# Prepoznavanje znakovnog jezika korišćenjem dubokih neuronskih mreža i kompjuterske vizije

*Nevena Bojović (47/2018), E-mail: 47-2018@pmf.kg.ac.rs*

*Milica Vučić (52/2018), E-mail: 52-2018@pmf.kg.ac.rs*

*Stefan Petrović (86/2017), E-mail: 86-2017@pmf.kg.ac.rs*

**Apstrakt**: Oštećenje govora je invaliditet koji utiče na sposobnost pojedinca da komunicira. Ljudi koji imaju ovakav problem koriste druge načine komunikacije kao što je znakovni jezik. To što je znakovni jezik sve prisutniji predstavlja izazov za one koji ne govore znakovnim jezikom da komuniciraju sa govornicima ili poznavaocima istog. Problem u prepoznavanju gestova može se rešiti tehnikama zasnovanim na dubokom učenju i kompjuterskoj viziji. Fokus ovog rada je kreiranje aplikacije, zasnovane na korišćenju modela konvolucione neuronske mreže, koja prepoznaje gest znakovnog jezika i na taj način pomaže u komunikaciji.

## 1. Uvod

Znakovni jezik predstavlja oblik komunikacije koji koriste osobe sa oštećenim sluhom i govorom. Oni koriste gestove znakovnog jezika kao sredstvo neverbalne komunikacije. Gestovima znakovnog jezika izražavaju se emocije i misli. Ljudi koji nemaju navedeni oblik invaliditeta teško razumeju znakovni jezik, te su im potrebni tumači kao medijatori u komunikaciji. Postoje usluge prevođenja online, ali je za to neophodan pristup internetu i nekom od uređaja za komunikaciju. Jednostavniji način prevođenja gestova znakovnog jezika, bez navedenih ograničenja, mogli bismo obezbediti korišćenjem tehnika veštačke inteligencije i kompjuterske vizije.

Za implementaciju prethodno navedenog načina prevođenja, konstruisali smo mnoštvo modela konvolucionih neuronskih mreža od kojih je šest trenirano na originalnom skupu podataka sa grayscale slikama [3], sedam trenirano na RGB slikama koje su dobijene transformacijom originalnih grayscale slika, pet nad RGB slikama koje su augmentovane (najbolji od ovih modela je Fine Tune-ovan da bismo utvrdili da li možemo doći do još boljih rezultata), tri modela na koje je primenjen Transfer Learning (korišćenjem VGG16, InceptionV3 i MobileNet) i tri modela sa Transfer Learningom koji su Fine Tune-ovani (takođe VGG16, InceptionV3 i MobileNet).

Kao najbolji model izdvojio se model koji je treniran nad RGB slikama koje su augmentovane. U nastavku izveštaja će biti više reči o njemu.

CNN modele koje smo konstruisali za rešavanje navedenog problema, napravili smo koristeći Tensoflow i Open-CV biblioteke.

Ostatak izveštaja istraživanja organizovan je na sledeći način: u Odeljku 2 dat je pregled literature, u Odeljku 3 opisani su skupovi podataka (trening i test) i njihove karakteristike, Odeljak 4 daje uvid u arhitekturu mreže koja je dala najbolje rezultate, dok u Odeljku 5 predstavljamo rezultate treniranja i testiranja modela. U Odeljku 6 prikazana je struktura aplikacije. U Odeljku 7 smo izložili probleme sa kojima smo se susreli prilikom predikcije i dali predloge za poboljšanja modela i aplikacije.

