

Upravljanje računalom pokretima prstiju uz podršku računalnog vida

Đerek, Marko

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, Faculty of Science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:166:889331>

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

Download date / Datum preuzimanja: **2024-10-20**

Repository / Repozitorij:

[Repository of Faculty of Science](#)



UNIVERSITY OF SPLIT



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

ZAVRŠNI RAD

**UPRAVLJANJE RAČUNALOM POKRETIMA
PRSTIJU UZ PODRŠKU RAČUNALNOG VIDA**

Marko Đerek

Split, rujan 2024.

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET

ZAVRŠNI RAD

**UPRAVLJANJE RAČUNALOM POKRETIMA
PRSTIJU UZ PODRŠKU RAČUNALNOG VIDA**

Marko Đerek

Mentor: Prof. dr. sc. Andrina Granić

Split, rujan 2024.

Temeljna dokumentacijska kartica

Završni rad

Sveučilište u Splitu
Prirodoslovno-matematički fakultet
Odjel za informatiku
Ruđera Boškovića 33, 21000 Split, Hrvatska

UPRAVLJANJE RAČUNALOM POKRETIMA PRSTIJU UZ PODRŠKU RAČUNALNOG VIDA

Marko Đerek

SAŽETAK

Računalni vid je grana umjetne inteligencije koja se bavi razvojem sustava sposobnih za obradu i interpretaciju informacija iz digitalnih slika, videa ili drugih vizualnih medija. Dok ljudski vid ima prednost trodimenzionalnog opažanja i prepoznavanja objekata na temelju iskustva, nedavni napreci u dubokom učenju značajno su unaprijedili sposobnost računalnog vida da prepoznaje i razumije vizualne podatke. Jedan od najizazovnijih zadataka u računalnom vidu je prepoznavanje objekata, koje se najčešće primjenjuje na prepoznavanje lica, očiju i ruku. Prepoznavanje ruke predstavlja posebne poteškoće zbog fleksibilnosti ruke, sličnosti između prstiju te širokog raspona poza koje ruka može zauzeti. Ovi čimbenici čine prepoznavanje ruke složenim, ali važnim područjem istraživanja, osobito u kontekstu interakcije čovjeka i računala. Ovaj rad fokusira se na prepoznavanje ruke pomoću web kamere, gdje se dijelovi ruke prepoznaju na temelju članaka kostura, metodom koja se često koristi u okvirima računalnog vida poput MediaPipe-a. Cilj rada je pronaći i testirati optimalan način interakcije ruke, odnosno prstiju, s računalom, s krajnjim ciljem zamjene tradicionalne uporabe miša.

Ključne riječi: računalni vid, prepoznavanje ruke, upravljanje računalom, multimodalna interakcija

Rad je pohranjen u knjižnici Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Rad sadrži: 33 stranice, 20 grafičkih prikaza, 0 tablica i 14 literaturnih navoda. Izvornik je na hrvatskom jeziku.

Mentor: **Dr. sc. Andrina Granić**, redoviti profesor Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Ocjenjivači: **Dr. sc. Andrina Granić**, redoviti profesor Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Dr. sc. Goran Zaharija, docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Antonela Prnjak, asistent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Rad prihvaćen: rujan, 2024.

Basic documentation card

Thesis

University of Split
Faculty of Science
Department of Informatics
Ruđera Boškovića 33, 21000 Split, Croatia

Movement-based Computer Control Supported by Computer Vision

Marko Đerek

ABSTRACT

Computer vision is a branch of artificial intelligence that focuses on developing systems capable of processing and interpreting information from digital images, videos, or other visual media. While human vision has the advantage of three-dimensional perception and object recognition based on experience, recent advancements in deep learning have significantly improved computer vision's ability to recognize and understand visual data. One of the most challenging tasks in computer vision is object recognition, commonly applied to detecting faces, eyes, and hands. Hand recognition presents unique difficulties due to the flexibility of the hand, the similarity between fingers, and the wide variety of poses a hand can assume. These factors make hand recognition a complex yet important area of study, particularly in human-computer interaction. This thesis focuses on hand recognition using a webcam, where hand parts are identified based on skeletal joints, a method commonly employed in computer vision frameworks like MediaPipe. The aim of the work is to find and test the optimal way of interacting with a computer using hand, specifically finger, movements, ultimately offering a potential replacement for the traditional use of a mouse.

Key words: computer vision, hand recognition, computer control, multimodal interaction

Thesis deposited in library of Faculty of science, University of Split

Thesis consists of: 33 pages, 20 figures, 0 tables and 14 references
Original language: Croatian

Mentor: **Andrina Granić, Ph.D.** Professor of Faculty of Science, University of Split

Reviewers: **Andrina Granić, Ph.D.** Professor of Faculty of Science, University of Split
Goran Zaharija, Ph.D. Assistant Professor of Faculty of Science, University of Split

Antonela Prnjak, Assistant of Faculty of Science, University of Split

Thesis accepted: September, 2024.

Sadržaj

Uvod	1
1. Računalni vid	2
1.1. Povijest računalnog vida	3
1.2. Sadašnjost i budućnost računalnog vida	4
2. Prepoznavanje ruke	7
2.1. Građa ruke	7
2.2. Metode prepoznavanja ruke	8
2.2.1. Prepoznavanje ruke pomoću rukavice	9
2.2.2. Prepoznavanje ruke pomoću računalnog vida	11
3. Korištene tehnologije	15
3.1. OpenCV	15
3.2. MediaPipe	18
4. Implementacija računalnog vida	20
4.1. Identifikacija problema	20
4.2. Rješenje problema	20
4.3. Odabir interakcije	21
4.4. Implementacija	22
4.5. Primjena interakcije	25
Zaključak	26
Literatura	27
Skraćenice	29
Privitak	30

Uvod

U današnjem svijetu smo okruženi računalima i pametnim telefonima, kroz zadnjih 20 godina prešli su iz uloge nekakvih pomoćnih uređaja do toga da su nam neophodni u svakodnevnoj interakciji sa svijetom. Iako su uređaji i njihova korisnost nezamislivo napredovali, naš način interakcije s njima baš i nije.

Glavna interakcija s uređajima je taktilna, no interakcija je limitirana zbog toga što moramo dirati neku podlogu da bi interakcija bila moguća, bio to zaslon uređaja, miš ili nešto drugo. Česta tema u znanstvenoj fantastici je budućnost gdje je moguće koristiti uređaje samo preko pokreta prstiju gdje uređaj prepoznaje ruke i nije potrebna dodirna podloga. Zbog nedostatka tehnologije, prvobitno ovakva vrsta interakcije je zahtijevala dodatne uređaje što bi je činilo neučinkovitim i većinom ti uređaji nisu baš bili efikasni.

Zbog velikog napretka tehnologije u nekoliko posljednjih godina, ponajviše razvoja AI-a, ova interakcija će možda u bliskoj budućnosti napokon postati stvarnost.

U ovom radu je prikazan jednostavan sustav upravljanja korisničkim sustavom pomoću pokreta prstiju gdje je potrebna samo web kamera. Sustav pomoću računalnog vida prepoznaje ruku te označuje članke prstiju. Definirani članci kontroliraju pokazivač, njihovim pomicanjem se također pomiče pokazivač, dok dodirom članaka ostalih prstiju se izvode funkcije miša.

1. Računalni vid

Računalni vid je grana umjetne inteligencije (eng. *Artificial Intelligence*, AI) koja omogućuje računalima i sustavima da obrade značajne informacije iz digitalnih slika, videa ili drugih vizualnih medija i izvrše radnje ili daju preporuke na temelju tih informacija. Ako AI omogućuje računalima da razmišljaju, računalni vid im omogućuje da vide, promatraju i razumiju.

Računalni vid funkcionira gotovo isto kao i ljudski vid, bitna je razlika što ljudski vid ima prednost životnog iskustva za treniranje kako razlikovati objekte, koliko su udaljeni, kreću li se i postoji li nešto pogrešno na slici.

Zahtjevi računalnog vida su da obavi iste funkcije kao i ljudski vid, ali u mnogo kraćem vremenu s kamerama, podacima i algoritmima umjesto mrežnice, optičkih živaca i vidnog korteksa. Da bi bio u mogućnosti to ispuniti, računalni vid zahtjeva mnogo podataka. Iznova i iznova provodi analize podataka dok ne uoči razlike i konačno prepozna slike. Na primjer, da bi se računalo osposobilo za prepoznavanje automobilskih guma, potrebno mu je unijeti ogromne količine slika guma i predmeta povezanih s gumama kako bi naučilo razlike i prepoznalo gumu, posebno onu bez nedostataka.

Da bi se to postiglo koriste se dvije osnovne tehnologije: vrsta strojnog učenja koja se naziva duboko učenje i konvolucijska neuronska mreža (eng. *Convolutional Neural Network*, CNN).

