SVEUČILIŠTE U ZAGREBU

**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 6354

**Sustav za raspoznavanje znakovnog jezika**

Nikola Tomažin

Zagreb, lipanj 2019.



Sadržaj

[1. Uvod v](#__RefHeading___Toc1179_4002615392)

[2. Podaci viii](#__RefHeading___Toc1181_4002615392)

[2.1. Prikupljanje podataka viii](#__RefHeading___Toc1183_4002615392)

[2.2 Izdvajanje ruke ix](#__RefHeading___Toc1185_4002615392)

[2.2.1. Histogram boje ruke ix](#__RefHeading___Toc1187_4002615392)

[2.2.2. Oduzimanje pozadine x](#__RefHeading___Toc1189_4002615392)

[2.3 Nadopunjavanje skupa podataka xi](#__RefHeading___Toc1191_4002615392)

[Implementacija sustava xiii](#__RefHeading___Toc1193_4002615392)

[3.1 Općenito o strojnom učenju xiii](#__RefHeading___Toc1195_4002615392)

[3.1.1 Neuron xiv](#__RefHeading___Toc1197_4002615392)

[3.1.2 Neuronska mreža xv](#__RefHeading___Toc1199_4002615392)

[3.1.3 Konvolucijska neuronska mreža(*Convolutional Neural Network* - CNN) xvi](#__RefHeading___Toc1201_4002615392)

[3.1.4 Povratna neuronska mreža(*Recurrent Neural Network* – RNN) xvii](#__RefHeading___Toc1203_4002615392)

[3.2 Arhitektura mreže xvii](#__RefHeading___Toc1205_4002615392)

[3.2.1. Prvi pristup - Konvolucijska neuronska mreža xvii](#__RefHeading___Toc1207_4002615392)

[3.2.2. Drugi pristup – 3D konvolucijska neuronska mreža i povratna neuronska mreža(RNN) xx](#__RefHeading___Toc1209_4002615392)

[3.3. Treniranje mreže xx](#__RefHeading___Toc1211_4002615392)

[4. Rezultati xxii](#__RefHeading___Toc1213_4002615392)

[4.1. Problemi xxvi](#__RefHeading___Toc1215_4002615392)

[4. Instalacija i upute za korištenje xxvii](#__RefHeading___Toc1217_4002615392)

[4.1 Potrebne instalacije (*Requirements*) xxvii](#__RefHeading___Toc1219_4002615392)

[4.2 Pokretanje xxviii](#__RefHeading___Toc1221_4002615392)

[4.2.1 Dodavanje geste xxix](#__RefHeading___Toc1223_4002615392)

[4.2.2 Treniranje mreže xxx](#__RefHeading___Toc1225_4002615392)

[4.2.3 Prepoznavanje geste xxx](#__RefHeading___Toc1227_4002615392)

[5. Zaključak xxxiii](#__RefHeading___Toc1229_4002615392)

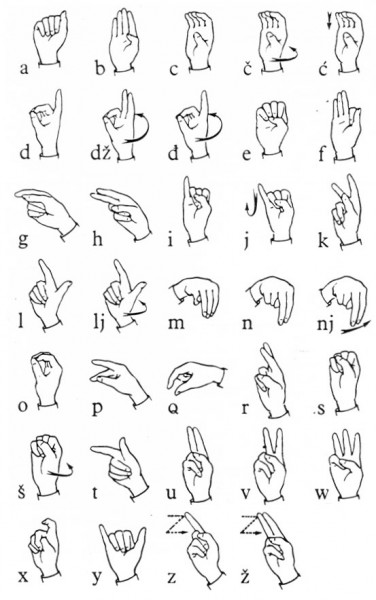
[6. Literatura xxxiii](#__RefHeading___Toc1231_4002615392)

# Uvod

U Republici Hrvatskoj službeni je jezik hrvatski, a službeno pismo latinica. Gluhih i nagluhih osoba u Republici Hrvatskoj uvijek je bilo, no njihov način jezik, njihov način komunikacije(sa zajednicom i društvom) u zakonima pojavljuje se tek 2015. godine. Hrvatski znakovni jezik(HZJ) sustav je vizualnih znakova koji, uz pomoć posebnog položaja(oblika šake), orijentacije, položaja i smjera pokreta ruke, tvore koncept odnosno smisao slova i riječi. Koristi se najčešće u obiteljima u kojima ima gluhih, zajednicama gluhih te, nešto manje u školama za gluhe.

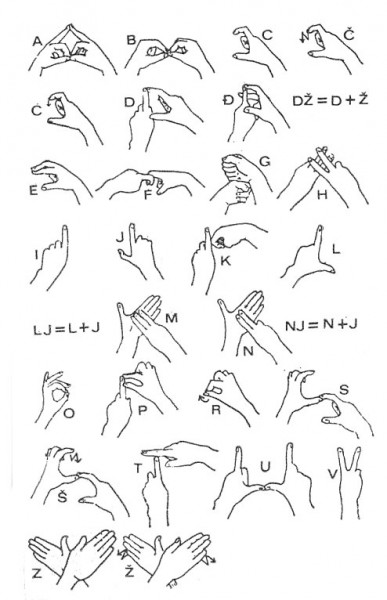
Postoje dvije vrste hrvatske znakovne abecede:

* Jednoručna abeceda hrvatskog znakovnog jezika koja je zapravo verzija američke(internacionalne) jednoručne abecede.  Proširena je onim slovima koja se koriste samo u našem jeziku. U osnovi, oponaša mala tiskana slova. Uglavnom se koristi u školama za gluhu djecu jer se tako na učinkovit način prenose riječi i informacije



Slika 1: Primjer jednoručne abecede hrvatskog znakovnog jezika

Dvoručna abeceda hrvatskog znakovnog jezika koja je posebno određena položajima prstiju obiju ruku, a oponaša velika tiskana slova hrvatske abecede. Dvoručna abeceda ima dugu tradiciju u zajednici gluhih. U dvoručnoj se abecedi koriste veliki pokreti ruku pa je moguće da se dvoručnom abecedom koriste i gluhonijeme osobe s značajnim ostatkom vida.



Slika 2: Primjer dvoručne abecede hrvatskog znakovnog jezika

Cilj ovoga završnoga rada napraviti je sustav koji bi prepoznavao geste znakovne abecede (jednoručne i dvoručne) te još neke geste/znakove u stvarnom vremenu (engl. *real-time*) te omogućio komunikaciju gluhonijeme osobe s nekime tko primjerice ne zna znakovni jezik/abecedu. Sustav je širok pojam, ovdje  on obuhvaća niz procesa i radnji, počevši od prikupljanja podataka(*dataseta*), obrada tih podataka (izdvajanje bitnoga), treniranje i testiranje neuronske mreže na problemu te konačni rezultat koji preko kamere interpretira znakove koje mu pokazujemo te nam “prevodi” HZJ u slova abecede.