## 2. Pregled literature

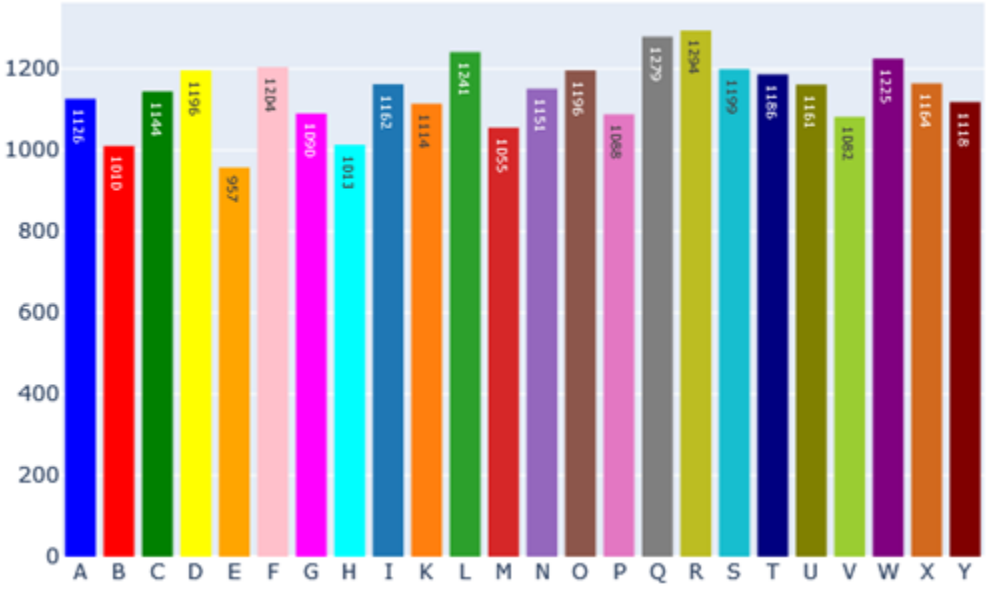
[8] Ovaj rad se fokusira na prepoznavanju gestova znakovnog jezika engleske abecede koja sadrži 26 gestova ruku (A-Z) i 10 cifara (0-9). Rešenje se bazira na korišćenju duboke neuronske mreže. Modeli neuronske mreže su konstruisani nad postojećim arhitekturama (LeNet-5 i MobileNetV2), ali i nad arhitekturom koju su osmislili autori rada. Napravljena je web aplikacija koristeći Django Rest Framework da bi se rezultati testirali u real time-u i dobile predikcije.
[9] Sistem za prepoznavanje je napravljen korišćenjem algoritama mašinskog učenja u MATLAB-u. Autori su radili sa gestovima koji zahtevaju jednu ili obe ruke. Korišćena su dva algoritma, K najbližih suseda i Back Propagation algoritam. Postignuta je tačnost od 93-96%. Iako je veoma precizan, ovaj sistem nije implementiran da radi u real time-u.
[10] U ovom radu, region u kom se nalazi ruka se segmentuje koristeći model baziran na ljudskoj boji kože u YCbCr modelu boja. Zatim se primenjuje threshold u cilju razdvajanja prikazanog gesta od pozadine I na kraju template matching tehnika je razvijenja korišćenjem PCA za prepoznavanje gesta.
[11] Autori predlažu sistem koji je implementiran da prepozna pokrete ruke u real time-u. On može da prepozna 35 pokreta rukama, kojima su predstavljena slova američkog i indijskog znakovnog jezika. Slike su iz RGB formata konvertovane u grayscale format da bi se smanjio broj lažno pozitivnih. Autori koriste SIFT (Scale Invariant Feature Transform) metodu. Rešenje je implementirano u MATLAB-u. Za aplikaciju postoji i GUI model.
[12] Osnovna funkcionalnost aplikacije koja se u ovom radu navodi jeste konverzija gestova u tekst. Koraci procesiranja su: preprocesiranje podataka, izdvajanje gesta sa slike, utvrđivanje kom simbolu znakovnog jezika pripada gest i dobijanje glasovne reprezentacije od teksta dobijenog u prethodnom koraku. Za izdvajanje gesta su korišćene dve metode: Skin color segmentation i Region Growing, dok su za prepoznavanje gesta korišćene tehnike SIFT i Correlation matching.

## 3. Skup podataka

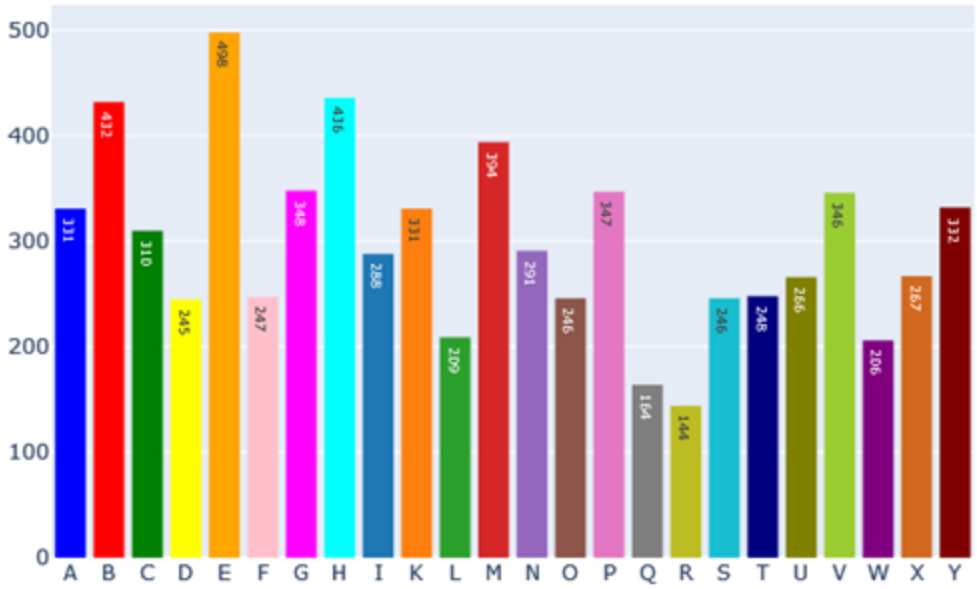
Skup podataka [3] sadrži 2 `csv` fajla, `sign_mnist_training.csv` i `sign_mnist_test.csv`. U `csv` fajlovima su slike predstavljene preko vrednosti piksela. Pošto su originalne slike dimenzija 28 X 28, postoji ukupno 784 featura za slike i jedan feature koji predstavlja labelu. Postoje 24 labele (ne računaju se labele za znakove “J” I “Z” jer se ovi znakovi predstavljaju pokretom, a ne samo položajem ruku). Inače, labele su numerisane na sledeći način: 0 - A, 1 - B, 2 – C …. Y – 23. Vizuelni prikaz skupa podataka dat je na *Slici 1*.