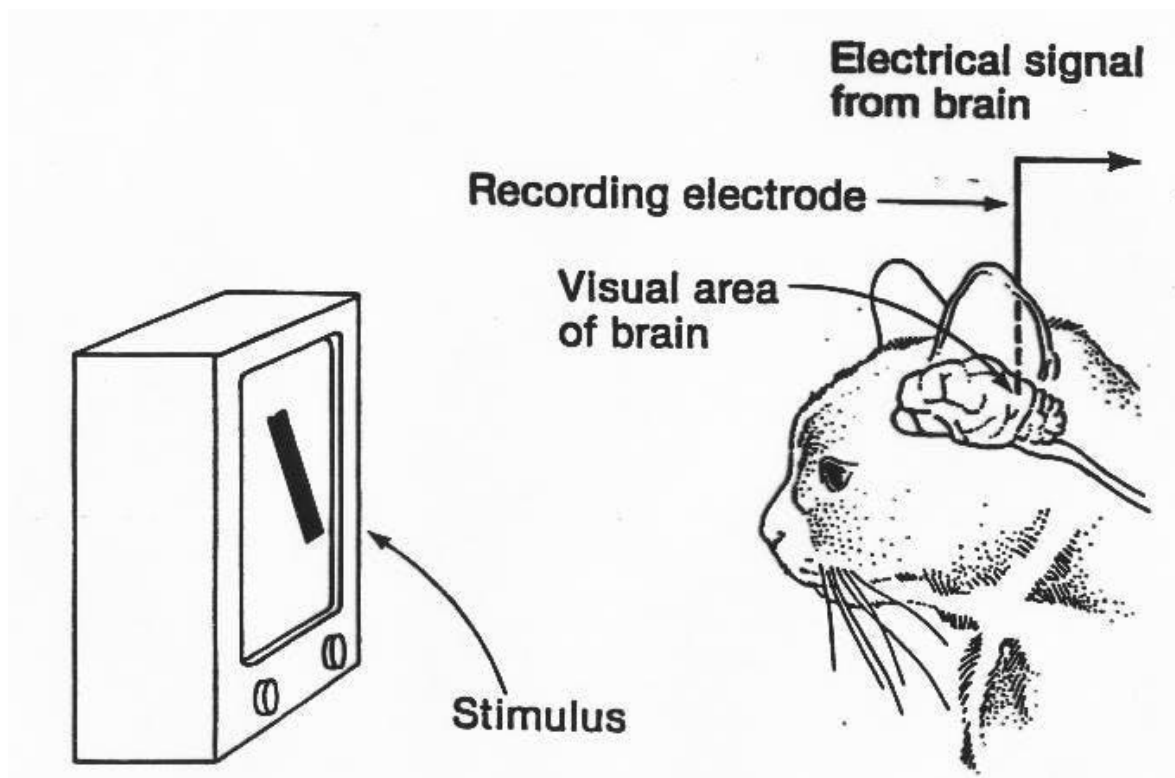
Neki od primjera gdje se danas sve računalni vid koristi su:

- Prepoznavanje lica i otisaka prstiju za biometrijsku identifikaciju i autentifikaciju
- Prepoznavanje govora i teksta za prevođenje, transkripciju i komunikaciju
- Prepoznavanje objekata i scena za navigaciju, prepoznavanja prometnih znakova i semafora, autonomnu vožnju i robotsku manipulaciju
- Prepoznavanje emocija i gesti za interakciju s ljudima, igre, umjetnu inteligenciju i virtualnu stvarnost
- Prepoznavanje bolesti i anomalija za dijagnostiku, praćenje i liječenje

- Prepoznavanje uzoraka i tekstura za klasifikaciju, segmentaciju i restauraciju slika
- Generiranje novih slika i videa za umjetnost, dizajn, animaciju i sintezu

1.1. Povijest računalnog vida

Znanstvenici i inženjeri već oko 60 godina pokušavaju razviti načine na koje strojevi mogu vidjeti i razumjeti vizualne podatke. Davne 1959. godine započelo je eksperimentiranje kada su neurofiziolozi mački pokazali niz slika pokušavajući povezati odgovor u njezinu mozgu, što se može vidjeti na slici 1.1. Otkrili su da prvo reagira na oštre rubove ili linije, a znanstveno to znači da obrada slike počinje jednostavnim oblicima poput ravnih rubova što je više opisano u [1].



Slika 1.1 Skica eksperimenta neurološke reakcije mačke

Otprilike u isto vrijeme razvijena je prva računalna tehnologija skeniranja slika, koja je računalima omogućila digitalizaciju i dobivanje slika. Još jedna prekretnica postignuta je 1963. godine kada su računala uspjela transformirati dvodimenzionalne slike u trodimenzionalne oblike. Nakon tih otkrića umjetna inteligencija se pojavila kao akademsko područje proučavanja, a također je označila početak potrage umjetne inteligencije za rješavanjem problema ljudskog vida što je opisano u [12].

Sljedeći bitan razvoj je tehnologija optičkog prepoznavanja znakova (OCR), koja je uvedena 1974. godine i mogla je prepoznati tekst ispisan bilo kojim fontom ili tipom slova. Slično tome, inteligentno prepoznavanje znakova (ICR) moglo je dešifrirati rukom pisani tekst pomoću neuronskih mreža. Od tada, OCR i ICR zajedno su našli svoj put u obradi dokumenata i računa, prepoznavanje tablica vozila, mobilna plaćanja, strojno prevođenje i druge uobičajene aplikacije.

Godine 1982. radovi dvaju znanstvenika su imali ogroman utjecaj na računalni vid. Neuroznanstvenik David Marr utvrdio je da vid funkcionira hijerarhijski i uveo algoritme za otkrivanje rubova, kutova, krivulja i sličnih osnovnih oblika. Istodobno, računalni znanstvenik Kuniyuki Fukushima razvio je mrežu stanica koje mogu prepoznati obrasce. Mreža, nazvana Neocognitron, uključivala je konvolucijske slojeve u neuronsku mrežu.

Skup podataka ImageNet postao je dostupan 2010. godine. Sadržavao je milijune označenih slika u tisuću klasa objekata i pruža temelj za CNN-ove i modele dubokog učenja koji se danas koriste. Godine 2012. tim sa Sveučilišta u Torontu prijavio je CNN na natjecanje u prepoznavanju slika. Model, nazvan AlexNet, značajno je smanjio stopu pogrešaka za prepoznavanje slike. Nakon ovog otkrića, stope pogrešaka pale su na samo nekoliko postotaka.

1.2. Sadašnjost i budućnost računalnog vida

Zbog rapidnog i znakovitog tehnološkog napretka, pogotovo u području umjetne inteligencije, koja se svakim danom sve više i više uključuje u naš svakodnevni život, računalni vid će sigurno biti bitan faktor u novonastalim ili nadograđenim interakcijama.

Računala više nisu samo uređaji koji primaju neki input i obnašaju nekakav zadatak, ona se sve više uključuju u vizualni svijet i pomoću računalnog vida, računala će u budućnost u potpunosti promijeniti svijet i mnogo olakšati kompleksne zadatke, možda čak i izvršavati ih bolje nego i sami ljudi.

Računalni vid bi imao veliki utjecaj na polja gdje se zahtjeva velika preciznost ili ima jako veliki broj podataka. Danas se najviše medijski prikazuje mogućnost računalnog vida da prepoznaje prometne linije i znakove kao što se može vidjeti na slici 1.2 što nas sve više

približava budućnosti gdje ćemo imati samovozeće automobile, što će stvoriti efektivniji i sigurniji promet.



Slika 1.2 Primjer računalnog vida u prometu

Područje u kojem će možda računalni vid biti najviše koristan je medicina. Medicina je vizualno područje s velikim brojem slika što je jako pogodno za računalni vid, jer ima potrebnu bazu slika iz koje bi algoritam mogao naučiti.

Načine na koje računalni vid može pomoći u medicini:

- Automatizacija procesa i smanjenje mogućnosti ljudske greške
- Precizniji i efikasniji pregledi
- Mogućnost promatranja pacijenta na daljinu
- Poboljšano zaustavljanje oboljenja i pravovremena prevencija
- Prepoznavanje bolesti i anomalija za dijagnostiku, praćenje i liječenje
- Mogućnost razvijanja alata za efikasnija i olakšana prepoznavanja bolesti

Mogućnost računalnog vida ne poznaje granice, gore navedeni primjeri su samo trenutno dvije grane za koje se smatra da bi najviše napredovale uz pomoć računalnog vida, no tu tehnologiju možemo koristiti za bilo koji aspekt svakodnevnog života, bilo to obrazovanje,

proizvodnja ili čak istraživanje svemira. Pošto sam naravno trenutno student i obrazovanje mi je trenutno najbliži entitet izrazito mi je zanimljivo na koje načine bi računalni vid mogao utjecati na različite dijelove obrazovanja, može to biti poboljšanje kvalitete nastave do same sigurnosti. Neki od primjera na koje načine bi računalni vid mogao poboljšati obrazovanje su:

- Analiziranje reakcija učenika tijekom nastave za mogućnost poboljšavanja nastave, predavanja ili čak razvijanje personalizirane nastave
- Praćenje učenika tijekom ispita što bi smanjilo razinu varanja
- Dodatno razvijanje kvalitete online nastave i uključenosti učenika u samo nastavu
- Poboljšana sigurnost nastave zbog kvalitetnije razine nadzora
- Mogućnost razvijanja interakcija koje omogućavaju veću razinu surađivanja između učenika i učitelja
- Automatizirano praćenje prisutnosti

Računalni vid ima mogućnost promijeniti svijet, na koji već trenutno ima popriličan utjecaj iako to možda još ne primjećujemo.

