

ELEKTRO IN RAČUNALNIŠKA ŠOLA VELENJE  
TRG MLADOSTI 3, 3320 VELENJE  
MLADI RAZISKOVALCI ZA RAZVOJ SAŠA REGIJE

RAZISKOVALNA NALOGA

VZPOSTAVITEV INTELIGENTNE POVEZAVE MED RAČUNALNIŠKO VIZIJO IN  
GIBANJEM ROBOTSKE ROKE

Tematsko področje: ROBOTIKA

AVTORJA:

Blaž Kristan, 2.TRA

Tim Rednjak, 2.TRA

MENTORJA:

Uroš Remenih, inž.

Samo Železnik, inž.

VELENJE, 23.2.2024

Kristan B., Rednjak T. Vzpostavitev inteligentne povezave med računalniško vizijo in gibanjem robotske roke.  
Raziskovalna naloga, ŠC Velenje, 2024

---

Raziskovalna naloga je bila opravljena na Elektro in Računalniški šoli Velenje

Mentorja : Uroš Remenih, inž. inf., Samo Železnik, inž. inf

Datum predavitve:

## KLJUČNA DOKUMENTACIJA INFORMACIJA

ŠD SCV elektro in računalniška šola, šolsko leto 2023/2024

KG Robotika/Programiranje/Natančnost

AV Kristan Blaž

SA Rednjak Tim

KZ 3320 Velenje, SLO, Trg mladosti 3

ZA ŠCV elektro in računalniška šola

LI 2024

IN Vzpostavitev inteligentne povezave med računalniško vizijo in gibanjem robotske roke

TD Raziskovalna naloga

OP

IJ SL

JI Sl / En

AI Raziskovalna naloga "Inteligentna povezava računalniške vizije in gibanja robotske roke" raziskuje uporabo naprednih algoritmov za avtomatizacijo gibanja robotske roke. Namen raziskave je izboljšati natančnost in učinkovitost kirurških posegov s pomočjo robotske tehnologije. Rezultati kažejo na potencialno uporabo robotskega sistema kot kirurškega asistenta, ki bi lahko pomagal kirurgom pri izvajanju natančnih in kompleksnih operativnih posegov. Raziskava prinaša nove možnosti za izboljšanje varnosti in natančnosti kirurških postopkov ter napredek v razvoju medicinskih robotov. Pomembne ključne besede: računalniška vizija, gibanje robotske roke, kirurški asistent, natančnost operativnih posegov, medicinski roboti.

## KEYWORD INFORMATION

ND SCV Electro and Computer School, academic year 2023/2024

CX Robotics/Programming/Precision

AU Kristan Blaž

AA Rednjak Tim

PP 3320 Velenje, SLO, Trg mladosti 3

PB SCV Electro and Computer School

PY 2024

TI Establishment of an intelligent connection between computer vision and robotic arm movement

DTResearch Project

NO

LA SL

AL SI / En

AB The research project "Intelligent Connection of Computer Vision and Robotic Arm Movement" explores the use of advanced algorithms for automating the movement of a robotic arm. The purpose of the research is to improve the accuracy and efficiency of surgical procedures through robotic technology. The results indicate the potential use of a robotic system as a surgical assistant, which could assist surgeons in performing precise and complex operative procedures. The research brings new opportunities to improve the safety and accuracy of surgical procedures and advances in the development of medical robots. Important keywords: computer vision, robotic arm movement, surgical assistant, accuracy of operative procedures, medical robots.