Prvi dio rada obuhvaća prikupljanje podataka, obradu istih te način oblikovanja (kako ih pripremit za mrežu). Drugi dio odnosi se na neuronske mreže te teoriju u njihovoj pozadini. Bit će opisana arhitektura konvolucijske neuronske mreže(CNN - *Convolutional Neural Network*) te pokušaj upotrebe arhitekture povratne neuronske mreže(RNN - *Recurent Neural Network*) za analiziranje dinamičkih pokreta. Treći dio obuhvaća analizu rezultata, probleme kod sličnih slova, probleme dinamičkih slova te instalaciju i uputstva za uporabu.

# 2. Podaci

Najbitnija stvar kod strojnog učenja je imati dovoljno veliki i smislen skup podataka(engl. *dataset*). Bez kvalitetnih podataka rezultati mogu biti neprecizni te nema koristi od takve mreže. Kod odabira podataka mora se mnogo faktora uzeti u obzir, primjerice veličina i kvaliteta slike, osvjetljenje, uzima li se cijela slika ili dio nje, pozicija/poza objekta kojeg promatramo… Na sreću, postoje već gotovi skupovi podataka koji su dizajnirani za određene zadatke i probleme, kao što su praćenje objekata, detekcija, procjena poze, prepoznavanje akcija itd. Veličina skupova podataka također može biti raznolika, od nekoliko stotina slika, za jednostavne probleme do više stotina tisuća podataka kod kompleksnih i širokih problema.

## 2.1. Prikupljanje podataka

Pri prvotnom prikupljanju podataka za raspoznavanje znakovnog jezika uzet je skup podataka (referenca) koji sadrži 40 različitih gesti. Skup je sadržavao znakove američke jednoručne znakovne abecede, brojeve od 1 do 9 te neke dodatne znakove. Svaka gesta bila je prikazana kroz 2400 slika: 1200 polaznih slika te njihova zrcalna varijanta kako bi se dobila mogućnost prikazivanja geste neovisno gleda li se sa prednje ili stražnje strane što omogućava promatranje pomoću prednje i stražnje kamere. Slike su bile crno-bijele (kontrast je napravljen bijelim znakom šake / dlana na crnoj podlozi), dimenzija 50x50 piksela.



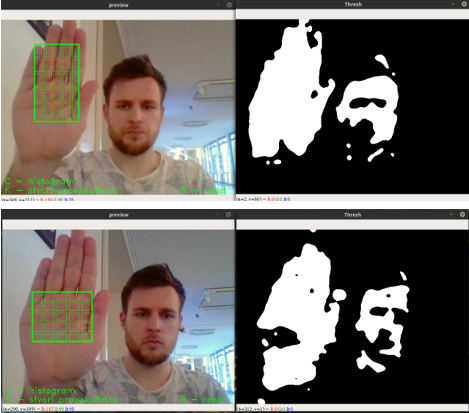
Slika 3: Prikaz skupa podataka

## 2.2 Izdvajanje ruke

Budući da je skup podataka crno bijela slika dlana/šake, treba tome prilagoditi ulaz (prikaz geste).

### 2.2.1. Histogram boje ruke

Prvi pristup tom problemu je bila metoda histograma boje ruke(engl. *Hand hist*). U ovom pristupu se prilikom početku snimanja(?) postavi dlan na određen prostor prikaza kamere, te se onda uzme spektar boja koji se nalazi u tom određenom prostoru. Sav spektar boje kože postavi se u bijelo, a sve izvan toga spektra u crno. Nakon što dobijemo zadovoljavajući histogram možemo ga spremiti te dalje koristiti kod prepoznavanja. Problem u ovom pristupu je da uvjeti snimanja moraju biti “savršeni” kako bi se dobio 'čist' histogram. Svaka promjena u osvjetljenju, nastajanje sjene te ne uzimanje u spektar sve nijanse boje kože dovodile su do nepotpunog histagrama (Slika 4). Ova metoda nije davala zadovoljavajuće rezultate jer se nije poklapala sa već postojećim skupom podataka, a i nije bila precizna, ovisila je o previše parametara.

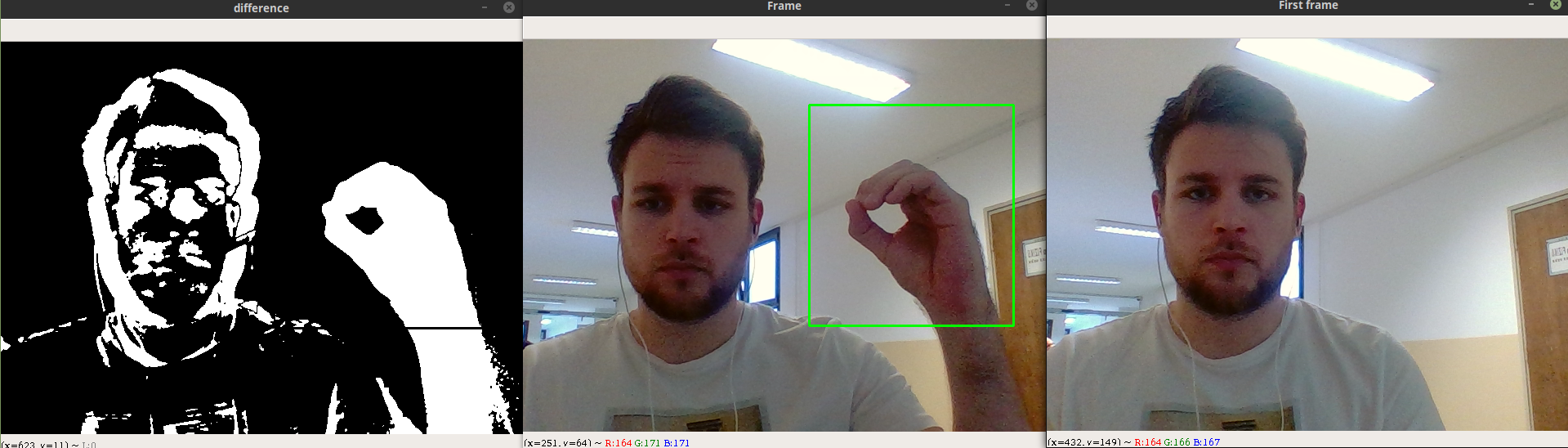


Slika 4: Prikaz dohvata histograma ruke

### 2.2.2. Oduzimanje pozadine

Drugi pristup je preko “oduzimanja pozadine” (engl. background substraction). Oduzimanje pozadine je široko korištena metoda koja se uglavnom koristi za dobivanje maske prvog plana (engl. foreground mask), odnosno objekata koji se pomiču u sceni koristeći statične kamere. Sastoji se od dva dijela, od inicijalizacije pozadine te od ažuriranja pozadine. U inicijalizaciji pozadine uzima se početni model pozadine, koji u sebi sadrži samo statične elemente dok se u drugom koraku svaki kadar (engl. frame) uspoređuje s inicijalnim modelom pozadine te se detektiraju razlike.