2. Prepoznavanje ruke

Prepoznavanje ruke je jedan od najbitnijih predmeta istraživanja u grani računalnog vida. Ruka sama po sebi je jako kompleksan objekt za prepoznati zbog dinamičnosti svojeg oblika koju omogućuje mobilnost prstiju. Računalu je puno lakše prepoznati lice gdje se svi prepoznatljivi dijelovi kao što su oči, nos, usta ili uši (skoro) uvijek nalaze na očekivanom mjestu. Također bitan faktor je sličnost prstiju što pravi izraziti problem računalu, gore navedeni dijelovi lica su lako prepoznatljivi računalu zbog svoje specifičnosti u bilo kojem trenutku, dok kod prstiju (izuzev palca) je skoro pa nemoguće računalu raspoznati o kojem točno prstu se radi.

2.1. Građa ruke

Šaka, kao dio gornjeg uda kod čovjeka, sastoji se od kostiju, zglobova, ligamenata, tetiva, mišića, živaca i krvnih žila, što je prikazano na slici 2.1. Šaku čini pet prstiju: palac, kažiprst, srednjak, prstenjak i mali prst. Na vrhu svakog prsta nalazi se nokat koji služi kao zaštita. Palac je poseban jer je usporedan s ramenom i može se zarotirati za 90 stupnjeva, dok ostali prsti mogu rotirati za 45 stupnjeva.

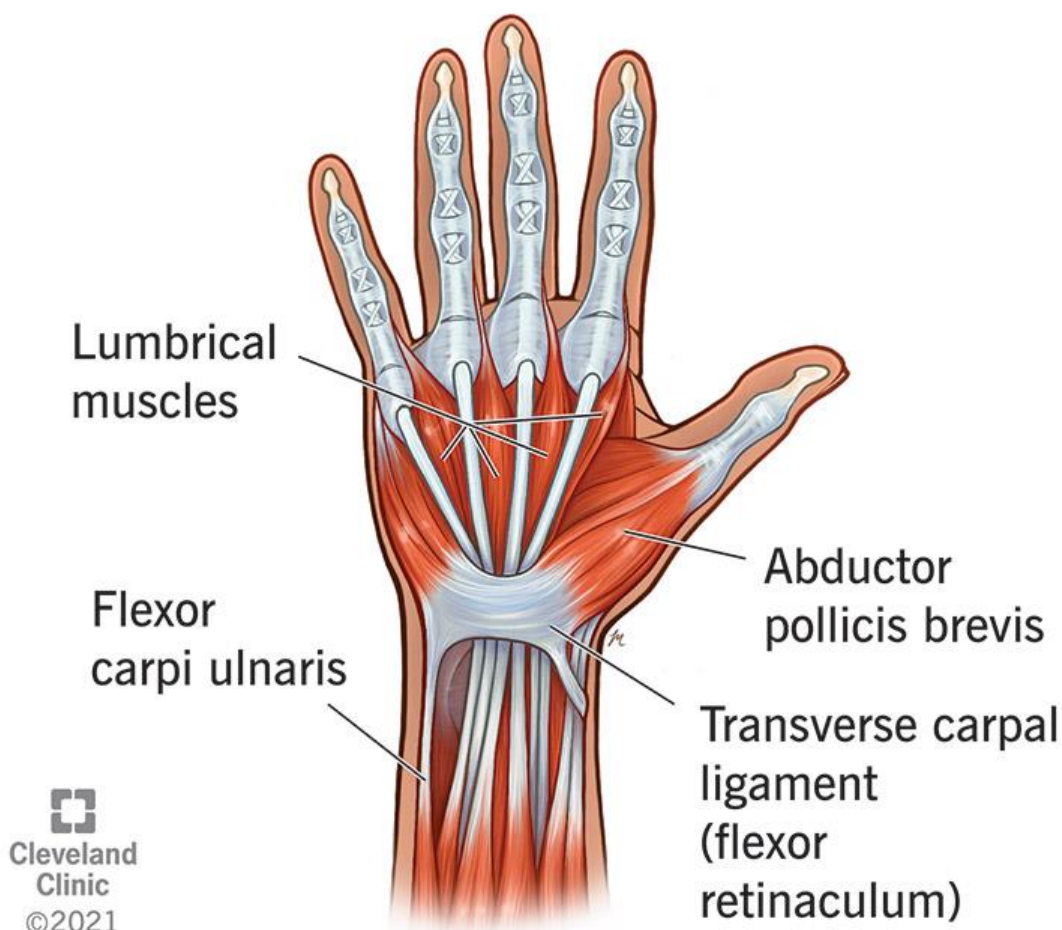
Šaka se sastoji od 27 kostiju: osam kostiju karpusa (pešća), pet kostiju metakarpusa (zapešća) i 14 falangi (članaka prstiju). U širem smislu, tu su i dvije kosti podlaktice: radijus (palčana kost) i ulna (lakatna kost). Titive mišića šake pripadaju fleksorima i opružaćima prstiju, koji su povezani s kostima podlaktice. Ova struktura omogućuje specifične pokrete palca i malog prsta kroz složen sistem ovojnica i kotura, koji omogućuju opružanje.

U šaci se nalaze mišići koji podržavaju najfleksibilnije, najpreciznije i najfinije pokrete palca i malog prsta. Šake su krajnji dio ruke i imaju ključnu ulogu u svakodnevnom životu. Omogućuju hvatanje i manipulaciju predmetima, što je temelj za mnoge ljudske aktivnosti, od osnovnih kao što je hranjenje, do složenih kao što su pisanje ili sviranje instrumenata. Palac igra ključnu ulogu u hvatanju i držanju predmeta, dok ostali prsti podržavaju te

kretnje. Šake su također iznimno važne za komunikaciju i izražavanje emocija. Više detalja o ruci može se pronaći u [6].

Anatomy of the hand and wrist

Palmar view of the right hand



Slika 2.1 Anatomija šake

2.2. Metode prepoznavanja ruke

Svrha prepoznavanja ruku je razvijanje sistema koji bi mogao ne samo prepoznati ruke nego na kraju i razumjeti pokrete ruke i prenijeti neku informaciju ili izvršiti komandu. Zbog kompleksnosti prepoznavanja ruku postoji veliki broj različitih metoda, dvije

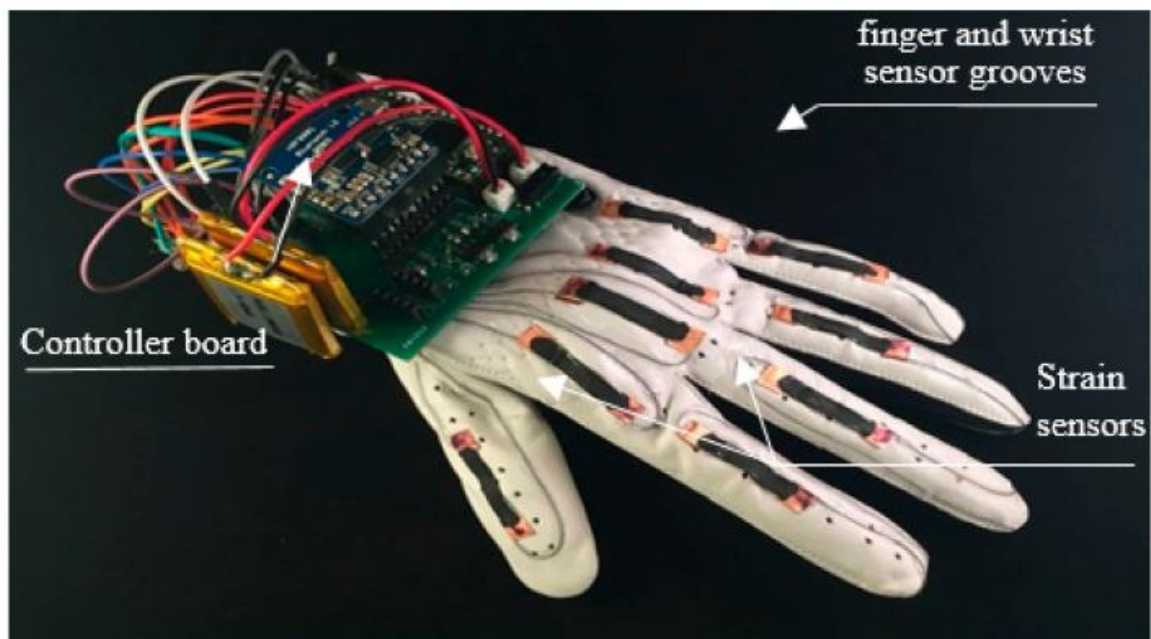
najpopularnije i najviše korištene metode bi bile prepoznavanje pomoću rukavice i prepoznavanje pomoću računalnog vida. Detaljno o razvoju metoda prepoznavanja ruke se može pronaći u [3; 5]

2.2.1. Prepoznavanje ruke pomoću rukavice

Ova metoda zahtjeva posebnu rukavicu koja bi računalu olakšala proces prepoznavanja i podigla razinu uspješnosti interakcije. Za ovakvu vrstu interakcije postoje dvije metode koje zahtijevaju različite tipove rukavica.