*Slika 1. Grayscale slike – prikaz gestova*

U fajlu sign_mnist_training.csv se nalazi trening skup, koji sadrži 27455 slika, dok test skup (sign_mnist_test.csv) sadrži 7172 slika. Zastupljenost svakog slova u trening i test skupu data je na *Slici 2 i Slici 3* :




*Slika 2. Distribucija na trening skupu*




*Slika 3. Distribucija na test skupu*

Primećujemo da je različita distribucija podataka na ovim skupovima. Potencijalni problem koji može da nastane je velika razlika u broju instanci određenih klasa. Na primer, loša predikcija na nekom broju instanci klase R će više uticati na rezultate modela nego na istom broju instanci klase E.

Koristimo `LabelBinarizer()` za enkodiranje labele (kao izlaz dobijamo vektor). Radimo sa originalnim grayscale slikama, ali ćemo ih takođe transformisati u RGB slike, tako što dodajemo još 2 kanala, sa istim vrednostima piksela koje se nalaze u prvom kanalu, da bismo mogli da proverimo da li će rezultati treniranja biti bolji na RGB slikama.

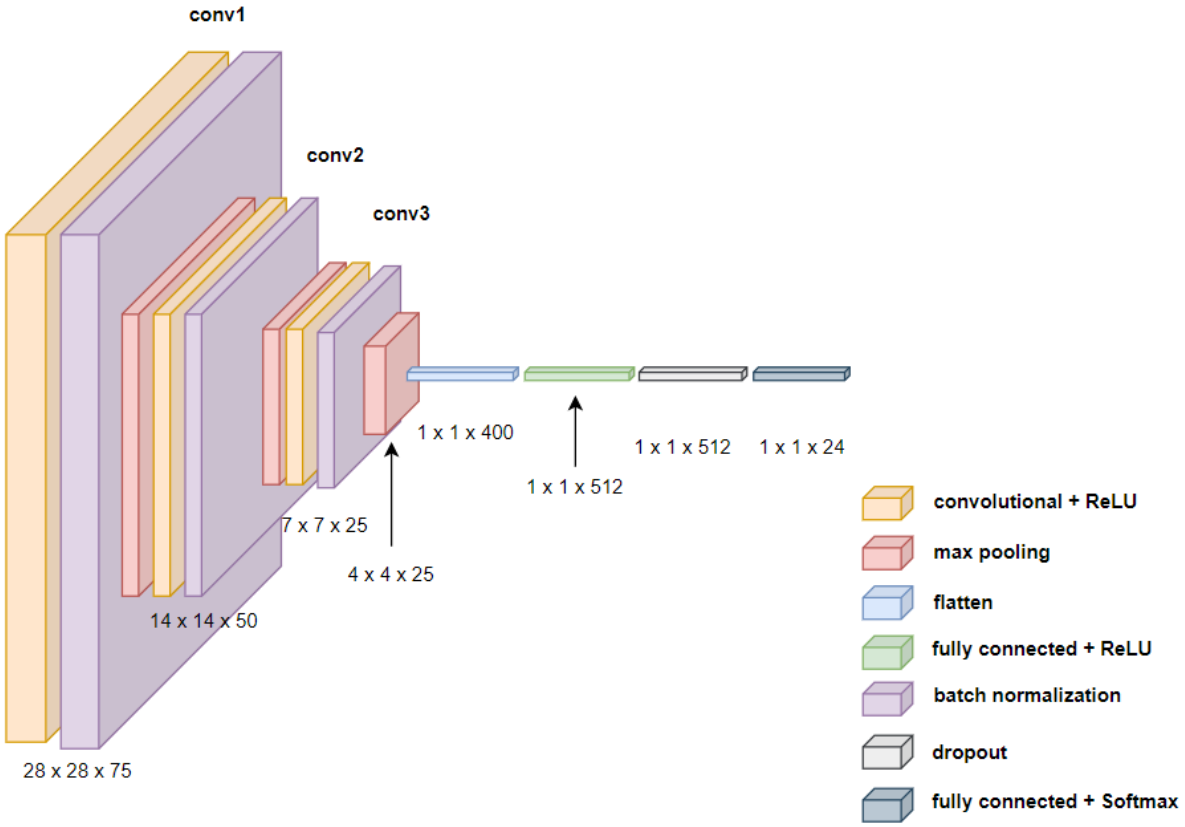
Primećujemo da slike u RGB formatu izgledaju skoro isto kao slike u grayscale formatu (*Slika 1 i Slika 4*). Razlog je to što se prilikom dodavanja neophodnih kanala „prepisuje“ ista vrednost piksela koji se nalaze na prvom kanalu.