1. UVOD.....	6	3.4 PROGRAMSKI DEL	
1.1 PROBLEM.....	6	3.4.1 IZBIRA PROGRAMSKEGA OKOLJA .....	12
1.2 HIPOTEZE .....	6	3.4.2 POTEK PRIPRAVE VZPOZTAVITVE	
1.2 HIPOTEZE		KOMUNKACIJE MED PYTHONOM TER	
1.2.1 Hipoteza 1.....	6	ARDUINOTOM .....	13
2.1 PROGRAMSKA OPREMA		3.4.1 IZBIRA PROGMSKEGA OKOLJA	
2.1.1 PYTHON .....	6	3.4.1.2 ARDUINO .....	13
2.1.2 C++.....	7	3.4.1 IZBIRA PROGRAMSKEGA OKOLJA	
2.2 STROJNA OPREMA		3.4.1.1 PYCHARM .....	12
2.2.1 ARDUINO .....	7	3.4.2 POTEK PRIPRAVE VZOSTAVITVE	
2.2.2 KAMERA.....	8	KOMUNIKACIJE MED PYTHONOM TER	
2.2.3 SERVO MOTOR SG90 .....	8	ARDUINOTOM	
2.PREGLED OBJAV.....	6	3.4.2.1 PRIPRAVA ZAZNAVANJE ROKE .....	13
2.1 PROGRAMSKA OPREMA .....	6	3.4.2 POTEK PRIPRAVE VZPOSTAVITVE	
2.2 STROJNA OPREMA.....	7	KOMUNIKACIJE MED PYTHONOM TER	
3. SERIJSKA KOMUNIKACIJA .....	8	ARDUINOTOM	
3.1 SERIAL COMUNICATION.....	8	3.4.2.2 PRIPRAVA ZA KODE NA ARDUINOTU ...	16
3.2 MATERIAL IN METODE DE LA.....	8	4. REZULTATI .....	19
3.3 POTEK DE LA .....	10	4.1 ODZIV ROKE .....	19
3.4 PROGRAMSKI DEL.....	12	5. RAZPRAVA .....	20
3.2 MATERIAL IN METODE DE LA		6.POVZETEK.....	20
3.2.1 STROJNA OPREMA .....	8	7. ZAKLJUČEK.....	20
3.2.1 STROJNA OPREMA		8. ZAHVALA.....	21
3.2.1.1 IZBIRA MATERIALA.....	9	9. PRILOGE.....	22
3.2.1.2 IZBIRA MOTORČKOV .....	9	10. VIRI IN LITERATURA .....	26
3.2.1.3 IZBIRA VRVI.....	9		

## 1. UVOD

Računalniška vizija in robotika sta se v zadnjih desetletjih izjemno razvili, kar je omogočilo napredke na številnih področjih, vključno s kirurgijo. Vendar pa je ključna izzivna prepreka pri razvoju robotov v medicini ohranjanje natančnosti in prilagodljivosti pri kirurških posegih. Raziskovalna naloga se osredotoča na vzpostavitev inteligentne povezave med računalniško vizijo in gibanjem robotske roke z namenom izboljšanja kirurških postopkov.

### 1.1 PROBLEM

ki ga obravnavamo, je delo pod stresom, utrujenost in nenatančnost kirurgov, kar lahko privede do neželenih posledic in zapletov. Zato smo se odločili raziskati, kakšno natančnost je mogoče doseči z izdelavo robotske roke ter ali bi bila sposobna delovati v kirurških posegih.

### 1.2 HIPOTEZE

#### 1.2.1 Hipoteza 1

Razvoj in implementacija inteligentne povezave med računalniško vizijo in gibanjem robotske roke bo omogočila zmanjšanje nenatančnosti pri kirurških posegih ter izboljšanje rezultatov in varnosti operativnih postopkov

## 2. PREGLED OBJAV

### 2.1 PROGRAMSKA OPREMA

#### 2.1.1 PYTHON

Python je programski jezik, ki je interpretiran in objektno usmerjen. S svojimi vgrajenimi podatkovnimi strukturami na visoki ravni, dinamičnim tipiziranjem in dinamičnim vezanjem predstavlja učinkovit jezik za hitri razvoj aplikacij ter za uporabo kot skriptni ali povezovalni jezik pri integraciji obstoječih komponent.



Slika 1: Prikaz logotipa ogrodja Python

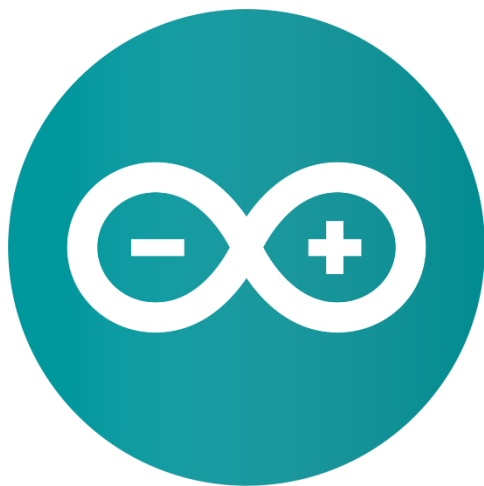
### 2.1.2 C++

C++ je programski jezik, ki je objektno usmerjen in se uporablja za razvoj različnih programskih aplikacij, vključno z vgrajenimi sistemi, operacijskimi sistemi, grafičnimi uporabniškimi vmesniki in igrami. Zaradi svoje učinkovitosti, zmogljivosti in razširjene uporabe se C++ pogosto uporablja pri razvoju aplikacij, kjer je pomembna hitrost izvajanja in učinkovita uporaba sistemskih virov. Kot večnamenski jezik omogoča tako visokonivojsko kot nizkonivojsko programiranje ter integracijo s sistemskimi knjižnicami in zunanji orodji. C++ je priljubljena izbira med programerji, ki potrebujejo zmogljiv jezik za razvoj kompleksnih in zahtevnih aplikacij.