Prepoznavanje ruke pomoću rukavice sa sensorima:

Za ovu metodu je potrebna posebno napravljena rukavica koja sadrži senzore, njen izgled se može vidjeti na slici 2.2, ti senzori nam daju veliku preciznost i izraziti broj informacija o stanju u kojem se ruka nalazi. Pomoću rukavice dobivamo točne koordinate dlana ruke te položaj i orijentaciju prstiju. Problem kod ovog pristupa je što rukavica mora biti spojena s računalom što uvelike limitira interakciju korisnika te naravno sama rukavica je poprilično skupa, pa je teško da će se naći u komercijalnoj upotrebi u bliskoj budućnosti. Prednost modernog tipa ovakvih rukavica je mogućnost taktilne povratne informacije korisniku što mu daje mogućnost da osjeti teksturu ili oblik virtualnih objekata.



Slika 2.2 Rukavica sa sensorima

Prepoznavanje ruke pomoću rukavice s bojama:

Ova metoda zahtjeva rukavicu s različitim bojama i oblicima, jedan primjer te rukavice je prikazan na slici 2.3, njena svrha pojednostavljuje samo prepoznavanja računalu tako da pridoda raznolikosti samoj ruci te također olakša prepoznavanja svakog prsta pomoću personaliziranih boja. Prednost ove metode je što za razliku od rukavice sa sensorima, zahtjevi za ovu interakciju nisu skupi i ne bi bilo problema za komercijalnu upotrebu. Naravno limit ovakve interakcije je što nedostaje spontanosti kod interakcije zbog zahtjeva za rukavicom.



Slika 2.3 Primjer obojane rukavice

2.2.2. Prepoznavanje ruke pomoću računalnog vida

Za računalni vid je naravno potrebna video kamera preko koje bi računalo bilo u stanju „vidjeti“. Računalni vid omogućuje potpunu beskontaktnu interakciju s računalom, no također je mnogo kompleksniji zbog različitih prepreka kao što bi bile: kvaliteta senzora kamere, razina svjetlosti, sličnost s pozadinom, udaljenost od kamere... Zbog mnogolikih prepreka postoji veliki broj različitih pristupa i metoda u prepoznavanja ruke, pristup se većinom razlikuje u dubini i kompleksnosti te glavna podjela metoda bi bila na 2D i 3D metode.

2D metode su lakše za isprogramirati i izvesti, zahtijevaju manje učenja i pripremanja umjetne inteligencije, no manje su precizne i greške su češće, dok 3D metode su generalno kompleksnije, no ako su pravilno izvedene onda su većinom preciznije od 2D metoda.

Prepoznavanje ruke pomoću boje kože:

Jedna od najpopularnijih metoda prepoznavanja ruku, postoje dva načina na koja se može izvesti. Prvi način je prepoznavanje kože bazirana na pikselima, funkcionira tako što se svaki piksel zasebno provjerava te se klasificira kao koža ili ne što se može vidjeti na slici 2.4. Druga metoda je regionalno prepoznavanje kože, umjesto individualnog pregledavanja piksela, kod ove metode se pikseli prostorno definiraju po teksturi i intenzitetu. Nakon prepoznavanja se odabire jedan prostor boja i izolira se ruka od ostatka slike. Više o obje metode se može pročitati u [2].



(a)



(b)

Slika 2.4 Prepoznavanje ruke pomoću boje kože

Prepoznavanje ruke pomoću izgleda:

Za ovu metodu su potrebne slike iz kojih bi računalo učilo detalje modela koji se traži te bi se to uspoređivalo sa slikama koje se dobivaju iz videa. Da bi se ova metoda mogla izvesti u pravom vremenu zbog svoje jednostavnosti, umjesto građenja 3D modela iz videa, samo se izvlače 2D slike iz videa gdje se pronalaze traženi detalji. Metoda funkcionira na način da se računa intenzitet piksela kao što je kod prepoznavanja pomoću boje kože, no zbog već dobivenih informacija iz početne slike nije potrebno birati prostor boja kao što je prikazano na slici 2.5. Prednost ove metode je što može prepoznati različite boje kože.



Slika 2.5 Prepoznavanje ruke pomoću izgleda

Prepoznavanje ruke pomoću pokreta:

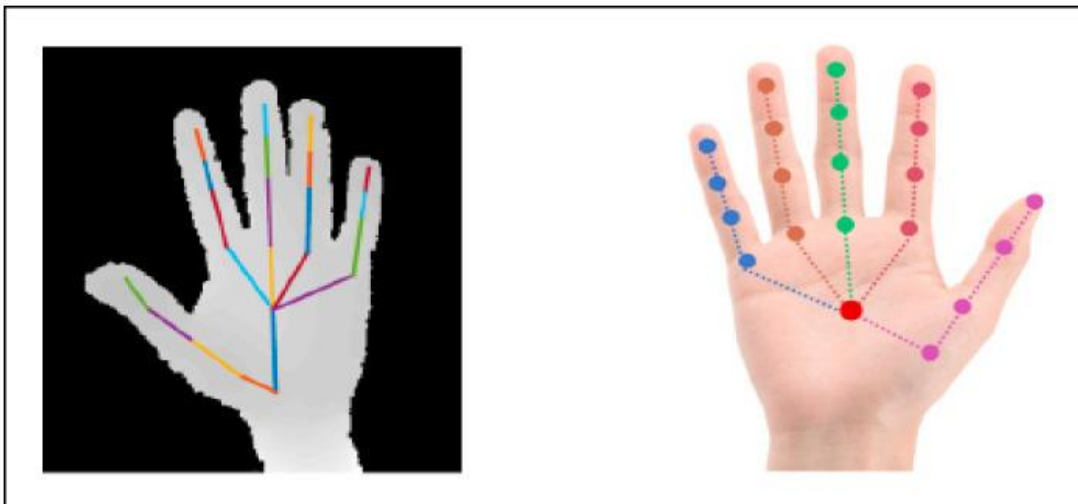
Metoda prepoznavanja ruke pomoću pokreta može biti jako korisna zbog svoje mogućnosti izvlačenja objekta kroz niz sličica, prikaz metode je vidljiv na slici 2.6. Za metodu potrebni su algoritmi prepoznavanja, karakterizacije, modeliranja pokreta i prepoznavanja uzorka da bi se mogao prepoznati pokret. Kod ove metode postoje dva velika problema, ako u istom trenutku se prikazuje više od jednog pokreta može se dogoditi da prepoznavanje neće pravilno raditi te ako je pozadina dinamična vrlo je teško računalo izdvojiti objekt od pozadine.



Slika 2.6 Prepoznavanje ruke pomoću pokreta

Prepoznavanje ruke pomoću strukture kostura:

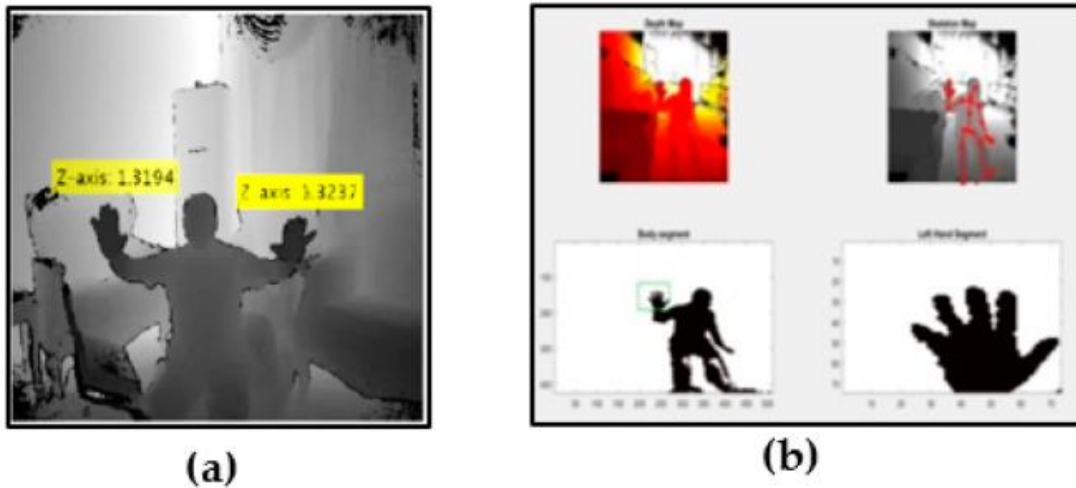
Struktura kostura se bazira na modelu prepoznavanja gdje se biraju dijelovi objekta koji mogu poboljšati kvalitetu prepoznavanja. Kod ruke najčešće korišteni dijelovi za prepoznavanja bi bili prsti, to jest članci prstiju kao na slici 2.7. Članci mogu računalu dati ogroman broj podataka koji se mogu pretvoriti u specifične parametre koji poboljšavaju prepoznavanja, neki od primjera bi bili pozicija članka u kosturu, udaljenost između članaka, orijentacija članka... Pomoću tih parametara računalo lakše razlikuje pojedinačne dijelove objekta, što su ovdje članci i tako dolazi do prepoznavanja cijele ruke.