U ovom pristupu na ovaj problem inicijalizacija pozadine obavlja se pri paljenju kamere, gdje se uzima prvi kadar, a ažuriranje pozadine se gleda samo na određenom dijelu prikaza kamere, omeđenom pravokutnikom, jer taj dio nam predstavlja ulaz u mrežu, odnosno dio koji uspoređujemo i gledamo. Ova metoda se pokazala zadovoljavajućom jer ne ovisi o vanjskim uvjetima, ovisi jedino o pretpostavci da je kamera statična. Kod dinamične kamere pozadina mora biti jednolična (npr. Jednobojni zid) kako bi ova metoda funkcionirala.

Prvi dio slike predstavlja prvi snimljen kadar, drugi dio predstavlja prikaz kamere i treći dio predstavlja izlaz metode oduzimanja pozadine na temelju prve dvije slike.

Slika 5: Prikaz metode oduzimanja pozadine

## 2.3. Nadopunjavanje skupa podataka

Kako se hrvatska i američka jednoručna abeceda imaju neke razlike te kako postoji i dvoručna hrvatska znakovna abeceda trebalo je povećati skup podataka. Program “add\_gesture.py” dopunjen je skup podataka. Program kreira direktorij gdje će se spremiti geste te uključuje kameru sa standardnim zelenim pravokutnikom kao pokazivačem koji dio se promatra i sa prozorom koji prikazuje što računalo „vidi”. Pritiskom na tipku „c”, uz uvjet da je površina geste zadovoljavajuća, pokreće se snimanje gesti. Program snima 1200 slika i njegov napredak se može pratiti u kutu prikaza kamere. Korisnik svakog trenutka može pauzirati/nastaviti snimanje ponovnim pritiskom na tipku „c”, a snimanje se ujedno i pauzira čim površina geste manja od dozvoljene. Snimanje može potrajat do minute. Nakon snimanja slika se zrcali pomoću metode „flip\_images” te se konačno dobije 2400 slika nove geste.



Slika 6: Prikaz nadopunjavanja skupa podataka

Nakon nadopunjavanja, skup podataka trenutno sadrži 64 geste, gdje su sadržane geste jednoručne abecede, dvoručne abecede, brojevi te još neki dodatni znakovi/geste.

https://lh5.googleusercontent.com/WnR9YT-hMtFtBSaM5ty3zG1TNKZ1bT-orbQyiuIH8QqcHOpjVPehXZ07r1STsionbkKYXex7vtEC1adJn3mpqOpkXdwATEum3Mk20A-ffGPj85tVi9s7Nm5_idSjZpdvVAbsfxcThttps://lh6.googleusercontent.com/gsYpMTkBfR9_2cTDF1MpdB5-2CSTUnyP5w40eZizCVJas2TNGSAWPQeTing42iS8IkmD2fCzC0F246bWu8bhoxZKF4Q7BzRD4UdBZlpnkfssICn62bcvfuLaEIAuLkQyRa6zUad3https://lh4.googleusercontent.com/qPJD8Sh2QJllDfmB-2hC0QL63le0kAxVPAzWB1XpwLR8jXoDexY2OMqCTE7WfxjbrTL-OGfnqDk7-beObn451QKpTOW9DbNVI-9eYAUwt4v6ygAJKmaDhGsZxPiEg3rWkdvC1GKphttps://lh5.googleusercontent.com/S103SDqLD6zgCpPr1Soo2XAN8i4VjgPFg38rXEKi0ghpeGK4KGjY1DvYukJM1OufNatRCl_VzBUv-2wZZxIxvSaJodozPPHjVKvCULQKBlxmUeUIaXbGDzRLP3fudpXnprrKV4Jyhttps://lh5.googleusercontent.com/SYE14GQmWhqt7MlOb0l7xsoAS3nsSxqXYsLXqWfYmsHbn1nI-gzAJ9NgzRC7aV_MjsRZLejww4gHAnNUPITEGud5LHwCJxdhai2dQQ9_FGfF46GD7ozM73g0mpKJLB5HY7DAREZAhttps://lh4.googleusercontent.com/SAATWD1Mv_773X2ZJpshixobi-KqRHE-bg8kJtmS75o0gX54X28PMlMAvAjc4XYhUByd_wVt-Smv08S4QFuvitQuQWZEikBqVce3RcPaQ57NT11i8NIT7naxqe9xGXbhvwCSgHPo

Slika 7: Prikaz znakova dvoručne abecede

(dodat prikaz svih gesti kao i kod jednoručne)

# 3. Implementacija sustava

## 3.1 Općenito o strojnom učenju

„Strojno učenje jest programiranje računala na način da optimiziraju neki kriterij uspješnosti temeljem podatkovnih primjera ili prethodnih iskustava.“ - Alpaydin 2009.

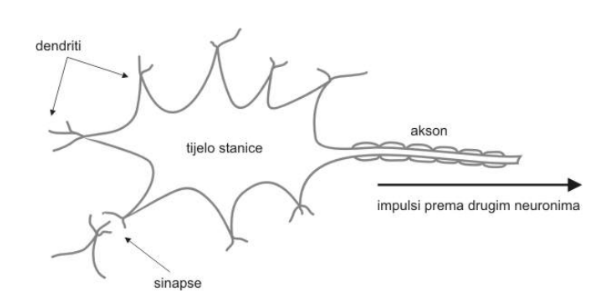
Strojno učenje se koristi iz raznih razloga, kao što su rješavanje složenih problema gdje ne postoji ljudsko znanje o procesu ili ljudi ne mogu dati objašnjenje o procesu (npr. Raspoznavanje govora) u šta spadaju i problemi koje nije moguće riješiti na klasičan algoritamski način, kod problema s ogromnim količinama podataka te kod sustava koji se dinamički mijenjaju gdje je potrebna prilagodba. Strojno učenje grana je umjetne inteligencije koja se bave oblikovanjem algoritama koji svoju učinkovitost poboljšavaju na temelju empirijskih podataka. Strojno učenje jedno je od danas najaktivnijih i najuzbudljivijih područja računarske znanosti, ponajviše zbog brojnih mogućnosti primjene koje se protežu od raspoznavanja uzoraka i dubinske analize podataka do robotike, računalnog vida, bioinformatike i računalne lingvistike.

Strojno učenje ima dva osnovna pristupa: nadzirano učenje (klasifikacija i regresija) i nenadzirano učenje (grupiranje i smanjenje dimenzionalnosti). Jedno od najboljih rješenje za strojno učenje su neuronske mreže.

