa vremena i prostora, pogotovo kada se radi o

velikom broju podataka. Također, nebalansirani skup podataka, gdje jedan tip podataka ima veću vjerojatnost pojavljivanja od druge, može također rezultirati lošim performansama ovog modela.

Poznat je po tome da mu je primjena raznolika, da je precizan i da može riješiti komplicirane klasifikacijske probleme. Algoritam se primjenjuje u mnogim područjima poput klasifikacije testova ili u medicini. U ovome radu odabran je zbog svojih odličnih performansi na manjem skupu podataka. Pokazao se kao klasifikator koji je s visokom preciznošću uspješno klasificirao sve tri poze ruke. Također je brz, efikasan i dobro radi sa podacima u prostoru.

### 3.1.5. ROS

Robotski Operacijski Sustav, ROS (engl. *Robot Operating System*) [12] platforma je otvorenog koda koja se koristi za izradu robotskih aplikacija. Omogućuje veliki broj programa i biblioteka koje pomažu pri upravljanju robotom. ROS je osnovan 2007. godine, a znanstvenik Morgan Quigley je njegov kreator koji mu je prvobitno dao ime Switchyard. Kreiran je u sklopu STAIR projekta, ali ROS je zapravo u potpunosti razvijen tek kasnije u tehnološko-istraživačkom laboratoriju Willow Garage [13].

Glavni princip ROS-a [14] jest fleksibilna platforma za lakši razvoj robotskih sustava. Takva platforma omogućava stvaranje komunikacijske infrastrukture koja dopušta raznim komponentama robotskog sustava da međusobno komuniciraju te omogućuje komunikaciju među različitim programima. Takve poruke se zapravo temelje na „peer-to-peer“ odnosu zato što se šalju bez centraliziranog usmjeravanja. Ovakva vrsta pristupa čini izmjenu podataka kompleksnijom, no omogućava slanje većeg broja podataka. Također, takav pristup olakšava razvoj robotskih sustava i omogućava lakšu integraciju različitih komponenti. Prednost ROS-a jest ta što njegov velik broj biblioteka sadrži i mnoge funkcionalnosti poput percepcije, planiranja pokreta, kontrola pokreta i simulacija. ROS je kompatibilan s raznim programskim jezicima poput C++, JAVA, MATLAB, Python i mnogim drugima. Kompatibilan je i s različitim verzijama hardvera i robota. Kada bi se napisao program za određeni model robota, on bi u pravilu trebao onda raditi na svim modelima tog tipa pa čak i na nekim njemu samo sličnim modelima. Fleksibilnost ROS-a je ono što ga čini toliko popularnim i korištenim u svijetu robotike.

Čvorovi 13 su glavne komponente projekata pisanih u ROS-u. Dije se na glavne čvorove i korisničke čvorove. Svaki čvor može se registrirati tokom svoga izvođenja i komunicirati s drugim čvorovima zahvaljujući glavnom čvoru. Dakle, oni su procesi koji izvode neku specifičnu radnju ili dio neke takve radnje. Postoje korisničke biblioteke koje se koriste pri

njihovom pisanju, koje omogućuju skup funkcionalnosti kao i samu komunikaciju među čvorovima. Razvijaju manje podsustave pomoću kojih pojednostavljaju rad kompleksnijih sustava. Glavni čvor sadrži i poslužitelja parametara koji omogućuje spremanje podataka u neku središnju lokaciju koja je dostupna svakom čvoru za korištenje i izmjenu. Ta izmjena vrši se preko tema koje predstavljaju sabirnice poruka. Svaki čvor putem teme može ili objavljivati podatke (slati ih) ili primiti podatke.

### 3.2. Opis izrađenog sustava

U svrhu izrade ovog završnog rada, napravljena je baza od preko 3000 slika ljudskih ruku u tri različite poze: otvoreni dlan s ispruženim prstima, stisnuta šaka s podignutim palcem („like“) i stisnuta šaka s podignutim kažiprstom. Te su slike služile za učenje i testiranje sustava klasifikacije. Prvo su slike predane kao ulazni podatci OpenPose modelu. Izlaz iz ovog modela je skup  $Y$  kojeg čine koordinate  $(x, y)$  značajki, tj. točke zglobova šake i prstiju sa svake slike. Skup  $Y$  zapisan je u obliku dvodimenzionalnog vektora značajki:

$$Y = [(x_1, y_1), (x_2, y_2), \dots, (x_{21}, y_{21})]. \quad (3-2)$$

Vrijednosti su potom upisane u CSV formatu, tako da je pritom svaka slika označena brojem 0, 1 ili 2, ovisno o pozi ruke koja se nalazi na slici. Koordinate su potom transformirane iz koordinatnog sustava slike u koordinatni sustav zapešća, kako bi se smanjio utjecaj pozicije šake na slici. Transformiranje se vrši translacijom pozicije zgloba pomoću vektora translacije  $d$ :

$$d = [0 - x, 0 - y], \quad (3-3)$$

gdje su  $x$  i  $y$  koordinate zapešća na slici. Tada se svaki zglob šake na slici translacija na sljedeći način:

$$p_{translated} = p + d, \quad (3-4)$$

gdje je  $p$  točka zgloba prije transformacije, a  $p_{translated}$  točka nakon translacije. Na slici 3.3. prikazan je ispis transformiranih podataka iz CSV datoteke gdje 0 na početku predstavlja klasu, a koordinate koje slijede pisane su formatom izlaznog vektora  $Y$  (3-2).

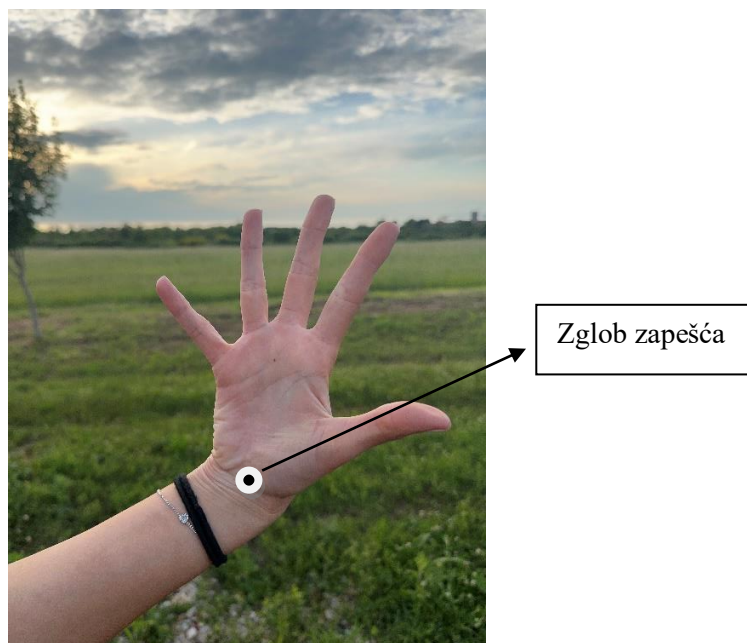


```
0,"[[0.0, 0.0], [10.0, -23.0], [8.0, -55.0], [-31.0, -72.0], [-73.0, -89.0], [-43.0, -125.0], [-50.0, -210.0], [-61.0, -262.0], [-75.0, -313.0], [-77.0, -101.0], [-116.0, -130.0], [-85.0, -90.0], [-69.0, -73.0], [-97.0, -74.0], [-136.0, -84.0], [-101.0, -54.0], [-78.0, -45.0], [-125.0, -40.0], [-154.0, -42.0], [-123.0, -24.0], [-104.0, -19.0]]"
```

```
0,"[[0.0, 0.0], [-9.0, -65.0], [0.0, -155.0], [48.0, -182.0], [18.0, -390.0], [8.0, -169.0], [15.0, -274.0], [22.0, -334.0], [17.0, -393.0], [35.0, -138.0], [139.0, -168.0], [116.0, -153.0], [88.0, -144.0], [61.0, -113.0], [140.0, -164.0], [132.0, -133.0], [107.0, -120.0], [88.0, -81.0], [156.0, -127.0], [146.0, -115.0], [126.0, -102.0]]"
```

Slika 3.3. Ispis transformiranih podataka u CSV datoteci

Zglob zapešća uvijek je prisutan na slici te je kao značajka označen rednim brojem 0, kao što je prikazano slikom 3.4.