Slika 2.7 Prepoznavanje ruke pomoću strukture kostura

Prepoznavanje ruke pomoću dubine:

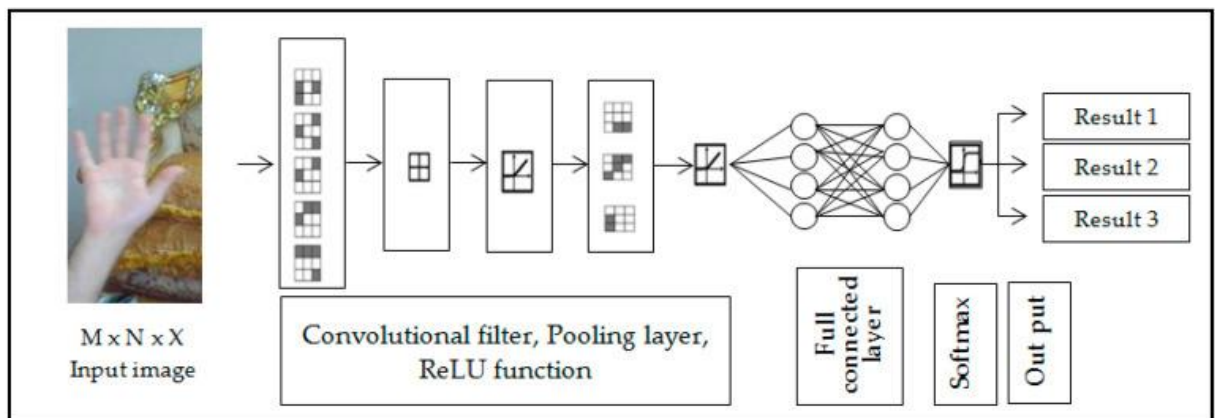
Nedostaci običnih kamera stvaraju velike probleme kod prepoznavanja ruke kao što je navedeno u prijašnjim primjerima, zato postoje i koriste se specifične kamere koje daju dodatne mogućnosti i olakšavaju samu prepoznavanja. Jedna od tih kamera je dubinska kamera koja računalu omogućuje 3D informacije o prostoru te samom objektu. Prednosti korištenja ovakve vrste kamere je što tipični problemi za 2D kamere kao što su razina svjetla ili boja nemaju nikakav utjecaj na mogućnost prepoznavanja što se može vidjeti na slici 2.8. Najveći nedostatak ovakve tehnologije bila bi sama kompleksnost kamere, kamere su prevelike i skupe za svakodnevnu uporabu te zbog toga ovakva metoda jednostavno nije isplativa za korištenje, kao gore navedene metode s rukavicama.



Slika 2.8 Prepoznavanje ruke pomoću dubine

Prepoznavanje ruke pomoću strojnog učenja:

Umjetna inteligencija se danas koristi u mnoštvu modernih aplikacija zbog svoje mogućnosti učenja, što je izrazito korisno u računalnom vidu. Prednost ove metode je što samo računalo uči kroz vrijeme što znači da sama razina prepoznavanja postaje točnija. Nedostatak ove metode osim potrebnog vremena treniranja bi također bio potreba za bazom podataka koja je potrebna za učenje što pridodaje vremenu koje je potrebno za procesiranje slike koju računalo prima. Kompleksnost mreže potrebne za ovakvu metodu je vidljiva na slici 2.9 dok se detaljnije može pronaći u [4].



Slika 2.9 Prepoznavanje ruke pomoću strojnog učenja

3. Korištene tehnologije

Trenutno postoje već razvijene tehnologije koje se ne samo služe računalnim vidom nego ga dodatno i unapređuju. Dvije tehnologije su korištene u ovom radu zbog svojih kvaliteta i korisnosti u računalnom vidu, osobito u prepoznavanju ruku.

Prva korištena tehnologija je OpenCV (eng. *Open Source Computer Vision Library*) koja je skup biblioteka specijalizirana za poboljšanje infrastrukture računalnog vida.

Druga korištena tehnologija je MediaPipe, okvir koji je razvijen za potrebe poboljšanja strojnog učenja i sadrži izrazito kvalitetan kanal za prepoznavanje ruku.

3.1. OpenCV

OpenCV je softverska biblioteka otvorenog koda za računalni vid i strojno učenje, logotip prikazan na slici 3.1. OpenCV je kreiran da bi osigurao zajedničku infrastrukturu za aplikacije računalnog vida te također ubrzao korištenje strojne percepcije u komercijalnim proizvodima. Biblioteka trenutno sadrži preko 2500 optimiziranih algoritama, što uključuje opsežan skup klasičnih i najsuvremenijih algoritama računalnog vida i strojnog učenja.

Algoritmi se mogu koristiti za razne svrhe: otkrivanje i prepoznavanje lica, identifikacija objekata, klasificiranje ljudskih radnji u videozapisima, praćenje pokreta kamere, praćenje pokretnih objekata, izdvajanje 3D modela objekata, proizvodnju 3D oblaka točaka iz stereo kamera, spajanje slika u jednu sliku visoke rezolucije, pronalaženje sličnih slika iz baze podataka slika, uklanjanje crvenih očiju iz fotografija, praćenje pokreta očiju, prepoznavanje krajolika, postavljanje oznaka za prekrivanje proširenom stvarnošću...

OpenCV trenutno broji preko 47 tisuća ljudi u svojoj zajednici i preko 18 milijuna preuzimanja, sama biblioteka se intenzivno koristi u tvrtkama, istraživačkim grupama i tijelima vlade. Neke od najvećih tvrtki koje koriste OpenCV su Google, Microsoft, Intel, IBM, Yahoo, Sony, Honda i Toyota te se također koristi za otkrivanje sumnjivaca u provalama preko nadzornih snimki, praćenja rudničke opreme u Kini, pomaganje robotima u navigaciji i skupljanju objekata u Willow Garage-u, otkrivanje nesreća utapanja u bazenima u Europskim državama, pokretanje interaktivne umjetnosti u Španjolskoj i New

Yorku, provjeri piste za krhotine u Turskoj, pregled naljepnica na proizvodima u tvornicama diljem svijeta do brzog prepoznavanja lica u Japanu.

OpenCV podržava C++, Python, Java i MATLAB sučelja i podržava Windows, Linux, Android te Mac Os što bi bile svi glavni operativni sustavi današnjice. Svi detalji o OpenCV-u se mogu pronaći u dokumentaciji u [7].



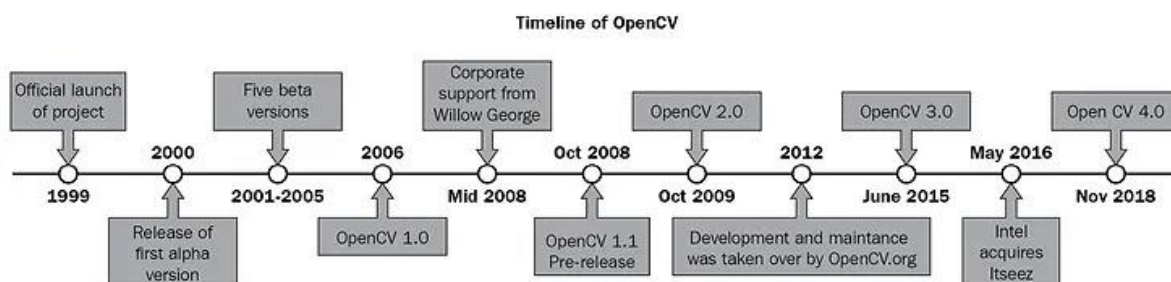
Slika 3.1 Logo OpenCV-a

OpenCV kao projekt je službeno lansiran 1999. godine od strane Intel Research inicijative da se dodatno razviju CPU zahtjevne aplikacije kao dio serije projekata kao što je *ray-tracing* u stvarnom vremenu. Početna ideologija OpenCV-a bila je:

- Unaprijediti istraživanje računalnog vida pružanjem otvorenog te ponajviše optimiziranog koda za infrastrukturu vida i odstraniti potrebu za kodiranjem cijelog projekta iz ničega u budućnosti.
- Proširiti znanje o vidu pomoću razvijanja zajedničke infrastrukture koji bi developeri koristili za naknadnu izgradnju što bi učinilo kod čitljivijim i lakše prenosivim.
- Unaprijediti komercijalne aplikacije temeljene na računalnom vidu koristeći kod koji je prenosiv, optimiziraniji i najbitnije dostupan besplatno s licencom bez zahtjeva da finalni kod bude otvorenog tipa ili besplatan.

Prva *alpha* verzija OpenCV-a bila je objavljena javnosti 2000. godine, a prva stabilna verzija 2006. godine.

Od 2012. godine podršku za OpenCV je preuzela bez profitna organizacija OpenCV.org koja se brine za održavanje OpenCV-a, ponajviše web stranica. 2016. godine Intel i Itseez dogovorili su suradnju gdje će Itseez biti vodeći developeri OpenCV-a. Najbitniji događaji u povijesti OpenCV-a se mogu vidjeti na slici 3.2.