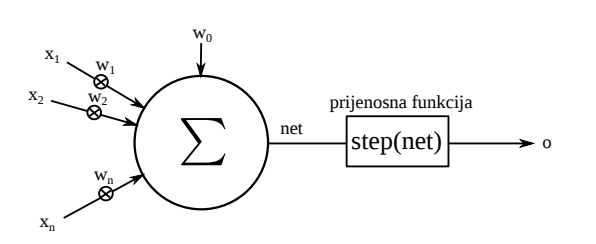
Umjetna neuronska mreža skup je međusobno povezanih jednostavnih procesnih elemenata (neurona) čija se funkcionalnost temelji na biološkom neuronu i koji služe distribuiranoj paralelnoj obradi podataka.

### 3.1.1 Neuron

Glavna jedinica neuronske mreže je neuron koji oponaša biološki neuron. Biološki neuron sastoji se od tijela (some), dendrita, aksona te završnih članaka te je u prosjeku, u ljuskom mozgu, svaki neuron povezan s 1000 do 10000 drugih neurona.

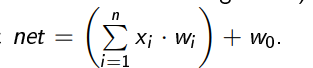


Slika 8: Biološki neuron



Slika 9: Računalni neuron

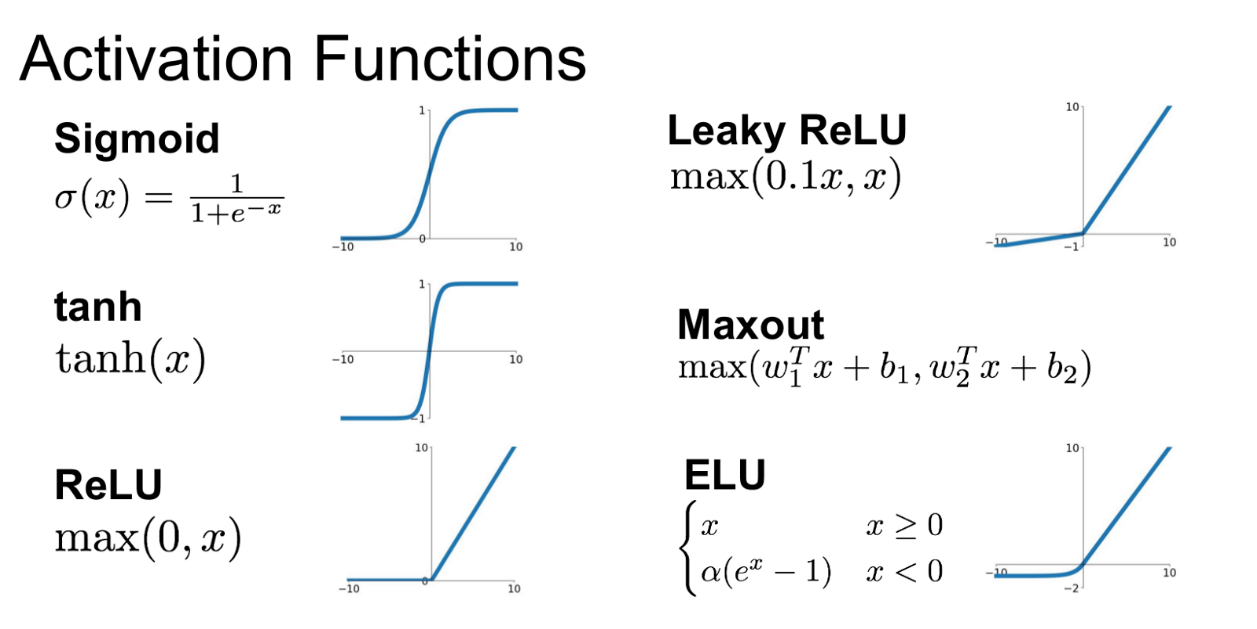
Vrijednost sa svakog ulaza xi množi se s osjetljivošću tog ulaza wi i akumulira u tijelu. Ukupnoj sumi dodaje se i pomak w0 (engl. bias) te se time definira akumulirana vrijednost net.



(napisat ovo u lijepom formatu)

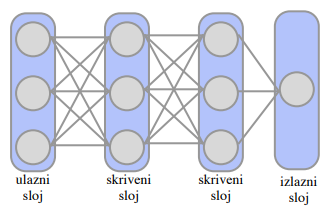
Ta se vrijednost propušta kroz prijenosnu (aktivacijsku) funkciju čime nastaje izlazna vrijednost O=step(net).

Česte prijenosne funkcije su Identitet (ADALINE-neuron), funkcija skoka (TLU-perceptron), sigumoidalna funckija (sigmoidalni neuron), zglobnica (Rectified Linear Unit, ReLU)...



Slika 10: Prijenosne (aktivacijske) funkcije

### 3.1.2 Neuronska mreža

Kako bi omogućili modeliranje složenijih odnosa u postupcima klasificiranja te regresije, koristi se više neurona. Višestruki neuroni povezani su u acikličkim grafovima, iako je jedna od najčešćih arhitektura ona višeslojnog neurona u kojima su neuroni organizirani u slojevima. Postoje ulazni i izlazni slojevi i dodatni skriveni slojevi koji povećaju veličinu i složenost mreže.

Slika 11: Model neuronske mreže

### 3.1.3 Konvolucijska neuronska mreža (*Convolutional Neural Network* - CNN)

Konvolucjske neuronske mreže mogu se opisati kao nadogradnja nad neuronskim mrežama. Konvolucijska, kao i obična, neuronska mreža sastoji se od jednog ulaznog, jednog izlaznog te jednog ili više skrivenih slojeva. Specifičnost konvolucijskih neuronskih mreža su konvolucijski slojevi i slojevi sažimanja. . Konvolucijske neuronske mreže najčešće kreću s jednim ili više konvolucijskih slojeva, zatim slijedi sloj sažimanja, pa ponovo konvolucijski sloj i tako nekoliko puta. Mreža najčešće završava s jednim ili više potpuno povezanih slojeva koji služe za klasifikaciju. Arhitektura konvolucijskih neuronskih mreža pokazala se izrazito dobra u radu sa slikama i prepoznavanju značajki s istih [Yann LeCun, Léon Bottou, Yoshua Bengio, i Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278– 2324, 1998] te se zato koristi pri prepoznavanju objekata na slici.

### 3.1.4 Povratna neuronska mreža (*Recurrent Neural Network* – RNN)

Povratna neuronska mreža može se smatrati neuronskom mrežom s pamćenjem (memorijom). Ideja povratnih neuronskih mreža je koristiti sekvencu informacija. Tradicionalne neuronske mreže pretpostavljaju da su ulazi i izlazi neovisni jedni o drugima, ali kod povratnih neuronskih mreža nije tako, kod njih su međuovisni. To se ostvaruje dodavanjem skrivenog kratkoročno memorijskog sloja između slojeva neuronske mreže. Tako se u mreži ne gleda samo ulaz nego i izlaz prijašnje iteracije te mreža gleda na njihov međusoban odnos.