## 2.2 STROJNA OPREMA

### 2.2.1 ARDUINO

Arduino je odprtokodna platforma za razvoj elektronskih naprav, ki jo sestavljajo strojna oprema (mikrokontroler) in programska oprema (IDE). Z uporabo preprostega in intuitivnega programskega jezika, ki je podoben C/C++ omogoča enostavno programiranje mikrokontrolerov za različne aplikacije. Arduino je priljubljena izbira med ustvarjalci, hobi navdušenci in profesionalci zaradi svoje preprostosti, prilagodljivosti in razširjene skupnosti uporabnikov.



Slika 2: Prikaz logotipa platforme Arduino

## **2.2.2 KAMERA**

Kamera je naprava za zajemanje slik ali video posnetkov. Z uporabo optičnih in digitalnih komponent omogoča pretvorbo svetlobnih signalov v digitalne podatke, ki jih lahko shranimo, obdelamo ali predvajamo na različnih napravah. Kamere se uporabljajo v številnih aplikacijah, vključno s fotografijo, videonadzorom, medicinsko diagnostiko, avtomobilsko industrijo, znanstvenimi raziskavami in zabavno elektroniko. Obstajajo različne vrste kamer, vključno s digitalnimi fotoaparati, videokamerami, termalnimi kamerami, nadzornimi kamerami, mikroskopskimi kamerami itd. Kamere so postale nepogrešljiv del sodobne tehnologije in omogočajo zajemanje in deljenje vizualnih informacij na širokem spektru področij.

## **2.2.3 SERVO MOTOR SG90**

Servo motor SG90 je majhen elektromehanski motor, ki ga odlikujejo visoka natančnost in hitrost ter enostavna integracija. Namenjen je za natančno krmiljenje položaja vrtenja na osi v določenem območju. Glavne značilnosti servo motorja SG90 vključujejo majhno velikost, lahko konstrukcijo, visoko zanesljivost in nizko ceno. Primeren je za številne aplikacije, kot so modelarstvo, robotika, avtomatizacija, radijsko vodenje in druge podobne aplikacije, kjer je potrebno natančno in stabilno krmiljenje gibanja. Servo motor SG90 je pogosta izbira med ustvarjalci in hobisti zaradi svoje preproste uporabe in učinkovitosti.

## **3. SERIJSKA KOMUNIKACIJA**

### **3.1 SERIAL COMUNICATION**

Serial Communication omogoča preprost in zanesljiv prenos podatkov med napravami preko fizičnega serijskega vmesnika. V primerjavi z NFC, ki je omejen na kratek doseg, serijska komunikacija omogoča daljšo razdaljo prenosa podatkov, kar jo naredi primerno za številne aplikacije, kot so industrijska avtomatizacija, pametne naprave in avtomobilski sistemi. Z uporabo zaporedja bitov preko ene same povezave omogoča enostavno in zanesljivo povezovanje mikrokontrolerov, senzorjev in drugih naprav z računalniki



## **3.2 MATERIAL IN METODE DE LA**

### **3.2.1 STROJNA OPREMA**

Za izdelavo modela robotske roke smo uporabili predlogo inmoov.fr, enega izmed najbolj priljubljenih odprtih virov za DIY (naredi sam) projekte na področju robotike. Ta vir smo izbrali zaradi njegove priljubljenosti med ustvarjalci ter razpoložljivosti podrobnih navodil in modelov za izdelavo. Pri nadaljnjem razvoju in prilagajanju smo se osredotočili na specifične potrebe našega projekta, ob upoštevanju možnosti prilagoditve, ki jih ponuja ta platforma

#### **3.2.1.1 IZBIRA MATERIALA**

Ko sva razmišljala o materialu za najino raziskovalno nalogo, sva se odločila za uporabo filamenov. Videla sva, da imajo ključno vlogo v sodobnih tehnoloških aplikacijah, predvsem v področju 3D-tiskanja. Filamenti so tisti gradniki, ki omogočajo, da lahko iz digitalnih modelov ustvarimo resnične tridimenzionalne objekte. Fascinirala naju je tudi njihova raznolikost - lahko najdemo filamente iz različnih vrst plastike, kovinskih in kompozitnih materialov. To odpira širok spekter možnosti za raziskave in razvoj novih izdelkov ter inovativnih rešitev. Zato sva se odločila natančno proučiti lastnosti filamentov ter razumeti, kako vplivajo na končni izdelek. Najin cilj je prispevati k razvoju naprednih tehnoloških aplikacij, ki bodo lahko izboljšale naše vsakdanje življenje.