Slika 3.2 Povijest OpenCV-a

Zbog svoje prilagodljivosti OpenCV se može koristiti u skoro svakom aspektu današnje tehnologije što pojašnjava milijunske brojke preuzimanja i upotrebu kod bitnih i kompleksnih projekata kod nekih od najvećih država svijeta. Polja tehnologije gdje se OpenCV koristi su:

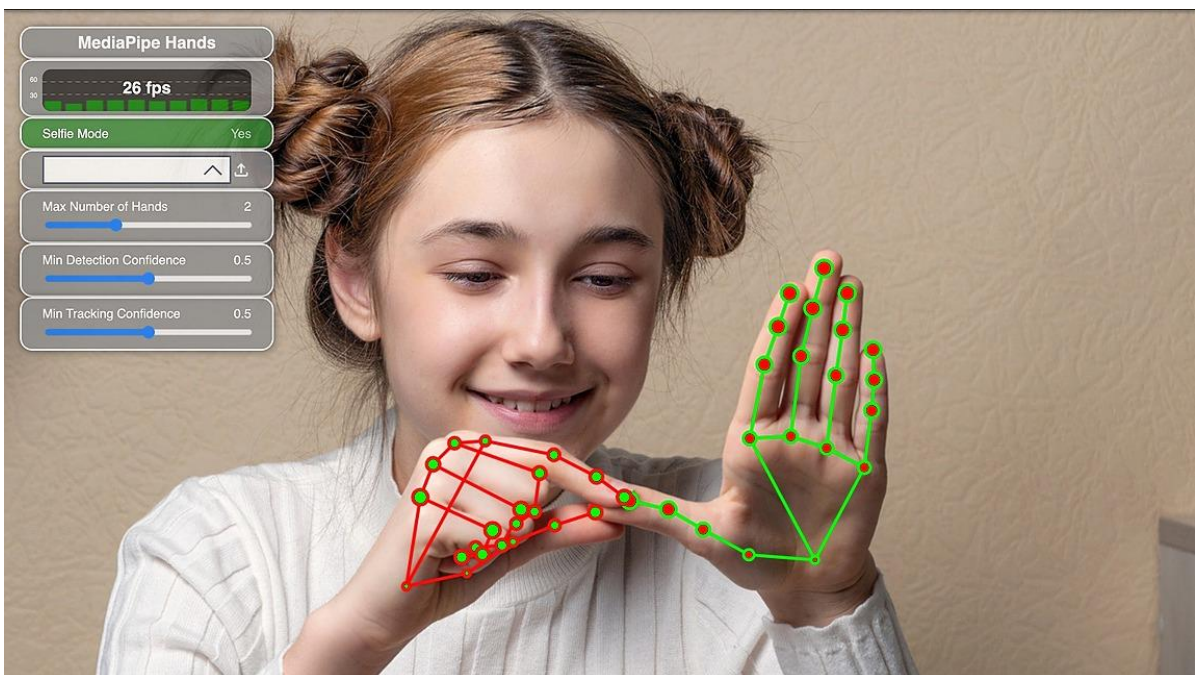
- Dronovi – Pomoću OpenCV-a dronovi mogu pratiti i analizirati objekte što je izrazito korisno u poljoprivredi i pogotovo kod pronalazaka nestalih osoba u izazovnim uvjetima.
- Mrežni uređaji – Uređaji koji su međusobno povezani na mrežu često koriste mogućnosti vida kao što bi bile nadzorne kamere koje prate i prepoznaju objekte, pametni frižideri koji prate kvalitetu i rok trajanja hrane, semafori koji prate gustoću prometa...
- Automobili – U pomoćnim sistemima proizvođači automobila često koriste OpenCV za značajke kao prepoznavanje cestovnih linija, prepoznavanje pješaka, izbjegavanje sudara...
- Zdravstvo – OpenCV se trenutno u zdravstvu koristi za otkrivanje anomalija, praćenje stanja pacijenata te pomaže kirurzima tijekom postupka operacija tako što ih navodi kroz slike.

3.2. MediaPipe

MediaPipe je okvir otvorenog koda razvijen od strane Google-a za izgradnju i implementaciju kanala za strojno učenje. Ti kanali mogu u obraditi različite multimedijske podatke poput teksta, video i zvuka. MediaPipe sadrži unaprijed izgrađene komponente poznate kao „kalkulatori“ koji se mogu međusobno kombinirati kako bi se stvorili kanali za računalni vid. Detaljnije opisano u [9].

MediaPipe djeluje poput okvira programiranja podataka, podaci se kreću kroz graf povezanih „kalkulatora“ gdje svaki kalkulator obavlja zadatak na podacima prije nego što se pošalju na sljedeći. Primjer korištenja je vidljiv na slici 3.2.

Kanali stvoreni pomoću MediaPipe-a mogu se pokrenuti na web aplikacijama, pametnim telefonima (Android i iOS) te malim ugrađenim sustavima. *Cross-platform* mogućnosti omogućuju programerima da stvore imerzivne i responzivne aplikacije za bilo koji uređaj.



Slika 3.2 Primjer rada MediaPipe-a

Počeci MediaPipe-a se mogu pronaći u ranim 2010-ima kad je Google radio na poboljšanjima tehnologija strojnog učenja i računalnog vida. MediaPipe je prvi put korišten 2012. godine za analizu videa i zvuka u stvarnom vremenu na YouTube-u.

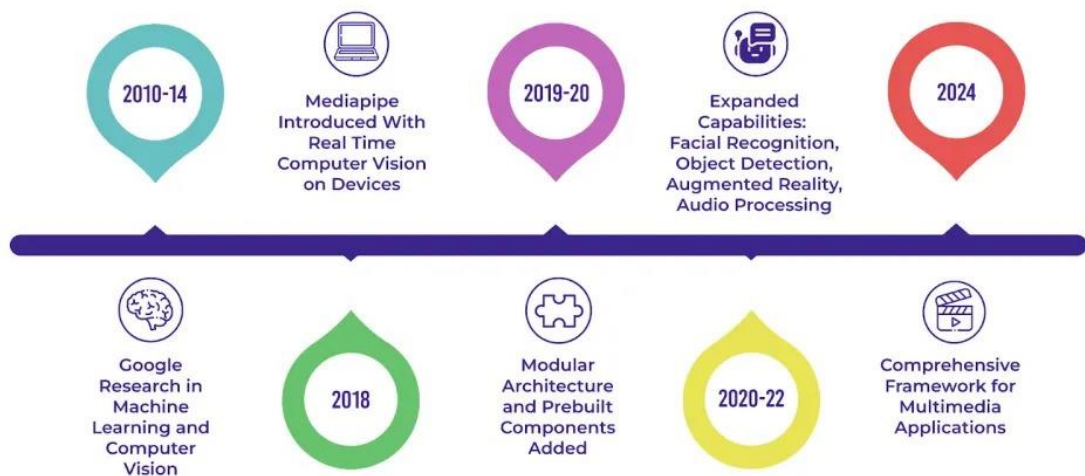
U 2018. godini, MediaPipe je počeo rješavati probleme povezane s korištenjem složenih modela računalnog vida na uređajima poput pametnih telefona i malih računala. Od 2019.

godine, MediaPipe je dostupan javnosti, otvarajući nove mogućnosti za istraživače i razvojne programere.

Početak 2020. godine povećavala se potreba za brzim i učinkovitim načinom obrade multimedije što je poteklo ažuriranje MediaPipe-a.

Trenutno MediaPipe stoji kao snažan okvir za programere koji žele stvoriti inovativne multimedijske aplikacije koje kvalitetno funkcioniraju.

Svi bitni trenutci razvoja povijesti MediaPipe-a su prikazani na slici 3.3.



Slika 3.3 Povijest MediaPipe-a

4. Implementacija računalnog vida

4.1. Identifikacija problema

Cilj rada je razviti program koji će prepoznati ruku te omogućiti da se pomoću ruke i web kamere upravlja računalom. Ruka će upravljati pokazivačem miša te zamijeniti funkcije miša. Potrebno je definirati pokrete ruku to jest prstiju koje će računalo prepoznati za definiranu akciju.

4.2. Rješenje problema

Razrada rješenja problema je definirana na slici 4.1.

1. Potrebno je definirati radnu površinu, prozor kamere se otvara u specifičnim dimenzijama te je potrebno podesiti radnu površinu da korisnik može doseći sve dijelove ekrana bez da ruka izlazi iz okvira videa, to ćemo dobiti računama dijeljenjem širine ekrana sa širinom prozora videa te množenjem s udaljenosti koju točka prsta mora prijeći, postupak ponavljamo za visinu.
2. Prepoznavanje ruke. Kod prepoznavanja ruke svaki prst ima točke po sebi koje predstavljaju članke kostura sa svojim zasebnim indeksom, potrebno je odabrati indekse za dijelove prstiju koji se koriste u interakciji.
3. Određivanje prstiju koji su potrebni za interakcije. Za rad je potrebno 5 akcija: kretnja pokazivača, lijevi klik, desni klik, klizanje prema gore, klizanje prema dolje. Da bi svaka akcija imala zasebnu interakciju će biti potrebno koristiti svih 5 prstiju.
4. Određivanje interakcije. Potrebno je odrediti interakcije za navedene akcije, po osobnom osjećaju najlakše interakcije prstima koje zahtijevaju najmanje mišićnog rada su dodiri ostalih prstiju s palcem.
5. Izvođenje navedenih akcija.