## 3.2 Arhitektura mreže

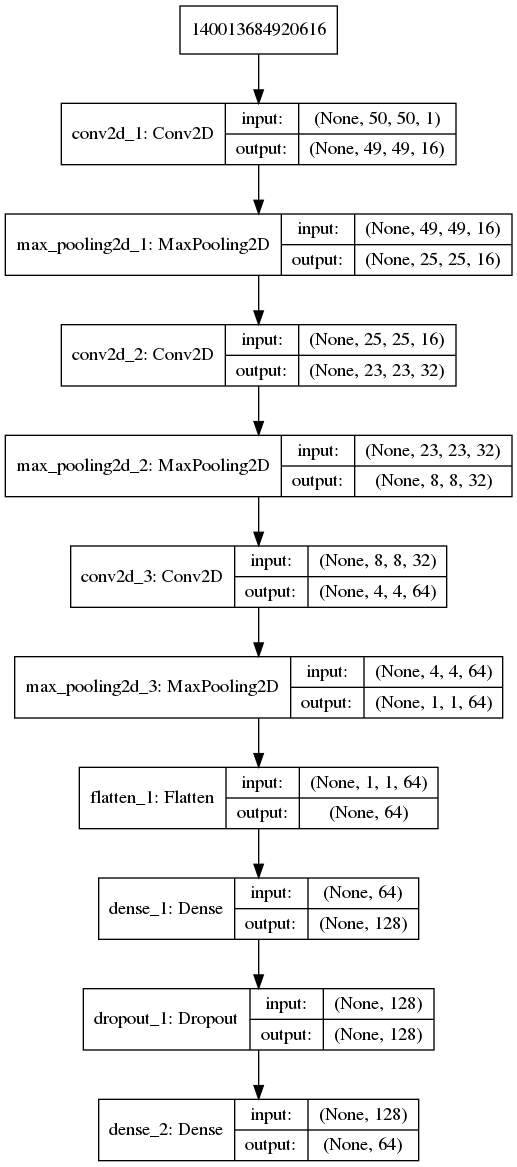
Ideja mreže je da za ulaz primi sliku, preko korisnikove kamere, a za izlaz vrati koja je to gesta. Ulazna slika je zapravo dvodimenzionalno (2D) polje veličine 50x50 piksela koja sadrži vrijednosti od 0-255 koji označavaju koje je boje taj određen piksel, gdje je 0 potpuno crna, a 255 potpuno bijela boja.

Ovom problemu pristupilo se na dva načina. Prvi pristup bio je preko konvolucijske neuronske mreže dok je drugi preko 3D konvolucijske neuronske mreže i preko povratne neuronske mreže.

### 3.2.1. Prvi pristup - Konvolucijska neuronska mreža

Mreža koja je korištena slična je (link). Mreža se sastoji od 3 dvodimenzionalna konvolucijska sloja (engl. *Convolutional 2D layer*) koji između sebe imaju slojeve sažimanja (engl*. Pooling layer*). Konvolucijski slojevi za aktivacijsku (prijenosnu) funkciju koriste zglobnicu *(engl. ReLU function*) te pokušavaju otkriti neke karakteristike na slikama.Izlaz trećeg sloja sažimanja povezan je sa slojem zaravnanja (engl. *Flatten layer*) koji dvodimentionalni izlaz pretvori u jednodimenzionalan niz. Nakon slijedi zbijen sloj (engl. *Dense layer*) koji služi kao potpuno povezan sloj (engl*. Fully connected layer*) te on svaki ulazni neuron povezuje na svaki izlazni neuron. U ovom slučaju povezuje sve ulazne neurone na 128 izlaznih neurona, također koristeći zglobnicu kao aktivacijsku funkciju, koji su povezani dalje na sloj izbacivanja (engl. *Dropout layer*) koji služi za regulaciju prenaučenosti (engl. *overfitting*) tako da zanemari nasumično odabrane neurone. Na kraju se nalazi još jedan zbijen sloj koji povezuje neurone koji su ostali nakon izbacivanja s konačnim izlaznim slojem (koji sadrži neurona koliko imamo različitih gesti) koristeći „*softmax*” aktivacijsku funkciju koja pretvara izlaze u razdiobu vjerojatnosti.

Ova arhitektura mreže bila je dobra za statičke geste, no nije mogla raspoznavati dinamičke geste.



Slika 12: Arhitektura mreže

### 3.2.2. Drugi pristup – 3D konvolucijska neuronska mreža i povratna neuronska mreža(RNN)

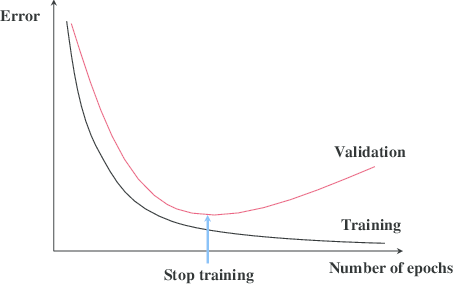
Dinamičku gestu možemo gledati kao skup od više statičkih gesti. Za taj pristup prvo je uzeta trodimenzionalna (3D) konvolucijska mreža, koja je za razliku od uobičajene konvolucijske mreže koja je primala jednu sliku (zapravo 2D matricu vrijednosti piksela), primala trodimenzionalnu matricu, gdje je treća dimenzija zapravo bila niz slika (2D matrica). Ako se odredi točan broj kadrova geste te se pretpostavi da će se dinamička gesta izvoditi poznatom brzinom može se uzeti slike u određenim kadrovima (primjerice prvom, srednjem i zadnjem kadru izvođenja geste) te ih spojiti u trodimenzionalni ulaz koji bi se mogao interpretirati kao vremenski slijed. (ovo još trebam nadopisati kada dovršim implementaciju)

## 3.3. Treniranje mreže

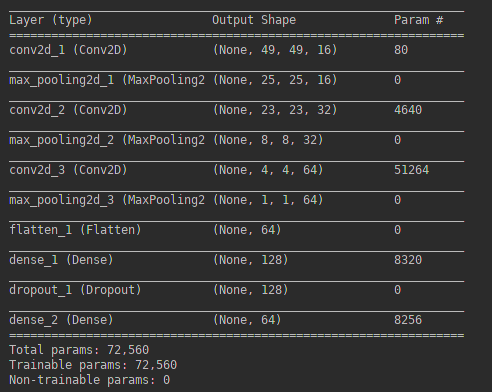
Sveukupni skup podataka sadrži 64 klasifikacijske kategorije, gdje svaka kategorija sadrži 2400 slika, što rezultira 153 600 slika sveukupno. Za treniranje je uzeto 5/6 skupa podataka (128 000 slika), za validaciju 1/12 skupa (12 800 slika) i za testiranje 1/12 skupa (12 800).

Za hiperparametri mreže uzeti su broj epoha koji iznosi 50 i veličina serije (engl. *Batch size*) koja iznosi 500. Broj epoha je broj iteracija kroz cijeli skup podataka dok veličina serije predstavlja broj koliko primjeraka algoritam prođe prije nego ažurira vrijednosti težina neurona.