*Slika 4. RGB slike – prikaz gestova*

## 4. Arhitektura istreniranog CNN modela

CNN model se sastoji od 14 slojeva:

- Ulazni sloj
- Konvolucioni sloj (75 filtera dimenzije 3 X 3)
- Batch Normalization
- Max Pooling (dimenzije filtera 2 X 2, stride = 2)
- Konvolucioni sloj (50 filtera dimenzije 3 X 3)
- Batch Normalization
- Max Pooling (dimenzije filtera 2 X 2, stride = 2)
- Konvolucioni sloj (25 filtera dimenzije 3 X 3)
- Batch Normalization
- Max Pooling (dimenzije filtera 2 X 2, stride = 2)
- Flatten
- Potpuno povezani sloj
- Dropout (0.3)
- Izlazni sloj




*Slika 5. Arhitektura CNN modela*

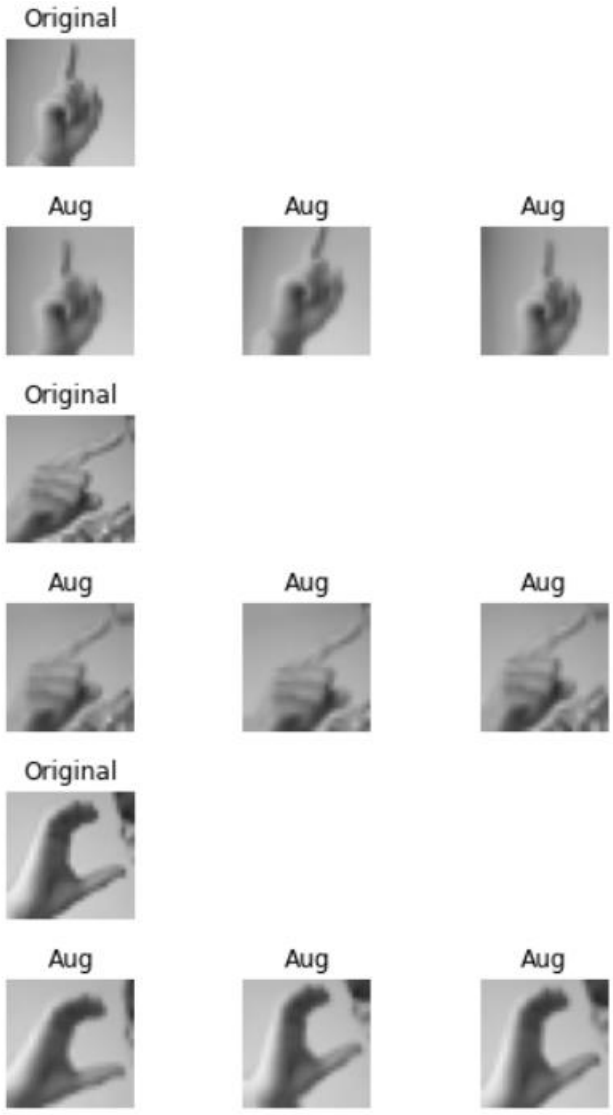
U modelu koji smo trenirali su korišćeni konvolucioni kerneli dimenzija 3 X 3. Svaki Max Pooling sloj je dimenzija 2 X 2 što utiče na znatno smanjenje dimenzionalosti tenzora. Konvolucije i potpuno povezani sloj koji se nalazi posle Flatten sloja imaju ReLu (Rectified Linear Unit), dok poslednji potpuno povezani sloj (klasifikacioni sloj) ima Softmax aktivaciju. Posle svake konvolucije rađena je Batch Normalization kako bi se poboljšalo treniranje podataka. Dropout sloj sa datom verovatnoćom od 30% je iskorišćen u ovom modelu kako bi se izbegao overfitting modela jer se „isključuje“ 30% neurona iz ovih potpuno povezanih slojeva. Izlaz iz mreže se dobija primenom SoftMax aktivacije.

Konačan model je model sa najmanjim vrednošću funkcije gubitka na validacionim skupu podataka.

Ispod su navedeni parametri augmentacije koji su dali najbolje rezultate:

```
rescale = 1./255,
rotation_range=10,
zoom_range = 0.05,
width_shift_range=0.1,
height_shift_range=0.1,
validation_split=0.2,
channel_shift_range = 0.2
```

Cilj augmentacije je bio da se dobije robusniji model.