Slika 4.4.1 Razrada problema

4.3. Odabir interakcije

Da bi bili u mogućnosti kontrolirati računalo pomoću pokreta prstiju potrebno je odabrati, to jest „pronaći“ najkvalitetniju interakciju. Da bi mogli koristiti sve značajke miša što bi bilo kretnja i zaustavljanje pokazivača, lijevi i desni klik i klizanje gore i dole, potrebna je „kompleksnija“ interakcija s barem dva prsta da bi se mogle pokriti sve navedene interakcije, a u isto vrijeme izbjeci nenamjerno aktiviranje interakcije, interakcija mora biti jasna i namjerna. Da bih bolje razumio kako pristupiti interakciji istražio sam na koje načine su izvedene slične ideje, gdje su me najviše inspirirala dva projekta koja se nalaze u [13; 14].

Naredba koja se najviše koristi je kretnja pokazivača, potrebno je napraviti razliku u interakciji kad se pokazivač pomiče ili ne da se izbjegne nenamjerno pomicanje što bi zbunilo korisnika i natjeralo ga da izgubi lokaciju pokazivača.

Prvobitna ideja je bila podizanje kažiprsta za pokretanje kretnje pokazivača što se koristi za neke multimodalne interakcije. Nastavak na to bi bio dodir kažiprsta i srednjeg prsta za klik, no tu je odmah bilo očito da će biti problema, sama interakcija je poprilično neprirodna i zahtijeva previše mišićne energije za nešto što bi se koristilo u većem vremenskom periodu. Također zbog odabira da samo jedan prst kontrolira pokazivač to znači da kažiprst se ne smije kretati i izoliran je iz bilo koje interakcije što dovodi do

potrebe da se interakcije sastoje od primjera kao što su dodir srednjeg prsta i prstenjaka ili prstenjaka i malog prsta što su apsolutno neprirodne interakcije koje se jedva mogu izvršiti s potpunim fokusom i nemoguće je da se koriste kao interakcije s ponavljanjem.

Nakon razmatranja primijetio sam da u svakodnevnici najviše koristim palac u dodiru s nekim prstom, odličan primjer toga je na koji način ljudi drže olovku.

S obzirom na tu saznanje, odabrana je interakcija gdje će se koristiti palac s nekim drugim prstom. Pošto lijevi klik se najčešće koristi nakon kretanje pokazivača potrebno je odabrati interakcije koje su bliske tako da se korisnik ne bi umarao i da bi mu interakcija bila razumljiva.

Odabrana je interakcija gdje će se pokazivač kretati tek kad su palac i kažiprst bliski u određenoj udaljenosti te se događa lijevi klik miša ako se ta dva prsta dotaknu. Za desni klik odabrana je interakcija dodira srednjeg prsta i palca, dok za klizanje su odabrana ostala dva prsta u dodiru s palcem, za gornje klizanje dodiruju se palac i prstenjak, a za donje palac i mali prst. Interakcija je tako postala mnogo prirodnija i lakša za izvesti pa ne dovodi do velikog umora prstiju.

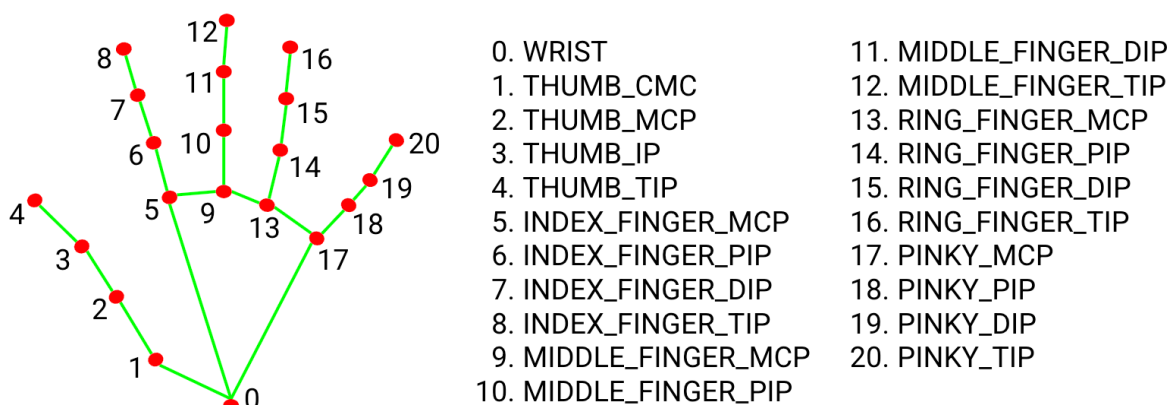
4.4. Implementacija

Rad je napisan u programskom jeziku Python uz pomoć modula OpenCV i MediaPipe.

OpenCV kao što je navedeno je biblioteka otvorenog koda za računalni vid i koristi se u ovom radu zbog svoje jednostavnosti i optimiziranosti. Ovaj modul je posebno koristan zbog svojih performansi, da bi se interakcija mogla izvoditi u pravom vremenu potrebno je da je program stabilan da bi bio u stanju pratiti u svakom trenutku pokrete korisnika te također da je dovoljno optimiziran da ne zahtijeva previše kompjuterskih resursa, jer računalo ne smije usporiti u nekom trenutku ili biti sporo kod obrade interakcije, to ruši kvalitetu interakcije i zbunjuje korisnika. OpenCV kao jedan od najrazvijenijih i najpopularnijih modula u računalnom vidu se nosi s tim problemima vrlo dobro i nametnuo se kao odlično rješenje.

MediaPipe se koristi zbog svojih izrazito kvalitetnih algoritama prepoznavanja ruke koji su prikazani u [10]. Kao što je navedeno u 2. točki ovog rada prepoznavanje ruke je poprilično kompleksan zadatak za koji ne postoji savršen ili najbolji pristup te zbog toga postoje mnogolike metode. MediaPipe koristi algoritam koji se sastoji na poveznici dviju

navedenih metoda koje bi bile prepoznavanje ruke pomoću strukture kostura i prepoznavanje ruke pomoću strojnog učenja gdje se strojno učenje koristi za poboljšanje kvalitete prepoznavanja strukture kostura. Modul je izabran prvo zbog svoje fantastične sposobnosti prepoznavanja ruke s vrlo malim problemima čak i u najtežim situacijama, također ne zahtjeva previše sistemskih resursa što nam je jako bitno za fluidnost interakcije. Dodatna kvaliteta MediaPipe-a je što nam zbog svoje metode prepoznavanja omogućuje da indexiramo što se može vidjeti na slici 4.2, to jest označimo samo jedan članak prsta koji nam je potreban za interakciju i tako pridodaje preciznosti interakcije. Više o tome se može pročitati u [8].



Slika 4.2 Indexi šake

Još se koristi modul PyAutoGUI koji omogućava korištenje komandi sustava preko koda, što je potrebno za komande miša kao primjer na slici 4.3 gdje se aktivira kretanje miša kad određena točka što je su u ovom slučaju vršci prstiju palca i kažiprsta dođu u određenu udaljenost. Više o samom modulu se može pronaći u [11].

```
elif abs(index_y - thumb_y) < 100:
    pyautogui.moveTo(index_x, index_y)
```

Slika 4.3 Kod koji aktivira kretanju pokazivača

Kad se ruka pojavi program automatski prikaže točkice na prstima i dijelu dlana koji predstavljaju članke kao što se može vidjeti na slici 4.4. Vrhovi prstiju koji se koriste za akcije imaju svaki krug s zasebnom bojom da bi bilo korisniku lakše raspoznati što će

koristiti u interakciji. Pokazivač je zamrznut dok se ne približe kažiprst i palac te kad se to dogodi pokazivač će se maknuti na koordinate gdje se kažiprst nalazi.

Na slici 4.5 se može vidjeti jedan primjer interakcije, što bi bilo dodir palca i kažiprsta, kod njihovog dodira događa se lijevi klik miša, za sve ostale akcije miša imamo jednake interakcije s palcem zbog jednostavnosti interakcije i najmanjeg mišićnog umora. Srednji prst kod dodira s palcem proizvodi desni klik miša, prstenjak klizanje prema gore i mali prst klizanje prema dole.



Slika 4.4 Primjer prepoznavanja ruke



Slika 4.5 Primjer interakcije dodira prstiju

4.5. Primjena interakcije

Interakcija kao što se može vidjeti je vrlo specifična i zbog toga neupotrebljiva u svakodnevnoj upotrebi, no mogla bi biti izrazito korisna za neakve specifične uporabe. Smatram da bi najviše pomogla za potrebe izlaganja gdje je da bi se omogućilo prezentiranje s obje ruke, potrebna još jedna osoba koja kontrolira izlaganje ili uređaj koji je potrebno držati u ruci što smanjuje kontrolu izlaganja i samu kvalitetu izlaganja. Što bi značilo da bi ovakva interakcija doista mogla poboljšati obrazovanje, javne govore, prezentacije, nastupe... Takva vrsta događaja također ima savršene uvjete za sakriti sve nedostatke tehnologije kao što bi bili promjene u svjetlu, različite udaljenosti, kvaliteta kamere... To su nedostaci koji dolaze do izražaja u svakodnevnom životu, ali u kontroliranim uvjetima kao što to većinom i jesu u navedenim aktivnostima ti nedostaci bi bili prisutni samo u ekstremnim slučajevima. S napretkom korištenih modula i tehnologije smatram da ovakav tip interakcije ima svoje mjesto u budućnosti.