Kako bi se izbjegla prenaučenost uvedena je metoda ranog zaustavljanja (engl. *Early stopping*). Ona zaustavlja treniranje čim se gubitak (engl. *Loss*), odnosno greška, na validacijskom skupu krene povećavati.



Slika 13: Prikaz metode ranog zaustavljanja

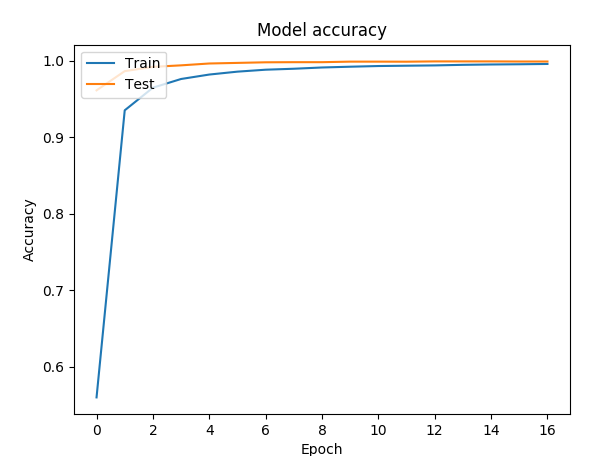


Slika 14: Parametri o oblik slojeva

(dati uvod u sliku)

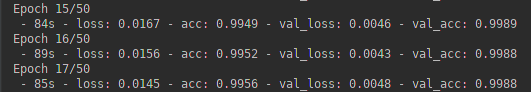
# Rezultati

Budući da sustav raspolaže s jednostavnim crno-bijelim slikama veličine 50x50 piksela već nakon prve epohe treniranja dosegne preciznost od preko 90% (Slika 15).

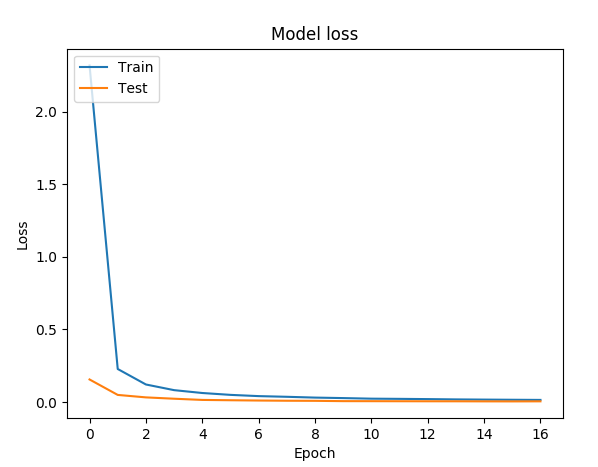


Slika 15: Prikaz preciznosti kroz broj epoha

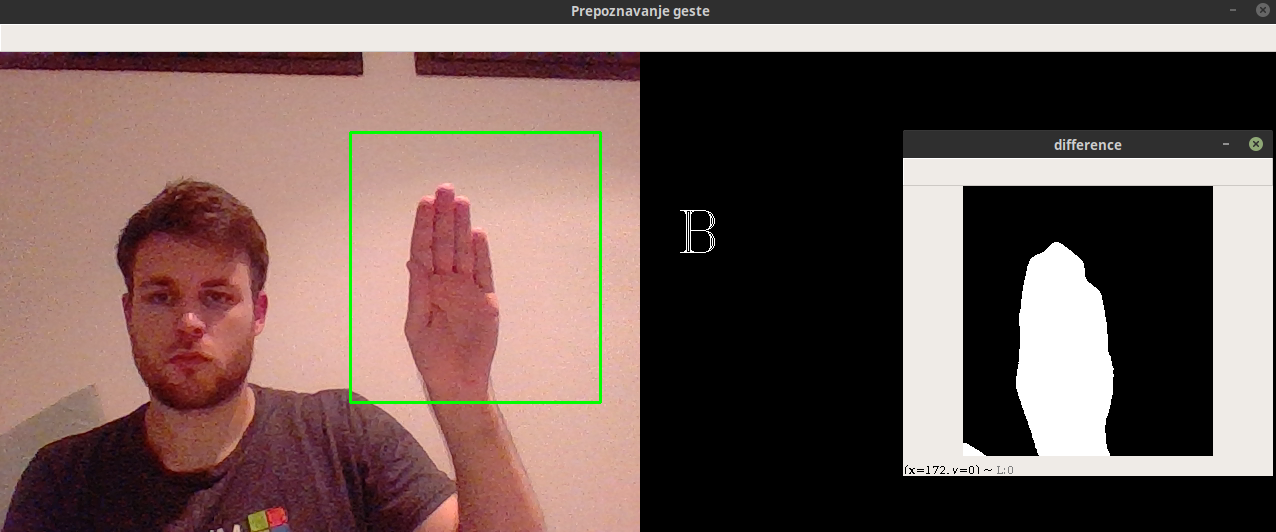
Metoda ranog zaustavljanja prekida treniranje mreže nakon 16 epoha zbog rasta gubitka te preciznost mreže tada iznosi 99.88%.