Slika 3.4. Prikaz zgloba zapešća

Ishodište novog koordinatnog sustava nalazi se u zglobu zapešća, a  $x$  i  $y$  osi orijentirane su kao u koordinatnom sustavu slike. Vektori značajki  $Y$  potom su predani kao ulazni podatci za Random Forests klasifikator. Potom se provodi učenje i dobiveni model koristi se za kasnije predviđanje znakova na novim slikama. Predviđanjem se određuje radi li se o klasi, tj. pozici ljudske ruke 0, 1 ili 2.

Na testnoj, novoj slici detektira se ljudska ruka i njezine značajke na sljedeći način. Slika se prvo predaje modelu OpenPose. Nakon detekcije značajki i transformacije podataka, dobivene se koordinate koriste za predviđanje klase istreniranim modelom. Predviđena klasa 0, 1 ili 2 predstavlja ulaz za robotsku manipulaciju. Ovisno o predviđenoj klasi, robot provodi zadanu predodređenu manipulaciju kako je opisano u idućem poglavlju.

## 4. Eksperimentalna evaluacija

Eksperimentalna evaluacija izvođena je na operacijskom sustavu Ubuntu 20.04 u ROS Noetic distribuciji. Cijeli kod dan je uz rad.

### 4.1. Učenje

Snimljen je skup od preko 3000 RGB slika. Slikane su uređajem iPhone 14 PRO, koji ima 3 kamere, gdje je glavna kamera od 48 MP, 24 mm. Rezolucija slika je potom smanjena na 974 px širine i 1848 px visine radi brže obrade. Za trening skup izdvojeno je 80% podataka. Odabrane su 3 poze: otvoreni dlan s ispruženim prstima (1026 slika), stisnuta šaka s podignutim palcem (1307 slika) i stisnuta šaka s podignutim kažiprstom (1075 slika). Slika 4.1. prikazuje slike sa sve 3 poze.



Slika 4.1. Prikaz korištenih poza šake

Učenje je provedeno kako je opisano u trećem poglavlju. Svaka slika trening skupa prvo je predana modelu OpenPose. Rezultat toga prikazan je slikom 4.2.



Slika 4.2. Rezultat OpenPose modela.

Slika 4.2. prikazuje zglobove šake i prstiju koje je detektirao OpenPose. Točke su potom transformirane u novi koordinatni sustav kojemu je ishodište zglob zapešća (nulti zglob označen na slici). Na slici 4.3. slijedi primjer vektora značajki prije i poslije transformiranja:

```
PRIJE: „[249, 466, 296, 430, 352, 394, 395, 364, 431, 356, 308, 329, 328, 270, 343, 237, 355,
208, 273, 329, 280, 257, 284, 219, 288, 190, 244, 336, 234, 268, 228, 230, 227, 199, 217, 350,
186, 309, 168, 286, 151, 262]“

POSLIJE: „[ 0 0 47 -36 103 -72 146 -102 182 -110 59 -137 79 -196 94 -229 106 -
258 24 -137 31 -209 35 -247 39 -276 -5 -130 -15 -198 -21 -236 -22 -267 -32 -116 -63 -
157 -81 -180 -98 -204]“
```

Slika 4.3. Ispis vektora značajki prije i poslije transformacije

Vektori značajki su potom predani klasifikatoru kao parametri. Parametri Random Forests klasifikatora [15] su sljedeći:

```
class sklearn.ensemble.RandomForestClassifier(n_estimators=100, *, criterion='gini', max_depth=None,
min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None,
min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False,
class_weight=None, ccp_alpha=0.0, max_samples=None) \[source\]
```

Slika 4.4. Parametri Random Forest klasifikatora

Kako je već navedeno u poglavlju 3.1.4., Random Forests radi tako što stvara veliki broj zasebnih stabala odluke. Svako od tih stabala donosi određenu odluku te na temelju ponuđenih odluka donese zaključak o klasi predanog ulaza. Pozivom `sklearn.model_selection` biblioteke u Pythonu omogućeno je korištenje `train_test_split` funkcije kojoj je dan udio od 80% podataka za učenje, kao što je prikazano u sljedećim linijama koda slike 4.5.

Linija	Kod
1:	<code>data = pd.read_csv('dana_base_new.csv')</code>
2:	<code>X = np.array([eval(keypoints_str) for keypoints_str in dana['Key points']])</code>
3:	<code>Y = dana['Label']</code>
4:	<code>X_train, X_test, y_train, y_test = train_test_split(X, y,</code>
5:	<code>test_size=0.2, random_state = 42)</code>

Slika 4.5. Definiranje skupa podataka za učenje i testiranje.