#### **3.2.1.2 IZBIRA MOTORČKOV**

Servo motor SG90 je majhen in cenovno ugoden servo motor, ki se pogosto uporablja v različnih elektronskih projektih in napravah. Gre za vrsto motorja, ki se lahko natančno kontrolira glede na določene vhodne signale. SG90 je znan po svoji zanesljivosti, majhnosti in nizki ceni, kar ga naredi priljubljeno izbiro za hobiste, študente in ustvarjalce. Uporablja se lahko za premikanje majhnih mehanizmov, kot so ročaji,

krmilni elementi ali celo manjše pomične kamere. Ključne značilnosti SG90 vključujejo lahko težo, visoko natančnost pozicioniranja in enostavno upravljanje preko digitalnih signalov, kot so PWM signali.

### 3.2.1.3 IZBIRA VRVI

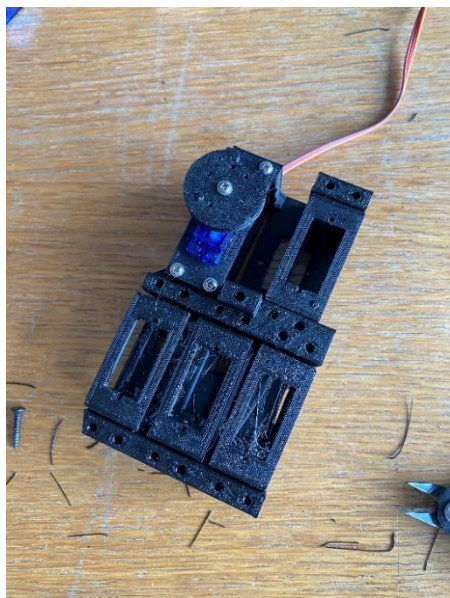
Uporaba laksa kot vrvi ima številne prednosti zaradi njegove moči, lahкости in vzdržljivosti. Laks je prilagodljiv material, ki se enostavno zaveže in oblikuje glede na potrebe, kar ga naredi primerne za različne namene, kot so vzpenjanje, vrvne vaje, vleka in priprava bivališč. Poleg tega je laks vodoodporen in odporen na različne vremenske pogoje, kar ga naredi primerne za uporabo tako na prostem kot v notranjih prostorih. Zaradi svoje razpoložljivosti in relativne nizke cene je laks pogosto izbrana možnost za številne aplikacije, kjer je potrebna zanesljiva in trajna vrv.

### 3.3 POTEK DELA

Iz internetne strani inmoov.fr sma vzela že narisane modele robotske roke, ki sva jih potem naložila na usb ključek in jih shranila po imenu. Potem sva datoteke prekopirala na s SD kartico, ki sva jih uporabila pri 3D printerju.

Modele sva pregledala, če ustrezajo meram in jih po potrebi tudi spremenila.

Ker nisva imela pravilno velikih in močnih motorčkov, sva svoj model prilagodila z novo skico. Novo skico sva narisala večkrat, kajti prišla sva do problema, da se ni ujemala s staro skico.



Slika 3: Prikazuje stojalo za servo motorje (Vir: lasten)

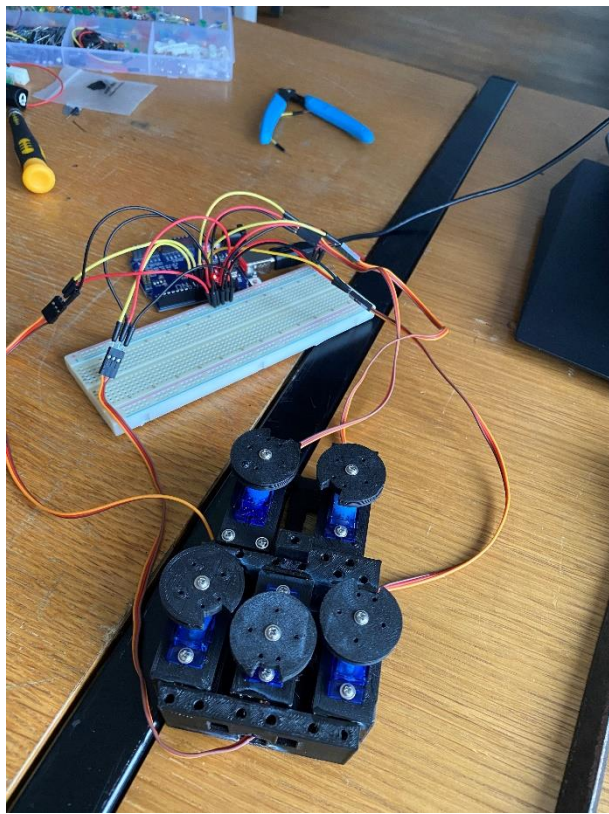
Po rešitvi tega problema sva nadaljevala s printanjem modelov. Sproti sva mu dele tudi prilagajala in lepila skupaj. Ko so bili vsi modeli narisanih, sva začela z prilagajanjem motorčkov v notranjost roke.