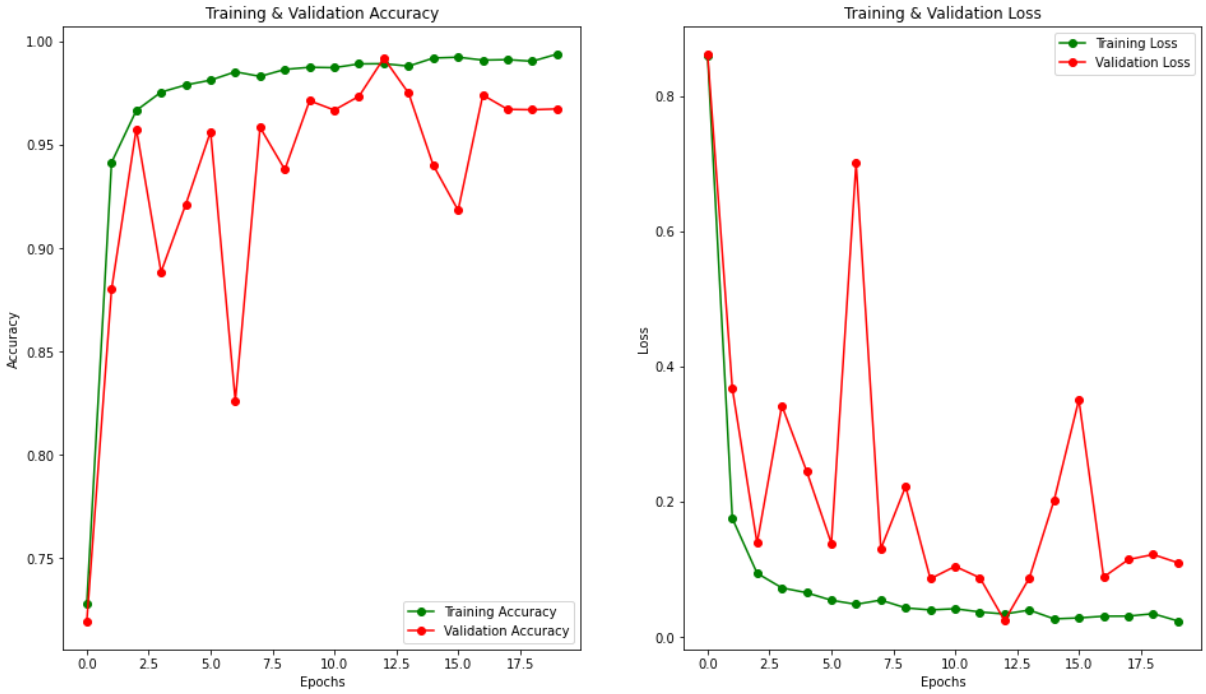

*Slika 6. Uporedni prikaz originalnih i augmentovanih slika*

Ovako konstruisan CNN model daje visok nivo od 99,97% tačnosti na trening skupu. Osim tačnosti, ispod su prikazane ostale metrike evaluacije na trening skupu:

```
loss: 0.0013
f1_score: 0.9997
precision: 0.9997
recall: 0.9996
```

## 5. Rezultati modela

Model je obučen tako da minimizuje funkciju gubitka (engl. *loss*) `categorical_crossentropy`, a optimizator koji je korišćen je *ADAM* [4]. Model je obučen na 20 epoha gde je batch size 32. Korišćena je podrazumevana stopa učenja (engl. *learning rate*) 0.001.
Na *Slici 7* prikazani su grafici tačnosti modela i funkcije gubitka na trening i validacionom skupu prilikom treniranja.