Podatci su potom pretvoreni u oblik dvodimenzionalnog niza. Svaki red niza predstavlja zaseban primjer, a svaki stupac predstavlja određenu točku. Taj korak je potreban kako bi Random Forests mogao raditi s podacima. Nakon treniranja, koje je prikazano na slici 4.6., provedeno je i testiranje.

<i>Linija</i>	<i>Kod</i>
1:	<code>classifier = RandomForestClassifier()</code>
2:	<code>Classifier.fit(X_train, y_train)</code>

*Slika 4.6. Treniranje modela.*

## 4.2. Testiranje

Preostalih 20% podataka izdvojeno je za testiranje. Ti podatci, kao i oni za treniranje, pretvoreni su u dvodimenzionalni niz nakon čega su predani kao parametri funkcije *predict* koja je pozvana na klasifikatoru kako je opisano slikom 4.7.

<i>Linija</i>	<i>Kod</i>
1:	<code>Y_pred = classifier.predict(X_test_flat)</code>

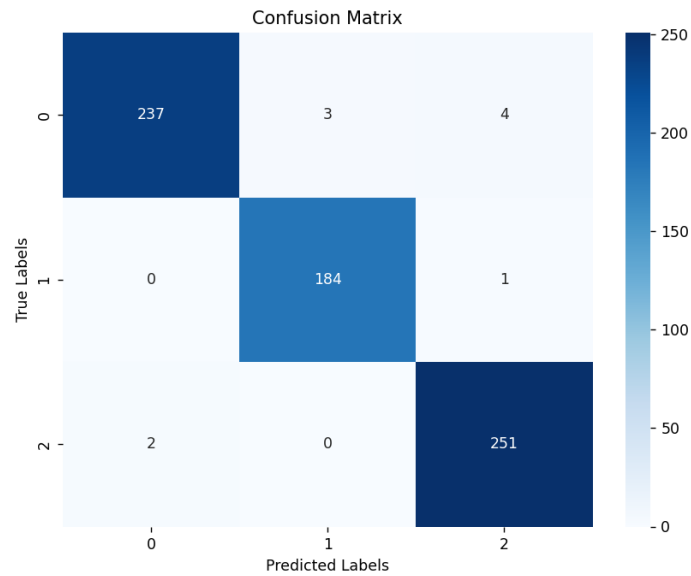
*Slika 4.7. Testiranje modela.*

Na samome kraju izvršena je i evaluacija klasifikatora, a točnost i preciznost provedenog testiranja dane su u tablici 4.1.

*Tablica 4.1. Točnost i preciznost provedenog testiranja*

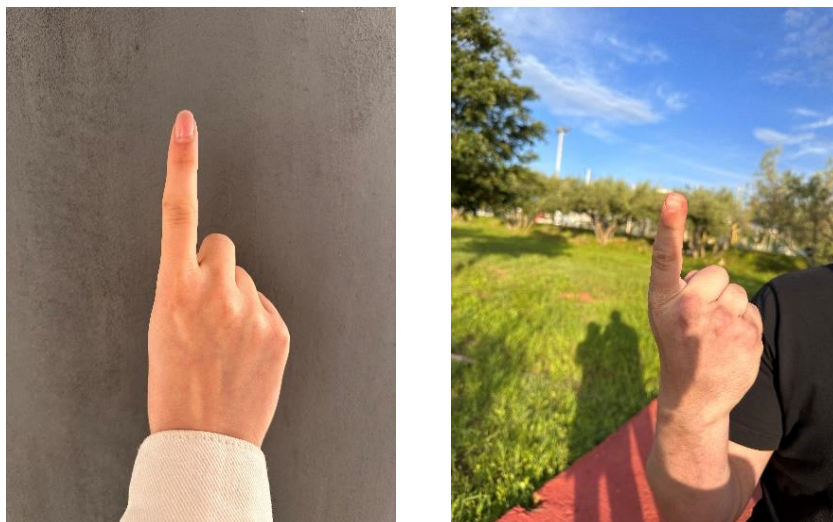
<b>Mjera:</b>	<b>Postotak:</b>
<i>Točnost</i>	98.534%
<i>Preciznost</i>	98.535%

Matrica zabune prikazana je slikom 4.8. Iz nje je vidljivo kako je klasifikator od 682 slike sveukupno točno odredio klasu za njih 672.