Motorke sva povezala na arduino, ker sva naletela na nov problem in to je bilo napajanje motorčkov. Problem sva rešila, tako da napajanje ni prihajalo iz računalnika, ampak je prihajalo iz vtičnice. Nato sva motorček e uspešno postavila na svoje mesto.



Slika 4: prikazuje nameščene motorčke v notranji del roke. (Vir: lasten)

Ko so bili motorčki uspešno postavljeni na svoja mesta sva jih povezala še z napajanjem.



Slika 5: Servo motorji povezani na svoje mesto

Po uspešni povezavi motorčku in arduino. Ta sva to vse povezala še s kamero, ki zaznava premike človeške roke in potem informacije pošlje naprej na arduino, kjer motorčki odreagirajo in odprejo oziroma zaprejo prste.

### **3.4 PROGRAMSKI DEL**

#### **3.4.1 IZBIRA PROGRAMSKEGA OKOLJA**

##### **3.4.1.1 PYCHARM**

PyCharm je integrirano razvojno okolje (IDE) za programski jezik Python, ki ga razvija podjetje JetBrains. Ponuja širok nabor funkcij in orodij, namenjenih olajšanju razvoja Python aplikacij. Značilnosti PyCharma vključujejo napredno urejevalno okolje z možnostjo samodejnega dokončanja kode, integrirano odpravljanje napak, podporo za upravljanje projektov, integracijo s sistemom za nadzor različic, možnost pisanja in izvajanja enotnih testov ter uporabo različnih vtičnikov za razširjanje funkcionalnosti. Zaradi svoje uporabnosti in učinkovitosti je PyCharm priljubljena izbira med programerji Pythona pri razvoju aplikacij v tem programskem jeziku.

### 3.4.1.2 ARDUINO

Arduino IDE (Integrated Development Environment) je programska oprema, ki jo je razvilo podjetje Arduino in je namenjena programiranju mikrokrmilnikov Arduino. To je posebno okolje za razvoj, ki omogoča enostavno pisanje, urejanje, nalaganje in izvajanje programske kode za Arduino mikrokrmilnike. Arduino IDE vključuje vse potrebne komponente in orodja, ki jih potrebujete za začetek programiranja, vključno s sintaksnim poudarjanjem, avtomatskim zaključevanjem kode, upravljanjem knjižnic, vmesnikom za nalaganje kode na mikrokrmilnik in orodji za spremljanje serijskega monitorja. S tem omogoča programerjem, da enostavno razvijajo različne projekte in aplikacije za Arduino platformo.

## 3.4.2 POTEK PRIPRAVE VZPOZTAVITVE KOMUNKACIJE MED PYTHONOM TER ARDUINOTOM

### 3.4.2.1 PRIPRAVA ZAZNAVANJE ROKE

Začela sma delati v Pycharm edu v pythonu

Ka prikazovanje kamere sma uporabila

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

Za zaznavanje roke sem uporabil

```
from cvzone.HandTrackingModule import HandDetector  
  
import cv2  
  
detector = HandDetector(maxHands=1, detectionCon=0.7)
```

Potem sma naredila loop kjer odpre kamero in zaznava roko

Ter da zaznava prste ali so dvignjeni ali ne

Nato jih sprinta

```

while success, img = cap.read()
    # Detect hands
    hands = detector.findHands(img)
    if hands:
        for hand in hands:
            fingers = detector.fingersUp(hand)
            print(fingers)

```

nato sma pripravila serial port

dodala sma

```

import serial

ser = serial.Serial("COM3", 9600, timeout=0.5)