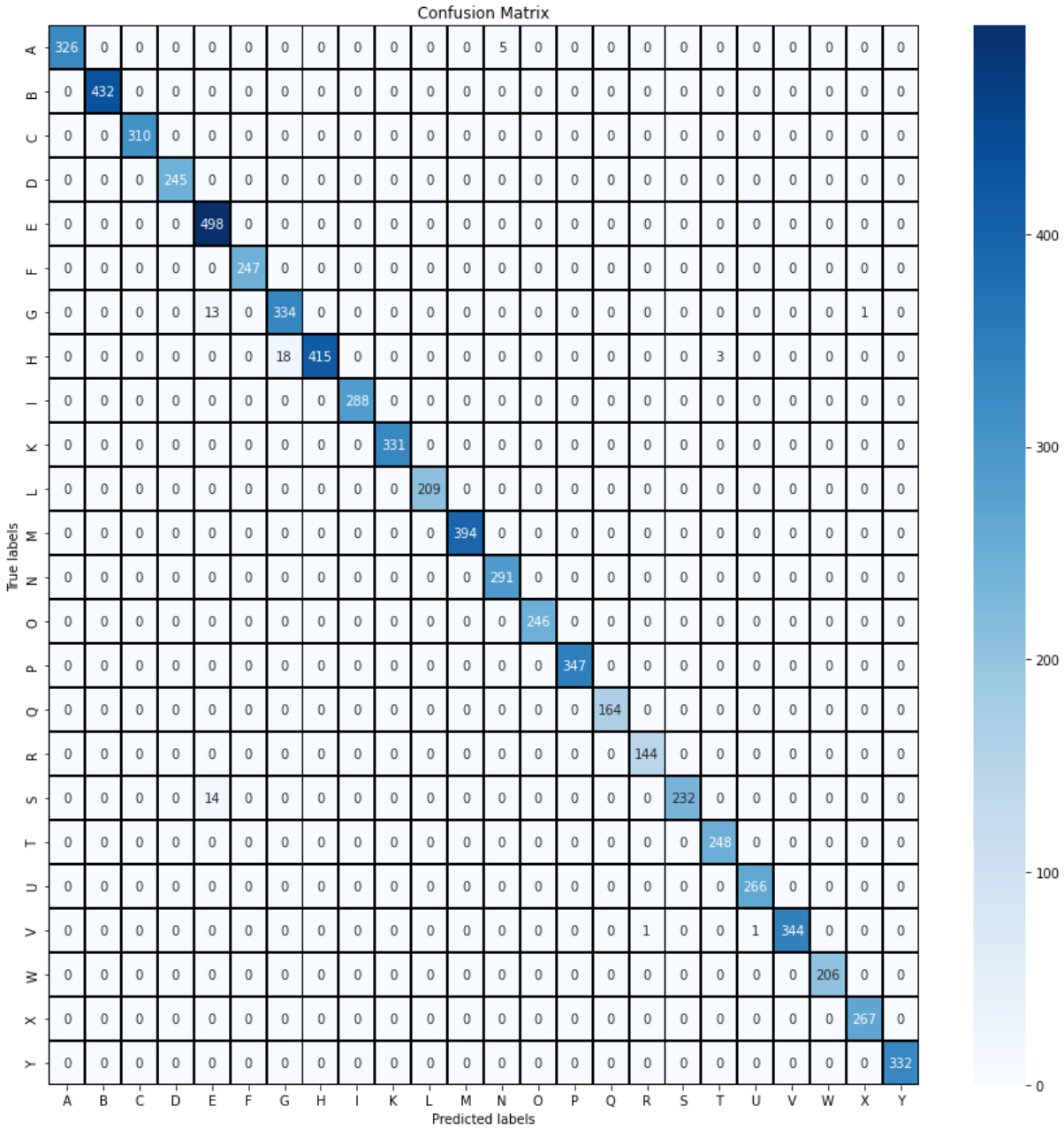

*Slika 7. Grafici promene tačnosti modela i funkcije gubitka na trening i validacionom skupu*

Na test skupu su dobijeni sledeći rezultati evaluacije:

```
loss: 0.0192
accuracy: 0.9922
f1_score: 0.9934
precision: 0.9923
recall: 0.9922
```

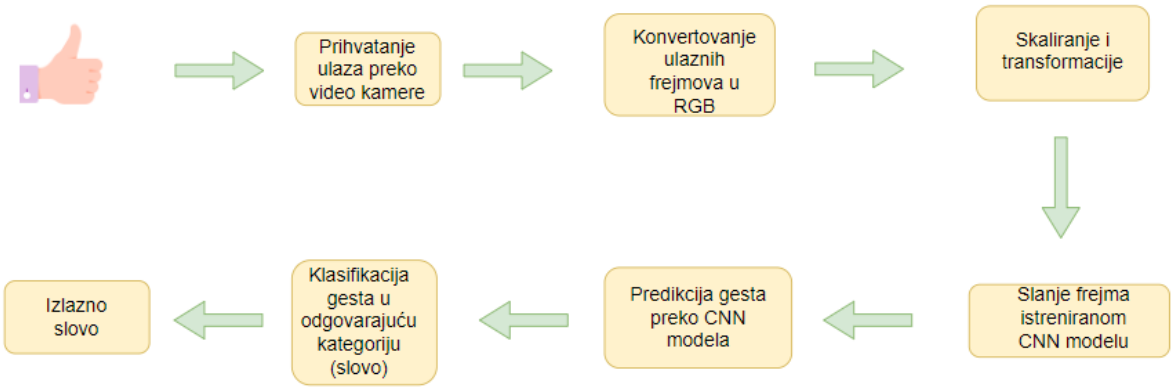
Primećujemo da je veoma visok nivo tačnosti na test skupu, čak 99,22%.

Na *Slici 8* prikazana je konfuziona matrica sa klasifikacijom svih instanci test skupa u definisane kategorije. Model radi veoma dobru klasifikaciju, mali broj slika se pogrešno klasifikuje.




*Slika 8. Konfuziona matrica klasifikacije na test skupu*

# 6. Struktura aplikacije




*Slika 9. Arhitektura aplikacije*

Kao što je prikazano na *Slici 9*, slika sa početka se deli iz video ulaza sa kamere. Frejmovi su izvučeni sa slike i nalaze se u regionu od interesa (Threshold Square Box) kako bi se izbegli konflikti u pozadini. Koristi se istrenirani CNN model sa 14 slojeva, koji je prethodno opisan. Slika gesta izdvojena iz video frejma se nalazi u RGB formatu. Pošto koristimo istrenirani CNN model nad RGB slikama, ulazna slika ostaje u istom formatu za dalju manipulaciju u aplikaciji. Dalje, slika se skalira u odnosu na veličinu slika sa kojima je model bio treniran. Unapred obučeni CNN model prihvata sliku nakon skaliranja i transformacije. Naredni korak je predviđanje gesta pomoću CNN modela. Klasifikovani gest se prikazuje kao slovo.