*Slika 4.8. Matrica zabune*

Jedna od grešaka koje su nastale tijekom testiranja je između klasa 1 i 2. Klasa 1 predstavlja pozu s ispruženim kažiprstom, a klasa 2 predstavlja pozu s ispruženim palcem. U lijevom primjeru na slici 4.9. u prvom planu je kažiprst, a palac je zbog poze šake nevidljiv. Gledajući potom desni primjer iste slike vidimo kako je kažiprst, radi kuta slikanja, ispio neproporcionalno manji u odnosu na ostatak ruke. Postoji više razloga zašto je moglo doći do pogrešne klasifikacije. Prvi je, da su zbog položaja ruke i kuta slikanja, točke na lijevoj slici pogrešno detektirane, odmah u početku tokom provlačenja slika kroz OpenPose model. Druga mogućnost je ta da su točke pravilno detektirane, no da su koordinate točaka sa lijeve slike sličnije onima palca nego kažiprsta pa je klasifikator pogrešno odredio klasu.



*Slika 4.9. Slika poze s ispravljenim kažiprstom koja je klasificirana kao šaka.*

### 4.3. Implementacija na hardver

U prvom dijelu ovog poglavlja opisana je korištena oprema, a potom integracija programskog i hardverskog dijela u ROS-u.

#### 4.3.1. UR5

Robotska ruka UR5, koju je razvila tvrtka Universal Robots, vrsta je robotskog manipulatora široke primjene koja se može koristiti u raznim područjima industrije. Napredne karakteristike i kontrola preciznosti nad ovim modelom omogućuju njegovu laku uporabu i široku fleksibilnost za izvođenje velike skupine zadataka. Na slici 4.10. prikazan je robotski manipulator UR5.



*Slika 4.10. UR5 Robotska ruka*

Dohvat ove robotske ruke je je 850 mm, a nosivost je do 5 kg. Napravljena je od aluminija, preciznost joj ide sve do 0.1 mm te se sastoji od 6 rotacijskih zglobova koji omogućuju lako i precizno kretanje unutar prostora.

UR5 je kolaborativni robot, kreiran u svrhu lake čovjekove interakcije s robotima, što znači da nema potrebe za dodatnim sigurnosnim alatima tijekom korištenja. Radi toga se lako može koristiti u razne svrhe: pomicanje objekata, slaganje, provjeru kvalitete itd. Robotom se može upravljati pomoću interaktivnog ekrana za upravljanje (engl. *Teach Pendant*) koji je prikazan na slici 4.

[16]. Pomoću njega moguće je kontrolirati robota, putem velikog crvenog gumba prisilno ga zaustaviti, konfigurirati njegov stisak, postaviti inicijalni položaj, itd. Ima jednostavno sučelje koje mogu koristiti i početnici i iskusni inženjeri.



*Slika 4.11. Interaktivni ekran za upravljanje robotom*

#### **4.3.2. Izvršavanje eksperimenta**

Tijek eksperimenta je opisan u nastavku. Prvo čovjek stane ispred kamere i napravi jednu od tri predefiniране poze koju snimi kamera. Slika potom prolazi kroz OpenPose i klasifikator te daje jedan od predviđenih izlaza: 0, 1 ili 2. Sljedeći korak je kreiranje čvora u ROS-u. Ta funkcija kao ulaz prima brojeve 0, 1 ili 2, koji označavaju pojedinu pozu ljudske ruke. Ovisno o primljenom ulazu, funkcija daje informaciju robotu. Upravljanje robotom je dodano kao drugi čvor u ROS-u i robot se, ovisno o ulaznoj informaciji, giba na predefiniрани način. Ako je robot primio vrijednost 0, on se postavlja u startnu poziciju gdje su mu koordinate  $x$ ,  $y$  i  $z$  predefiniране. U slučaju vrijednosti 1, robot se pomiče za 40 centimetara u negativnom smjeru  $x$  osi i za 50 centimetara u pozitivnom smjeru  $z$  osi. Ako mu je dan ulazni podatak 2, pomiče se za iste udaljenosti, ali u pozitivnom smjeru  $x$  osi i pozitivnom smjeru  $z$  osi. Programski kod prikazan je slikom 4.12.