```

in nadaljevala s pretvorbo v signal string

nato sma uporabila ser.write ta posljema singal na arduinota

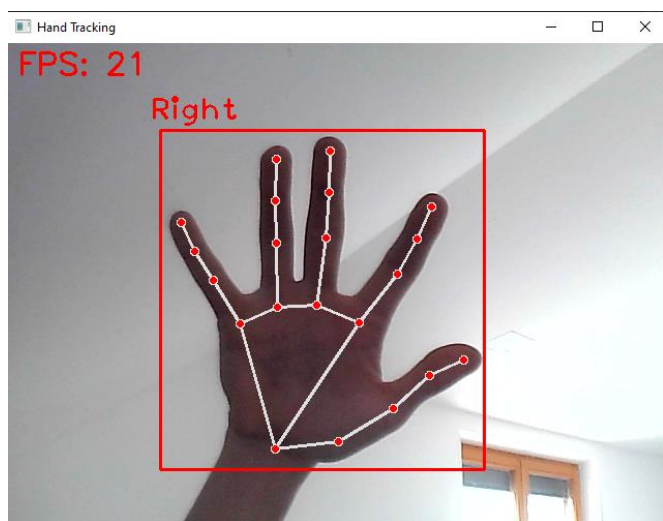
```

signal_str = ''.join(map(str, fingers))
signal_str = f"${signal_str}"

# Send signal to Arduino
try:
    ser.write(signal_str.encode())
except serial.SerialException as e:
    print(f"Error writing to serial: {e}")

```

zgledalo je tako



Slika 6: Prikazuje zaznavanje roke z kodo (Vir: lasten)

```
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
```

Slika 7: Prikaz Izpisa odprtih ali zaprtih prstov

po želji sma dodala še fps count

(frames per second, slike na sekundo)

```
import time

curr_time = time.time()
fps = 1 / (curr_time - prev_time)
```

```

    prev_time = curr_time
    cv2.putText(img, f'FPS: {int(fps)}', (10, 30),
               cv2.FONT_HERSHEY_SIMPLEX, 1, text_color, 2)

```

ter za izhod iz programa

```

    cv2.imshow("Hand Tracking", img)
    if cv2.waitKey(1) & 0xFF == 27:
        break

    ser.close()
    cap.release()
    cv2.destroyAllWindows()

```

### 3.4.2.2 PRIPRAVA ZA KODE NA ARDUINOTU

Najorej sem includal servo.h za delovanje pri pošiljanju signala

```
#include <Servo.h>
```

Definiral sma koliko znakov mora prejemati

```
#define numOfValsRec 5
```

```
#define digitsPerValRec 1
```

```
Servo servos[numOfValsRec];
```

```
int valsRec[numOfValsRec];
```

Tukaj se spremeni dolzina znakov saj se doda znak \$ v izpis

```
int stringLength = numOfValsRec * digitsPerValRec + 1;
```

```
int counter = 0;
```

```
bool counterStart = false;
```

```
String receivedString;
```





```
    if (num <= stringLength && num + digitsPerValRec <= stringLength) {  
        valsRec[i] = receivedString.substring(num, num + digitsPerValRec).toInt();  
    }  
}  
receivedString = "";  
counter = 0;  
counterStart = false;  
}  
}  
}  
}
```

Tukaj v loopu pa glede na to kakšen znak pridobi tako se odzove samo odprto ali zaprto

```
void loop() {  
    recieveData();  
    for (int i = 0; i < numOfValsRec; i++) {  
        if (valsRec[i] == 1) {  
            servos[i].write(180);  
        } else {  
            servos[i].write(0);  
        }  
    }  
}
```

## 4. REZULTATI

Povezana roka so odlično in hitro odzove na gibanje človeške roke, ki jih zazna kamera. In potem te informacije pošlje naprej. Arduino nato signale naprej pošlje v servo motorje kateri se premaknejo zdravstvo vrtijo glede na to, kakšen je signal arduina.

### 4.1 ODZIV ROKE

Roka se natančno in hitro odzove na dejanja človeške roke. Roka se lahko odpira ali zapira in se rokuje.



Slika 8: Prikazuje povezavo arduina z sestavljeno roko.

Razvoj in implementacija inteligentne povezave med računalniško vizijo in gibanjem robotske roke bo omogočila zmanjšanje nenatančnosti pri kirurških posegih ter izboljšanje rezultatov in varnosti operativnih postopkov.