Slika 16: Ispis podataka zadnje tri epohe treniranja

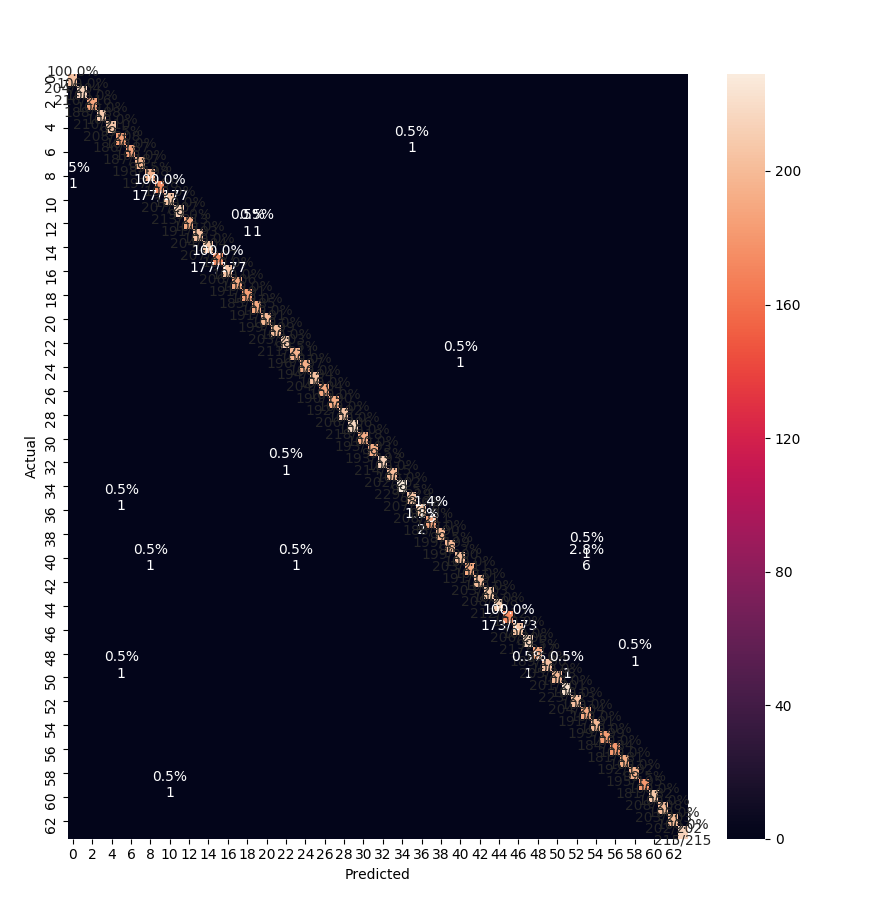


Slika 17: Prikaz gubitka kroz broj epoha



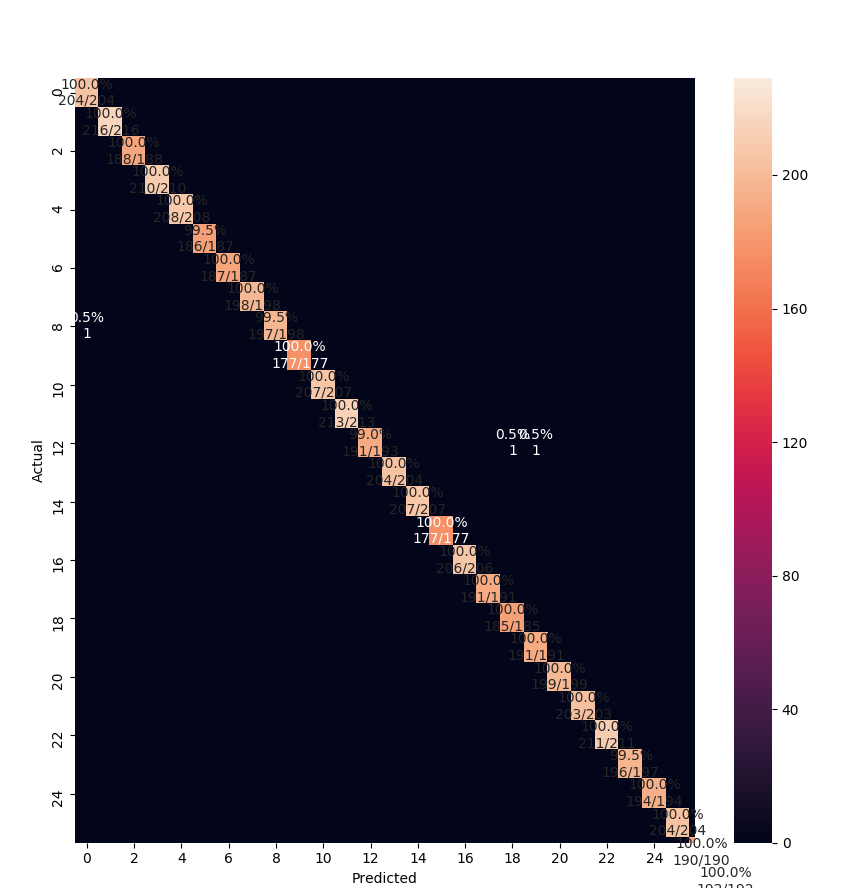
Slika 18: Primjer pravilnog rada

Dobar prikaz ispravnih i pogrešnih izlaza pruža matrica konfuzije koja može prikazati koji podaci su međusobno slični, odnosno za koju kategoriju ulaza je mreža dala krivu kategoriju izlaza.



Slika 19: Prikaz cijele matrice konfuzije

Kako je vjerojatnost modela mreže veoma visoka, nema ni puno odstupanja.



Slika 20: Prikaz dijela konfuzijske matrice

## 4.1. Problemi

Iako je veoma visoka preciznost mora se uzeti u obzir da je mreža trenirana u „savršenim” uvjetima gdje su slične geste vidljivo različite, kod prepoznavanja u stvarnom vremenu i stvarnim uvjetima, korisnik može nesavršeno izvesti gestu što ipak može prouzročiti krivi rezultat. Osim toga, znakovna abeceda sadrži neke veoma slične znakove i geste. Primjer toga su slova „a”, „s” i „e”, koja se mogu vidjeti na slikama.

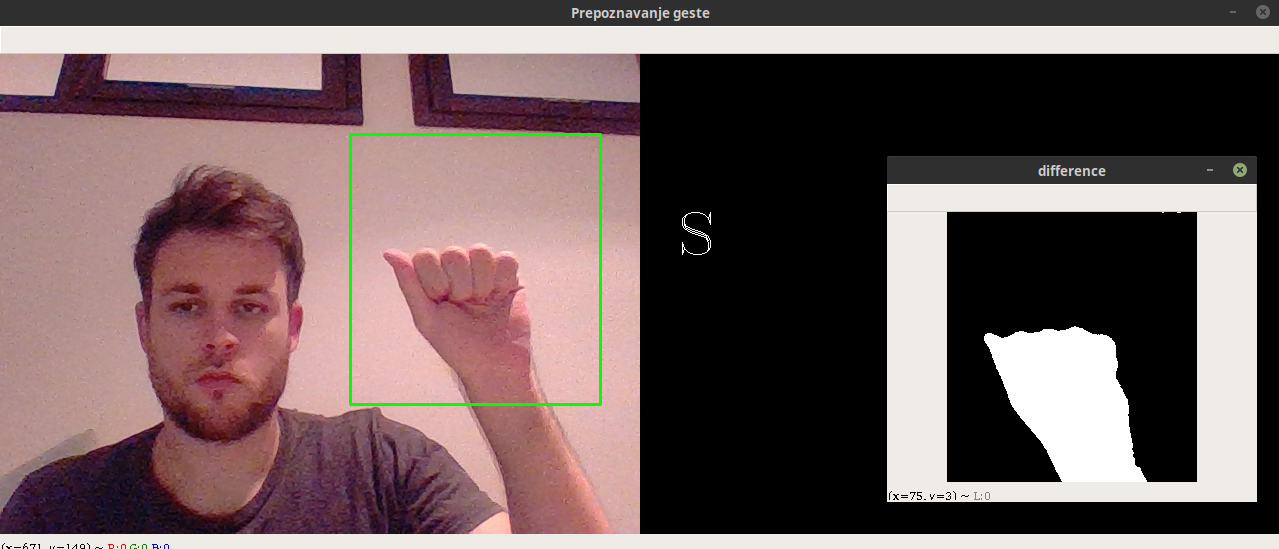


Slika 21: Prikaz slova "a", "e" i "s"

Slova su sva u obliku šake te se razlikuju samo u poziciji palca što je vidljivo na slikama, no kako mreža koristi crno-bijele slike koje nisu toliko detaljnje može doći do zabune.