**Linija**   **Kod**

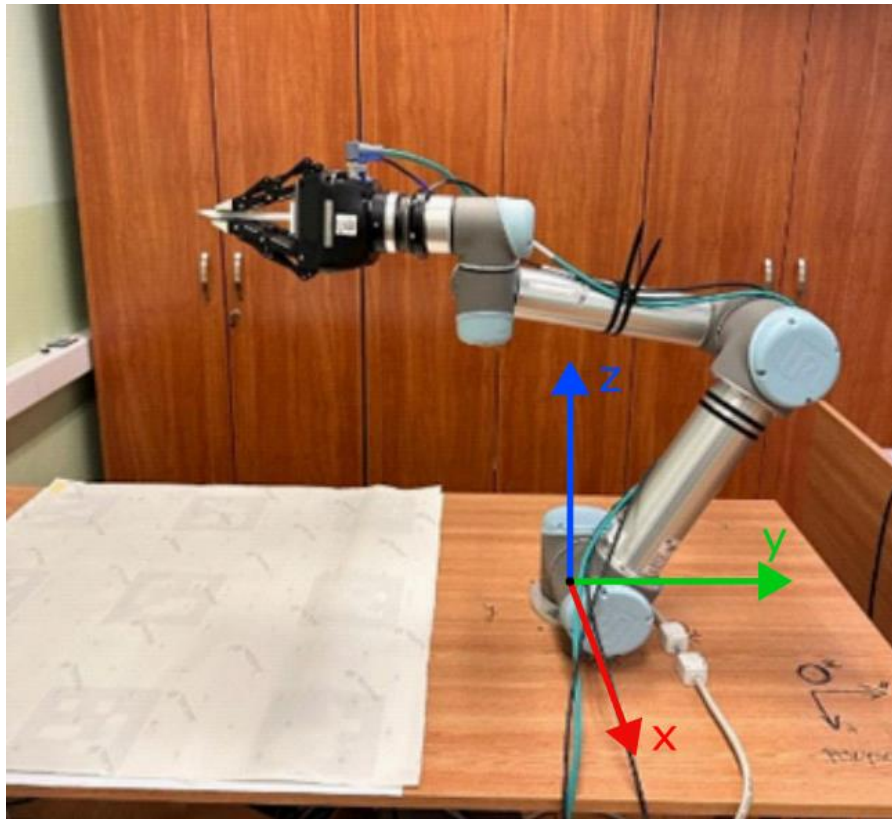
```
1:     if __name__ == '__main__':
2:         #instanca upravljacka robotom
3:         RC = RobotControl()
4:         StartPose = RC.get_current_pose()
5:         while not rospy.is_shutdown():
6:             current_pose = RC.get_current_pose()
7:             x = input(„ODABERI POZICIJU 1, 2 ILI 3“)
8:             goal_pose = Pose()
9:             goal_pose.orientation = current_pose.orientation
10:            goal_pose.position = current_pose.position
11:            if x == „0“:
12:                goal_pose = StartPose
13:            elif x == „1“:
14:                goal_pose.position.x = -0.4
15:                goal_pose.position.z = 0.5
16:            elif x == „2“:
17:                goal_pose.position.x = 0.4
18:                goal_pose.position.z = 0.5
19:            RC.set_pose_and_execute(goal_pose)
```

*Slika 4.12. ROS čvor za upravljanje robotom na temelju ulaznog podatka x*

Programski kod na slici 4.12. prikazuje gore opisane situacije. Dakle, funkcije primaju položaj robota koji je određen pozicijom i orijentacijom. Pozicija je definirana 3D koordinatama vrha alata u koordinatnom sustavu baze robota kako je prikazano na slici 4.13., dok je orijentacija prikazana kvaternionom. Kvaternion je matematički izraz koji opisuje orijentaciju objekta u 3D koordinatnom sustavu.

Robot se, nakon što je zaprimio ulazni podatak, kreće gibati i to radi sve dok vrh alata ne dođe u definirani položaj. Kako je prikazano na slici 4.13., robot stoji u početnom položaju iz kojega, ovisno o primljenoj klasi 0, 1 ili 2, napravi predefiniranu radnju. Nakon provedenih testova, zaključeno je kako je implementacija uspješno provedena na robotskoj ruci.