Zaključak

U današnjici gdje umjetna inteligencija postaje sve više ukomponirana u svakodnevni život, računalni vid postaje jako bitan alat za daljnje razvijanje tehnologije i projekte budućnosti. Interakcije pomoću pokreta ili gesti se mogu koristiti u razne svrhe i čak poboljšati trenutne interakcije. Sama tehnologija je tek u počecima te se i same ideje gdje se sve računalni vid može koristiti može koristiti postepeno razvijaju.

Ovaj rad je fokusiran na interakcije s računalom, primarno kontrolama miša pomoću šake, to jest prstiju. Šaka zbog svoje fleksibilnosti omogućuje velik broj različitih pokreta i interakcija koje bi se mogle koristiti u svrhe kontrole računala pomoću računalnog vida.

Problematika kod šake je kompleksnost prepoznavanja, za razliku od lica šaka nema toliko unikatnih dijelova koji bi mogli pripomoći računalu kod prepoznavanja. Najčešće se za interakcije koristi web kamera što pridonosi problemima kod prepoznavanja, kvaliteta prepoznavanja ovisi o kvaliteti kamere, pozadini iza subjekta, razini svjetlosti u prostoriji, udaljenosti subjekta od kamere, broju subjekata pred kamerom...

Trenutno u komercijalnim svrhama najčešće se ovakva vrsta interakcija koristila u multimedijskoj industriji za svrhe zabave. Iako je tehnologija izrazito zanimljiva trenutno postoji previše tehnoloških ograničenja za širu upotrebu, upotrebljivost je na zadovoljavajućoj razini samo kad su uvjeti u potpunosti savršeni te dovodi do bržeg umora za razliku od uobičajenih interakcija zbog kompleksnijih zahtjeva za interakciju.

Zbog navedenih problema smatram da trenutno tehnologija nije dovoljno razvijena za svakodnevno korištenje, ali sadrži ogroman potencijal i siguran sam da će pronaći svoje mjesto u svakodnevnoj upotrebi u budućnosti.

Osobno vidim korisnost ovakvog tipa interakcije ponajviše u obrazovnom sistemu, gdje su uvjeti uvijek kontrolirani što znači da uvijek ima dovoljno svjetla, subjekt je uvijek na određenoj udaljenosti od kamere, nema smetnji od drugih subjekata i sama predavanja traju određeni vremenski period što znači da umor neće previše utjecati, a dodatne mogućnosti predavanja koje nam ovakav beskontaktni tip interakcije omogućuje bi pridonijelo zanimljivoj nastavi.

Literatura

[1] Hubel D. H.; Wiesel T. N.; Receptive fields of single neurones in the cat's striate cortex. *J. Physiol.* **1959.**, 148, str. 574-591.

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1363130/?ref=hackernoon.com>

pristupljeno dana 01.07.2024

[2] Ma'asum F. F. M.; Sulaiman S., Saparon Azilah; An Overview of Hand Gestures Recognition System Techniques, *IOP Conference Series: Materials Science and Engineering* 99 **2015.**

<https://iopscience.iop.org/article/10.1088/1757-899X/99/1/012012> pristupljeno dana 05.07.2024.

[3] Oudah, M.; Al-Naji, A.; Chall. J; Hand Gesture Recognition Based on Computer Vision: A Review of Techniques, *J. Imaging* **2020.**, 6, 73.

<https://doi.org/10.3390/jimaging6080073> pristupljeno dana 10.07.2024.

[4] Sen A.; Mishra T. K.; Dash R.; Deep learning based Hand gesture recognition system and design of a Human-Machine Interface, *Cornell University* **2023.**

<https://arxiv.org/abs/2207.03112> pristupljeno 10.07.2024

[5] Sharma P. K.; Sharma S. Evolution of Hand Gesture Recognition: A Review, *International Journal Of Engineering And Computer Science (IJECS)* **2015.**, Vol. 4, str. 9962-9965.

<https://www.ijeecs.in/index.php/ijeecs/article/view/447> pristupljeno dana 05.07.2024.

[6] <https://www.enciklopedija.hr/clanak/saka> [Mrežno] pristupljeno 07.07.2024

[7] <https://opencv.org/> [Mrežno] pristupljeno 15.07.2024

[8] https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker [Mrežno] pristupljeno 15.07.2024

[9] <https://github.com/google-ai-edge/mediapipe> [Mrežno] pristupljeno 15.07.2024

[10] <https://ai.google.dev/edge/mediapipe/solutions/guide> [Mrežno]

pristupljeno 15.07.2024

[11] <https://pyautogui.readthedocs.io/en/latest/> [Mrežno] pristupljeno 15.07.2024

[12] <https://www.ibm.com/topics/computer-vision> [Mrežno] pristupljeno 01.07.2024

[13] <https://github.com/Viral-Doshi/Gesture-Controlled-Virtual-Mouse> [Mrežno]

pristupljeno 15.07.2024

[14] https://github.com/ProgrammingHero1/virtual_mouse [Mrežno]

pristupljeno 15.07.2024

Skraćenice

CNN	<i>Convolutated neural network</i>	konvolucijska neuronska mreža
AI	<i>Artifial intelligence</i>	umjetna inteligencija
ICR	Intelligent character recognition	pametno prepoznavanje znakova
OCR	Optical character recognition	optičko prepoznavanje znakova
CPU	Computer processing unit	središnja procesorska jedinica

Privitak

Instalacija programske podrške

Uz Python 3.12.5 potrebno je instalirati pakete opencv-python, mediapipe i pyautogui i pozvati ih u kodu.

Kod u Pythonu

```
import cv2
import mediapipe as mp
import pyautogui
cap = cv2.VideoCapture(0)
hand_detector = mp.solutions.hands.Hands()
drawing_utils = mp.solutions.drawing_utils
screen_width, screen_height = pyautogui.size()
index_y = 0
middle_y = 0
ring_y = 0
pinky_y = 0
while True:
    _, frame = cap.read()
    frame = cv2.flip(frame, 1)
    frame_height, frame_width, _ = frame.shape
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    output = hand_detector.process(rgb_frame)
    hands = output.multi_hand_landmarks
    if hands:
        for hand in hands:
            drawing_utils.draw_landmarks(frame, hand)
            landmarks = hand.landmark
            for id, landmark in enumerate(landmarks):
                x = int(landmark.x*frame_width)
                y = int(landmark.y*frame_height)
                if id == 8:
                    cv2.circle(img=frame, center=(x,y),
radius=10, color=(255, 0, 255))
                    index_x = screen_width/frame_width*x
                    index_y = screen_height/frame_height*y
```

```

        if id == 12:
            cv2.circle(img=frame, center=(x,y),
radius=10, color=(0, 255, 255))
            middle_x = screen_width/frame_width*x
            middle_y = screen_height/frame_height*y

        if id == 16:
            cv2.circle(img=frame, center=(x,y),
radius=10, color=(200, 0, 100))
            ring_x = screen_width/frame_width*x
            ring_y = screen_height/frame_height*y

        if id == 20:
            cv2.circle(img=frame, center=(x,y),
radius=10, color=(255, 255, 255))
            pinky_x = screen_width/frame_width*x
            pinky_y = screen_height/frame_height*y

        if id == 4:
            cv2.circle(img=frame, center=(x,y),
radius=10, color=(255, 255, 0))
            thumb_x = screen_width/frame_width*x
            thumb_y = screen_height/frame_height*y
            if abs(index_y - thumb_y) < 30 and
abs(middle_y - thumb_y) > 30 and abs(ring_y - thumb_y) > 30
and abs(pinky_y - thumb_y) > 30:
                pyautogui.click()
                pyautogui.sleep(1)
            if abs(index_y - thumb_y) > 30 and
abs(middle_y - thumb_y) < 30 and abs(ring_y - thumb_y) > 30
and abs(pinky_y - thumb_y) > 30:
                pyautogui.rightClick()
                pyautogui.sleep(1)
            if abs(index_y - thumb_y) > 30 and
abs(middle_y - thumb_y) > 30 and abs(ring_y - thumb_y) < 30
and abs(pinky_y - thumb_y) > 30:
                pyautogui.scroll(300)
                pyautogui.sleep(1)

```

```
        if abs(index_y - thumb_y) > 30 and
abs(middle_y - thumb_y) > 30 and abs(ring_y - thumb_y) > 30
and abs(pinky_y - thumb_y) < 30:
            pyautogui.scroll(-300)
            pyautogui.sleep(1)
        elif abs(index_y - thumb_y) < 100:
            pyautogui.moveTo(index_x, index_y)

cv2.imshow('Virtual Mouse', frame)
cv2.waitKey(1)
```