## 5. RAZPRAVA

Hipoteza 1: Uporaba robotske roke v kirurgiji

Po temeljitem testiranju zmogljivosti sestavljene robotske roke smo prišli do ugotovitve, da roka ni dovolj močna in natančna, da bi lahko uspešno delovala v kirurških posegih. Težko je simulirati potrebno moč v pregibih ter doseči zadostno natančnost pri odpiranju in zapiranju. Najno hipotezo smo raziskala tudi po spletu in prišla do enake ugotovitve

Kot je avtor v tem članku omenil <https://www.frontiersin.org/articles/10.3389/frobt.2017.00065/full>

»There is currently no good solution to simulate the force with precise motions of fingers.«

torej prevedeno »Trenutno ni dobre rešitve za simulacijo sile z natančnimi gibi prstov.«

Našo hipotezo zanikamo.

## 6. POVZETEK

Raziskovalna naloga je raziskovala povezavo med računalniško vizijo in gibanjem robotske roke za izboljšanje kirurških postopkov. Kljub temeljitemu testiranju sestavljene robotske roke rezultati niso pokazali zadostne natančnosti in moči za uporabo v kirurgiji. Kljub temu pa je raziskava prinesla pomembno spoznanje o izzivih in omejitvah uporabe robotske tehnologije v medicini, ki lahko koristi pri nadaljnjem razvoju na tem področju.

## 7. ZAKLJUČEK

V zaključku naše projektne naloge o izdelavi robotske roke lahko jasno opazimo, da smo dosegli večje razumevanje in spretnosti na področju inženirstva, tehnologije in inovacij. Izdelava robotske roke ni bila le tehnični podvig, ampak tudi izjemno poučen proces, ki je zahteval sodelovanje, ustvarjalnost in vztrajnost. Naša robotska roka, čeprav še ni v popolnosti izpopolnjena, kaže velik potencial za prihodnji razvoj in uporabo v različnih okoljih. Poleg tega smo med delom na projektu pridobili tudi spretnosti v tiskem delu, problem reševanju, načrtovanju in izvedbi, ki nam bodo koristile v prihodnjih projektih in karieri. Skozi to projektno nalogo smo dokazali, da inovacija in tehnologija lahko prinesejo pozitivne spremembe in rešitve za izzive, s katerimi se sooča sodobna družba. Prepričani smo, da bo naša

robotska roka le ena od mnogih inovativnih rešitev, ki bodo prispevale k napredku in izboljšanju kakovosti življenja v prihodnosti.

V zaključku sva prišla do ugotovitve da se takšna robotska roka ne more uporabljati na mestu kirurga zaradi tega ker ne moraš nadzirati sile prstov. Roka ni dovolj natančna da bi zamenjala kirurgovo delo saj za to delo potrebuješ veliko natančnost.

## **8. ZAHVALA**

Vesela smo, da smo z raziskovalno nalogo dosegli, kot smo si zadali. Najprej bi se zahvalili mentorjema Urošu Remenihu in Samo Železniku za vso pomoč in podporo pri delu. Zahvaljujema se bližnim in družini za podporo pri najinem raziskovalnem delu.

## 9. PRILOGE

### PYTHON PROGRAM ZA ZAZNAVO ROKE TER POŠILJANJU SIGNALA V ARDUINO

```

import cv2
import time
import serial
from cvzone.HandTrackingModule import HandDetector

# Initialize serial communication with Arduino
ser = serial.Serial("COM3", 9600, timeout=0.5)

cap = cv2.VideoCapture(0)
detector = HandDetector(maxHands=1, detectionCon=0.7)

prev_time = 0
curr_time = 0

boundingbox_color = (0, 0, 255)
text_color = (0, 0, 255)
velikost_txt = 2

while True:
    success, img = cap.read()

    # Detect hands
    hands, img = detector.findHands(img)

    if hands:
        for hand in hands:
            fingers = detector.fingersUp(hand)
            print(fingers)

            # Construct the signal string
            signal_str = ''.join(map(str, fingers))
            signal_str = f"${signal_str}"

            # Send signal to Arduino
            try:
                ser.write(signal_str.encode())
            except serial.SerialException as e:
                print(f"Error writing to serial: {e}")

            # Draw bounding box and text on the image
            bbox = hand["bbox"]
            cv2.rectangle(img, (bbox[0] - 20, bbox[1] - 20), (bbox[0] +
bbox[2] + 20, bbox[1] + bbox[3] + 20), boundingbox_color, 2)
            cv2.putText(img, hand["type"], (bbox[0] - 30, bbox[1] - 30),
cv2.FONT_HERSHEY_PLAIN, velikost_txt, text_color, 2)

            # Calculate FPS
            curr_time = time.time()
            fps = 1 / (curr_time - prev_time)