Na narednim slikama dat je prikaz gesta u aplikaciji (slika levo) i verovatnoća kojoj klasi taj gest pripada (date su tri klase kod kojih je najveća verovatnoća predikcije na slici desno):

```
Predicted Character 1:  C
Confidence 1:  100.0
Predicted Character 3:  U
Confidence 3:  6.257697156828415e-29
Predicted Character 2:  nothing
Confidence 2:  2.2600465336003461e-10
```

```
Predicted Character 2:  C
Confidence 2:  0.5896927323192358
Predicted Character 1:  F
Confidence 1:  99.4093120098114
Predicted Character 3:  H
Confidence 3:  0.000797191296442179
```

```
Predicted Character 1:  Y
Confidence 1:  99.37869310379028
Predicted Character 3:  T1
Confidence 3:  0.0012431904906406999
Predicted Character 2:  nothing
Confidence 2:  0.6200044881552458
```

*Slika 10. Prikaz gesta i predikcija (za slova C, F, Y)*

## 7. Diskusija

Osnovni problem sa kojim smo se susreli prilikom testiranja aplikacije jeste to što na kvalitet predikcije utiče količina osvetljenja u prostoriji i boja pozadine. Poželjno je da pozadina bude svetla, kao i da osvetljenja bude umereno, tako da ne bude previše tamno niti previše osvetljeno. Takođe, osoba koja pokazuje gest mora se udaljiti od kamere, da bi se frejm koji aplikacija prihvata dalje preprocesirao na odgovarajući način.

Ograničenje koje zadaje skup koji smo koristili u istraživanju, je to što slovo T nije prikazano na način koji se najčešće upotrebljava. Da bismo prevazišli ovu situaciju, primenili smo Fine Tuning nad modelom koji nam je dao najbolje rezultate sa proširenim skupom podataka (dodali smo dve klase, jednu klasu za „uobičajeno“ slovo T i jednu za pozadinu).

Problem „mešanja“ gestova koji su slični je primetan. Gestovi kojima se prikazuju slova M, N, S, A i E su vizuelno slični, što se može videti na Slici 1. Za znakove A, M, N, S, E položaj šake je sličan (pesnica), a položaj palca je različit (A - položaj palca je slobodan, E - palac je savijen ispod ostalih prstiju šake, M - palac se nalazi između 4. i 5. prsta, N - palac se nalazi između 3. i 4., S - palac se nalazi ispred 1. i 2. prsta šake). Veća sličnost je prisutna između slova M i N (ukoliko posmatramo manji skup slova koja su slična).

Ovaj model se može primeniti i na druge znakovne jezike kao što je indijski znakovni jezik, tako što će se uraditi Fine Tuning nad proširenim skupom slika. Model se može dodatno obučiti sa skupom podataka tako da automatski izdvoji gest iz frejma automatskim oduzimanjem pozadine. Takođe se može podesiti i proširiti model da bi se identifikovale frekventne reči i izrazi. Pored toga, obuka modela neuronske mreže za simbole nekog drugog znakovnog jezika zahteva dve ruke (britanski, australijski, novozelandski znakovni jezik). Integracija optimizovanog modela u postojeće AI sisteme kao što je Amazon Alexa.

# 8. Zaključak

Ovaj rad je zasnovan na CNN-u za prepoznavanje i klasifikaciju znakovnog jezika koristeći kompjuterski vid. Za razliku od drugih pristupa, ovaj pristup daje veoma visoku tačnost i znatno mali broj lažno pozitivnih (engl. *False Positive*). Ostala moguća proširenja ovog rada predstavljena su u Odeljku 7 (Diskusija).

# 9. Dodatak

Na Slici 11 prikazani su rezultati svih modela koje smo koristili prilikom istraživanja. Za svaki model istaknut je loss i accuracy za trening i test skup.