Slika 22: Prikaz slova "a", "e" i "s", ali crno-bijelo



Slika 23: Primjer krive interpretacije

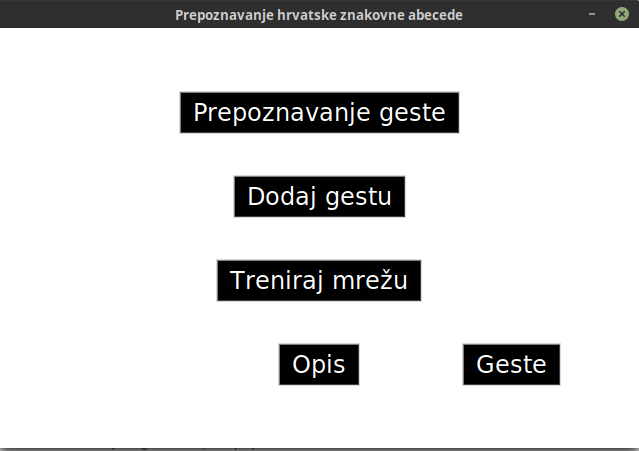
# Instalacija i upute za korištenje

## Potrebne instalacije (*Requirements*)

Za izvođenje programa potrebno je imati Python 3.x.

Za instaliravanje svih potrebnih alata potrebno je pozicionirati se terminalom u direktorij gdje se nalazi datoteka requirements.txt te pokrenuti naredbu (uz instaliran pip3): „pip install requirements.txt”

## Pokretanje

Postoji jednostavno sučelje izrađeno u „*Tkinteru*“ koje olakšavanje služenje. Ono sadržava opcije opisa, opcija prikaza gesti, opciju dodavanja geste, opciju treniranja modela nakon dodanih gesti te opciju pokretanja glavnog programa koji prepoznaje geste.

Slika 24: Prikaz početnog sučelja

Bitno je naglasiti kako prilikom paljenja programa bi bilo idealno kada bi površina ograničena kvadratom na kameri bila prazna, odnosno ništa se tamo ne bi trebalo nalaziti. Ako prozor „*difference*“ nije potpuno crn kod paljenja programa korisnik pritiskom na tipku “r” može ponovo pokrenuti program. Isto tako korisno je naglasit da korisnik iz programa izlazi pritiskom na tipku “Esc”.

### Dodavanje geste

Pokretanjem programa za dodavanje geste pali se kamera te se na ekranu može vidjeti prikaz kamere uz jedan zeleni kvadrat, na njegovo mjesto treba postaviti dlan s obzirom na gestu koju želite spremiti te kliknuti tipka “c” koja pokreće dohvaćanje (engl *capture*) nakon kratkog vremena. Dohvaćanje se može svakog trenutka pauzirati i odpauzirati ponovnim klikom na tipku “c”. Na dodatnom prozoru koji se otvorio i koji se zove “*difference*” može se vidjeti kakva se slika sprema, te ako to nije zadovoljavajuća gesta, na klik “r” se može ponovo pokrenut. Nakon otprilike minute, sustav uslika 1200 slika, koje zrcali te spremi. Nakon dodavanja geste potrebno je ponovo trenirati mrežu, kako bi se vidjeli rezultati dodavanja, što može potrajat i do 30 minuta.



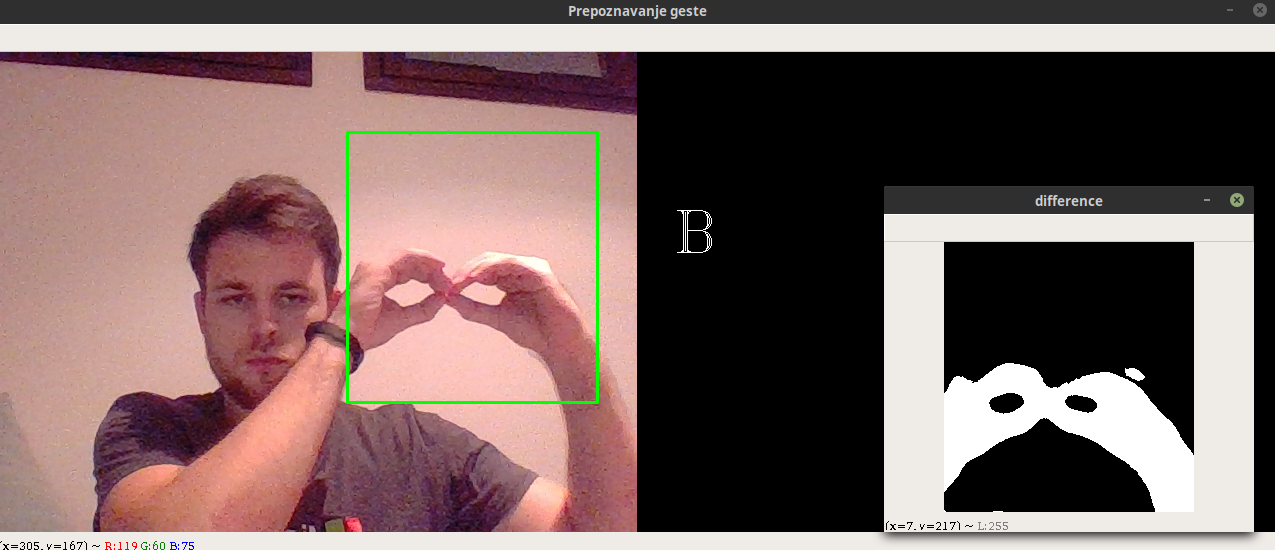
Slika 25: Primjer dodavanja geste "c"

### Treniranje mreže

Nakon dodavanja geste potrebno je ispočetka trenirati mrežu. Pokretanjem programa za treniranje mreže nasumično se rasporede slike i podijele u skupove za treniranje, učenje i testiranje. Nakon čega slijedi „treniranje“ neuronske mreže koje može trajati i do 30 minuta te će veoma opteretiti računalo. Nakon završetka treniranja, sprema se model mreže te je sve spremno za prepoznavanje gesti.

### Prepoznavanje geste

Pokretanjem programa prepoznavanja geste pali se korisnikova kamera gdje možemo, uz uobičajeni prikaz kamere i crno-bijeli prikaz ruke “*difference*” koji predstavlja što računalo vidi, vidjeti i nadodan dio na kameru koji predstavlja „ploču“ na kojoj se mogu iščitati geste.



Slika 26: Primjer prepoznavanja geste

Pritiskom na tipku “t” ulazimo u “textMode” koji omogućava spajanje slova, odnosno ispisivanje riječi. Za nadodavanje slova treba na ulazu biti jednaka gesta određen broj kadrova (otprilike 2 sekunde). Ponovnim pritiskom na tipku “t“ vraćamo se u standardni način.

# 

Slika 27: Primjer tvorenja riječi pomoću TextMode-a

# Zaključak

# Literatura