*Slika 4.13. Eksperimentalno okruženje s označenim koordinatnim sustavom baze robota*

## 5. Zaključak

U ovom je radu opisan razvijeni sustav za robotsku manipulaciju na temelju percepcije poze ljudske ruke. Na snimljenom skupu RGB slika, pomoću OpenPose modela, prepoznaje se ljudska ruka i detektiraju značajke, koordinate zglobova šake. Koordinate predstavljaju ulaz u klasifikator Random Forest. Na danom skupu podataka provedena je evaluacija te je utvrđena točnost od 98.5% na testnom skupu.

Na temelju izlaza iz klasifikatora, koji predstavlja pozu ljudske ruke, označenu klasom 0, 1 ili 2, provodi se predefinirana robotska radnja. Funkcija za provođenje robotskih radnji implementirana je kao ROS čvor koji kao ulaz prima vrijednosti 0, 1 ili 2 koji predstavljaju klase, tj. poze šake, koje konačno određuju koju predefiniranu radnju treba izvršiti. Eksperimenti su uspješno provedeni na stvarnom robotu UR5.

Razvijeni program bi se mogao unaprijediti dodavanjem čvora kamere, kao i razvijenih i postojećih programa za obradu slike u ROS. Na taj bi se način podatci u obliku slike prenosili direktno s kamere u OpenPose model te dalje u klasifikator. Na ovaj način bi se omogućilo detektiranje poze ljudske ruke i odgovarajuća robotska akcija u stvarnom vremenu.

Primjena ovakvog sustava je raznolika, počevši od nekih jednostavnih zadataka, poput igre „kamen, škare, papir“, pa sve do primjene u industriji i kućanstvima.

## LITERATURA

- [1] Falah Obaid, Amin Babadi, i Ahmad Yoosofan, „Hand Gesture Recognition in Video Sequences Using Deep Convolutional and Recurrent Neural Networks“, 05 Lipnja 2020., str. 57-61
- [2] Tae-Yeong Jeong, i Il-Kyu Ha, „OpenPose based Smoking Gesture Recognition System using Artificial Neural Network“, 2022., str. 251-259
- [3] Tobias van der Werf, „Detecting Fingerspelling in Sign Language Videos Using Pose Information“, Bachelor thesis Computing Science from Radboud University 28 Lipnja 2020.
- [4] Valerio Ponzi, Emanuele Iacobelli, Christian Napoli, i Janusz Starczewski, „A Real-time Hand Gesture Recognition System for Human-Computer and Human-Robot Interaction“, Valerio Ponzi et al. CEUR Workshop Proceedings, 2022., str. 52-58
- [5] Quentin De Smedt, Hazem Wannous, Jean-Philippe Vandeborre, „Skeleton-based Dynamic hand gesture recognition“, Tel'ecom Lille, Univ. Lille, CNRS, UMR 9189 - CRISAL, F-59000 Lille, France, 2016.
- [6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shin-En Wei, i Yaser Sheikh, „OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields“, 2014. IEEE International Conference on Consumer Electronics (ICCE), str. 172-186
- [7] Ivan Culjak, David Abram, Tomislav Pribanic, Hevoje Dzapo, i Mario Cifrek, „A brief introduction to OpenCv“, 2012. Proceeding of the 35th International Convention MIPRO
- [8] Wes McKinney, „Python for Data Analysis“, listopad 2012, Pristupljeno: lipanj 27, 2023. [Na internetu], Dostupno na: <https://wesmckinney.com/book/python-basics>
- [9] Leo Breiman, „Random Forests“, siječanj 2001., Statistics Department University of California Berkeley, CA 94720
- [10] Gerard Biau, Luc Devroye, i Gabor Lugosi, „Consistency of Random Forests and Other Averaging Classifiers“, rujna 2008., Journal of Machine Learning Research 9 (2008) 2015-2033
- [11] Cuong Nguyen, Yong Wang, i Ha Nam Nguyen, „Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic“, 2013., Journal of Biomedical Science and Engineering
- [12] Bernhard Dieber, Ruffin White, Sebastian Taurer, Benjamin Breiling, Gianluca Caiazza, Dr.