| Performanse modela | | | | | | | |
|---|---|---|---|---|---|---|---|
| Grayscale 1 | Training accuracy: 1.0000 | RGB 1 | Training accuracy: 1.0000 | Augmentacija 1 | Training accuracy: 0.9933 | Transfer Learning VGG16 | Training accuracy: 0.9904 |
| | Training loss: 1.4374e-05 | | Training loss: 2.1443e-06 | | Training loss: 0.0279 | | Training loss: 0.0306 |
| | Test accuracy: 0.8707 | | Test accuracy: 0.8922 | | Test accuracy: 0.9859 | | Test accuracy: 0.9387 |
| | Test loss: 0.8099 | | Test loss: 0.8221 | | Test loss: 0.0385 | | Test loss: 0.4083 |
| Grayscale 2 | Training accuracy: 1.0000 | RGB 2 | Training accuracy: 1.0000 | Augmentacija 2 | Training accuracy: 0.9948 | Transfer Learning Inception V3 | Training accuracy: 0.9937 |
| | Training loss: 3.1127e-06 | | Training loss: 3.5949e-07 | | Training loss: 0.0167 | | Training loss: 0.0236 |
| | Test accuracy: 0.9331 | | Test accuracy: 0.9246 | | Test accuracy: 0.9819 | | Test accuracy: 0.9600 |
| | Test loss: 0.3931 | | Test loss: 0.4282 | | Test loss: 0.0607 | | Test loss: 0.1207 |
| Grayscale 3 | Training accuracy: 1.0000 | RGB 3 | Training accuracy: 1.0000 | Augmentacija 3 | Training accuracy: 0.9997 | Transfer Learning MobileNet | Training accuracy: 0.9985 |
| | Training loss: 4.8709e-07 | | Training loss: 1.2234e-06 | | Training loss: 0.0013 | | Training loss: 0.0039 |
| | Test accuracy: 0.9416 | | Test accuracy: 0.9359 | | Test accuracy: 0.9922 | | Test accuracy: 0.9900 |
| | Test loss: 0.3264 | | Test loss: 0.3737 | | Test loss: 0.0192 | | Test loss: 0.0564 |
| Grayscale 4 | Training accuracy: 1.0000 | RGB 4 | Training accuracy: 1.0000 | | | Transfer Learning+Fine Tune VGG16 | Training accuracy: 0.0471 |
| | Training loss: 2.0206e-05 | | Training loss: 1.4397e-06 | | | | Training loss: 3.1755 |
| | Test accuracy: 0.9334 | | Test accuracy: 0.9434 | | | | Test accuracy: 0.0201 |
| | Test loss: 0.2778 | | Test loss: 0.3756 | | | | Test loss: 3.1995 |
| Grayscale 5 | Training accuracy: 1.0000 | RGB 5 | Training accuracy: 1.0000 | | | Transfer Learning+Fine Tune InceptionV3 | Training accuracy: 0.9945 |
| | Training loss: 1.3510e-07 | | Training loss: 1.9257e-07 | | | | Training loss: 0.0235 |
| | Test accuracy:0.9406 | | Test accuracy: 0.9590 | | | | Test accuracy: 0.9547 |
| | Test loss: 0.3127 | | Test loss: 0.2146 | | | | Test loss: 0.1238 |
| Grayscale 6 | Training accuracy: 1.0000 | RGB 6 | Training accuracy: 1.0000 | | | Transfer Learning+Fine Tune MobileNet | Training accuracy: 0.9986 |
| | Training loss: 2.8751e-06 | | Training loss: 1.1524e-05 | | | | Training loss: 0.0055 |
| | Test accuracy: 0.9605 | | Test accuracy: 0.9629 | | | | Test accuracy: 0.9915 |
| | Test loss: 0.3116 | | Test loss: 0.2095 | | | | Test loss: 0.0445 |
| | | RGB 7 | Training accuracy: 1.0000 | | | Augmentacija 3 + FT | Training accuracy: 1.0000 |
| | | | Training loss: 2.0969e-05 | | | | Training loss: 1.3083e-06 |
| | | | Test accuracy: 0.9621 | | | | Test accuracy: 0.9871 |
| | | | Test loss: 0.1117 | | | | Test loss: 0.1642 |

*Slika 11. Analiza svih treniranih modela - tačnost i funkcija gubitka*

Referenca praktičnog istraživačkog rada:
https://drive.google.com/drive/folders/1sH0DN4EX7ZPMTUtDWPqoL44a_Y_8BXQO

## 10. Literatura


[1] https://www.kaggle.com/datasets/ahmedkhanak1995/sign-language-gesture-images-dataset

[2] https://www.kaggle.com/code/madz2000/cnn-using-keras-100-accuracy

[3] https://www.kaggle.com/datasets/datamunge/sign-language-mnist

[4] Adam: A Method for Stochastic Optimization, Diederik P. Kingma, Jimmy Ba, Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[5]https://towardsdatascience.com/sign-language-recognition-with-advanced-computer-vision-7b74f20f3442

[6] https://www.kaggle.com/code/themlphdstudent/sign-classification-using-tl-inceptionv3

[7] https://digitalcommons.kennesaw.edu/cgi/viewcontent.cgi?article=1024&context=cs_etd

[8] Satwik Ram Kodandaram, Sunil Gl, N. Pavan Kumar. „Sign Language Recognition“. Turkish Journal of Computer and Mathematics Education (TURCOMAT) (2021) https://www.researchgate.net/publication/353141966_Sign_Language_Recognition

[9] Dutta, K.K., Bellary, S.A.S.: Machine Learning Techniques for Indian Sign Language Recognition. Int. Conf. Curr. Trends Comput. Electr. Electron. Commun. CTCEEC 2017. 333–336 (2018). https://doi.org/10.1109/CTCEEC.2017.8454988

[10] Mandeep Kaur Ahuja, Amardeep Singh, Hand Gesture Recognition Using PCA, International Journal of Computer Science Engineering and Technology (IJCSET), Volume 5, Issue 7, pp. 267-27, (2015) http://ijcset.net/docs/Volumes/volume5issue7/ijcset2015050714.pdf

[11] Nakul Nagpal,Dr. Arun Mitra.,Dr. Pankaj Agrawal, Design Issue and Proposed Implementation of Communication Aid for Deaf & Dumb People, International Journal on Recent and Innovation Trends in Computing and Communication ,Volume: 3 Issue: 5,pp- 147 149.

https://www.researchgate.net/publication/351358081_Design_Issue_and_Proposed_Implementation_of_Communication_Aid_for_Deaf_Dumb_People

[12] Ashish Sethi, Hemanth S,Kuldeep Kumar, Bhaskara Rao N,Krishnan R, SignPro-An Application Suite for Deaf and Dumb, (2012), https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d9663fbea941b3d1feba14d467f461eae744c33b