```

```

    prev_time = curr_time
    cv2.putText(img, f'FPS: {int(fps)}', (10, 30), cv2.FONT_HERSHEY_SIMPLEX,
1, text_color, 2)

    cv2.imshow("Hand Tracking", img)
    if cv2.waitKey(1) & 0xFF == 27: # Press 'Esc' to exit the loop
        break

# Close the serial connection and release the camera
ser.close()
cap.release()
cv2.destroyAllWindows()

```

## C++ KODA V ARDUINOTU

```
#include <Servo.h>
```

```
#define numOfValsRec 5
```

```
#define digitsPerValRec 1
```

```
Servo servos[numOfValsRec]; // Array to store all servos
```

```
int valsRec[numOfValsRec];
```

```
int stringLength = numOfValsRec * digitsPerValRec + 1;
```

```
int counter = 0;
```

```
bool counterStart = false;
```

```
String receivedString;
```

```
void setup() {
```

```
Serial.begin(9600);
```

```
// Attach each servo to a pin
```

```
for (int i = 0; i < numOfValsRec; i++) {
```

```
    servos[i].attach(7 + i); // Assuming servo pins are consecutive (e.g., 7, 8, 9, 10, 11)
```

```
}
```

```
}
```

```
void recieveData() {
```

```
    char c;
```

```
    while (Serial.available()) {
```

```
        c = Serial.read();
```

```
        Serial.print(c);
```

```
        if (c == '$') {
```

```
            counterStart = true;
```

```
        }
```

```
        if (counterStart) {
```

```
            if (counter < stringLength) {
```

```
                receivedString += c;
```

```
                counter++;
```

```
            }
```



```
    if (counter >= stringLength) {  
        for (int i = 0; i < numOfValsRec; i++) {  
            int num = (i * digitsPerValRec) + 1; // Adjust the indexing  
            if (num <= stringLength && num + digitsPerValRec <= stringLength) {  
                valsRec[i] = receivedString.substring(num, num + digitsPerValRec).toInt();  
            }  
        }  
        receivedString = "";  
        counter = 0;  
        counterStart = false;  
    }  
}  
}  
}  
}  
  
void loop() {  
    recieveData();  
  
    // Control each servo based on the corresponding value in valsRec  
    for (int i = 0; i < numOfValsRec; i++) {  
        if (valsRec[i] == 1) {  
            servos[i].write(180); // Rotate to 180 degrees if valsRec[i] is 1  
        } else {
```

```
servos[i].write(0); // Rotate to 0 degrees if valsRec[i] is 0
}
}

// Print received values for debugging
for (int i = 0; i < numOfValsRec; i++) {
    Serial.print("valsRec[" + String(i) + "]: " + String(valsRec[i]) + " ");
}

Serial.println(); // Print a newline to separate values

}
```

## 10. VIRI IN LITERATURA

Za izdelavo modela robotske roke smo uporabili predlogo inmoov.fr, ki predstavlja enega izmed najbolj priljubljenih odprtih virov za DIY (naredi sam) projekte na področju robotike. Ta vir smo izbrali zaradi njegove priljubljenosti med ustvarjalci ter razpoložljivosti podrobnih navodil in modelov za izdelavo. Pri nadaljnjem razvoju in prilagajanju smo se osredotočili na specifične potrebe našega projekta, ob upoštevanju možnosti prilagoditve, ki jih ponuja ta platforma

Slika 1: Prikaz logotipa ogrodja Python  
<https://upload.wikimedia.org/wikipedia/commons/thumb/c/c3/Python-logo-notext.svg/115px-Python-logo-notext.svg.png>

Slika 2: Prikaz logotipa platforme Arduino <https://brandslogos.com/wp-content/uploads/images/large/arduino-logo-1.png>

Slika 3: Prikazuje stojalo za servo motorje. (Vir: lasten)

Slika 4: prikazuje nameščene motorčke v notranji del roke. (Vir: lasten)

Slika 5: Servo motorji povezani na svoje mesto. (Vir: lasten)

Slika 6: Prikazuje zaznavanje roke z kodo. (Vir: lasten)

Slika 7: Prikaz Izpisa odprtih ali zaprtih prstov. (Vir: lasten)

Slika 8: prikazuje povezavo arduina z sestavljeno roko. (Vir: lasten)

RAZPRAVA <https://www.frontiersin.org/articles/10.3389/frobt.2017.00065/full>

MATERIAL <https://inmoov.fr//hand-and-forarm/>