Henrik I. Christensen, i Professor Agostino Cortesi, „Robot Operating System - The Complete Reference (Volume 4)“, 2019., Part of the Studies in Computational Intelligence book series (SCI, volume 831)

- [13] Joseph Lentin i Jonathan Cacace, Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System, Second Edition. Birmingham: Packt Publishing Limited, 2018., Pristupljeno: lipanj 30. 2023, preko stranice: <https://www.oreilly.com/library/view/mastering-ros-for/9781788478953/>
- [14] Pablo Estefoa, Jocelyn Simmondsa, Romain Robbesb, i Johan Fabry, „The Robot Operating System: Package Reuse and Community Dynamics“, listopad 2018., Journal of Systems and Software
- [15] Scikit learn, „sklearn.ensemble.RandomForestClassifier“, scikit 1.2.2, Pristupljeno: 15 srpnja 2023., preko stranice: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- [16] Universal UR5, Pristupljeno: 15 srpnja 2023., preko stranice: <https://robotsdoneright.com/Universal/Universal-UR5.html>

## SAŽETAK

U ovom završnom radu predstavljen je sustav u kojemu čovjek zadaje zadatke robotu simbolom prikazanog rukom. Nakon što je sakupljena baza RGB slika na kojima se nalazi ljudska ruka u nekoj od sljedećih poza: otvoreni dlan s ispruženim prstima, stisnuta šaka s podignutim palcem („like“) i stisnuta šaka s podignutim kažiprstom, slike su analizirane pomoću OpenPose modela. Pomoću modela, na slikama je detektirana ljudska ruka i značajke koje predstavljaju set od 21 točke zglobova prstiju i šake. Potom, izvađena polja točaka spremljena su u CSV datoteku i transformirana u novi koordinatni sustav s ishodištem u zglobu zapešća. Takvi podatci dani su klasifikatoru Random Forests kao ulaz. Na temelju međusobnog položaja transformiranih značajki treniran je klasifikator. Skup za treniranje i testiranje raspoređen je u omjeru 80:20. Uspješnost klasifikatora izmjerena je preciznošću i točnošću od 98.5%. Eksperimenti su provedeni i na stvarnom robotu UR5 pomoću ROS-a, gdje je na temelju izlaza iz klasifikatora, robot izvršavao predefimirane kretnje.

## **ABSTRACT**

Title: Robotic Perception of Human Hand Pose

In this final paper, a system is presented in which a person assigns tasks to a robot with the symbol shown by the hand. After collecting a database of RGB images of a human hand in one of the following poses: open palm with fingers extended, clenched fist with thumb raised ("like"), and clenched fist with index finger raised, the images were analyzed using the OpenPose model. Using the model, a human hand and features representing a set of 21 points of the finger and hand joints are detected in the images. Then, the extracted point fields were saved in a CSV file and transformed into a new coordinate system with the origin at the wrist. This data is given as input to the Random Forests classifier. The classifier was trained based on the mutual position of the transformed features. The set for training and testing is distributed in a ratio of 80:20. The success of the classifier was measured with a precision and accuracy of 98.5%. Experiments were also conducted with the real UR5 robot using ROS, where the robot performed predefined motions based on the output of the classifier.

## ŽIVOTOPIS

Antea Grgić rođena je 14.04.2001. godine u Puli. U Novigradu je pohađala Talijansku osnovnu školu Novigrad i osnovnu glazbenu školu. 2020. godine završila je klasičnu gimnaziju Pazinski, smjer klasična gimnazija s pravom javnosti u Pazinu. Iste godine upisala je preddiplomski sveučilišni studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek. Aktivni je volonter IEEE Hrvatske sekcije, potpredsjednica IEEE Sstudentskog ogranaka u Osijeku te višegodišnji ambasador za IEEEExtreme svjetsko natjecanje u programiranju. Bavi se organizacijom raznih događaja, te je stekla iskustvo višegodišnjim radom u organizaciji ATP 250 Croatia Open Umag teniskog turnira.

---

Potpis

## **PRILOG**

Cijeli kod je dostupan u mapi na sljedećem linku:

[https://gitlab.com/antea.grgic/antea\\_grgic\\_zavrzni\\_rad.git](https://gitlab.com/antea.grgic/antea_grgic_zavrzni_rad.git)