

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ARHITEKTONSKI FAKULTET

Gestikulator - Platforma za učenje hrvatskoga znakovnog jezika

Nikola Tomažin, Lovro Krčelić (Fakultet elektrotehnike i računarstva)
Bruna Krčelić (Arhitektonski fakultet)

Zagreb, lipanj 2022.

Ovaj rad izrađen je na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu pod vodstvom izv. prof. dr. sc. Tomislava Hrkaća. Predan je na natječaj za dodjelu Rektorove nagrade za timski znanstveni i umjetnički rad (tri do deset autora) u akademskoj godini 2021./2022.

SADRŽAJ

1. Uvod	1
2. Hrvatski znakovni jezik	2
2.1. Jednoručna znakovna abeceda	2
2.2. Dvoručna znakovna abeceda	3
2.3. Geste hrvatskoga znakovnog jezika	5
2.4. Hrvatski savez gluhih i nagluhih	6
3. Anketa	7
4. Implementacija rješenja	9
4.1. Mediapipe	10
4.2. Treniranje modela	14
4.2.1. Skup podataka	14
4.2.2. Model dubokog učenja	14
4.3. Python API	16
4.4. Baza podataka	17
4.5. Backend	17
4.6. Dizajn aplikacije	20
4.7. Frontend	22
5. Rezultati	23
5.1. Prijava i registracija	23
5.2. Početni ekran	24
5.3. Rječnik	27
5.4. Kviz	28
5.4.1. Pitanje video - odgovor tekst	28
5.4.2. Pitanje tekst - odgovor video	29
5.4.3. Pitanje tekst - odgovor kamera	29

5.4.4. Pitanje o gluhoj i nagluhoj kulturi	30
6. Upute za pokretanje	32
6.1. Pokretanje Python djela	32
6.2. Pokretanje baze podataka i backenda	32
6.3. Pokretanje frontenda	33
7. Zaključak	34
Literatura	35

1. Uvod

Prema podacima Svjetske federacije gluhih u svijetu danas živi oko 72 milijuna gluhih osoba. Prema podacima Registra osoba s invaliditetom u Hrvatskoj je 13.460 osoba s težim razmjerima gubitka sluha[6]. Danas osobe s oštećenjem sluha koriste više od 300 različitih znakovnih jezika. Znakovni su jezici punopravni jezici s vlastitom gramatikom i leksikonom, strukturno različiti od govornoga jezika. Pri korištenju znakovnoga jezika ne sudjeluju samo ruke, nego i lice i cijelo tijelo zbog čega rješenja poput senzornih rukavica mogu isporučiti samo dio jezika. Sustav koji bi objedinio detekciju ruku, lica i tijela u stvarnom vremenu te omogućio raspoznavanje gesta bi omogućio raznu primjenu u području razumijevanja, prevođenja i učenja znakovnoga jezika gdje bi se korisniku mogla dati povratna informacija o pravilnom izvođenju pojedine geste.

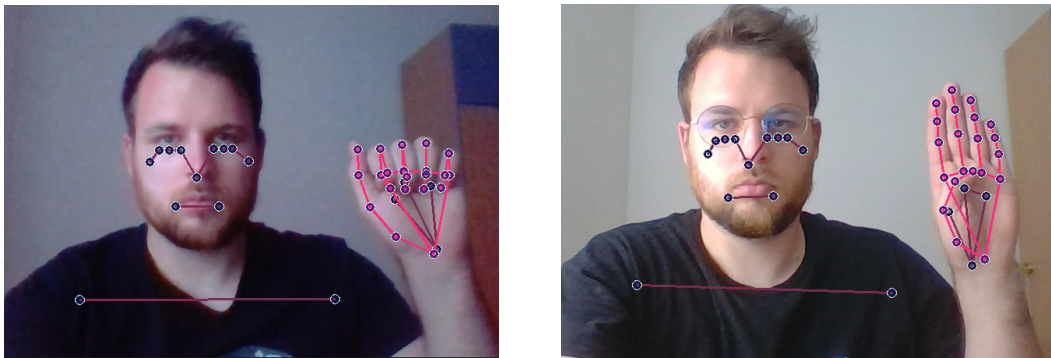
Gestikulator je platforma za učenje osnova hrvatskoga znakovnog jezika (HZJ), ali ne na dosadan način gledanja videa, nego interaktivnije, kroz igru, odnosno rješavanje kvizova gdje bi se poticalo korisnika da uz minimalni napor od pet do petnaest minuta dnevno rješava različite tipove pitanja i tako nauči osnove HZJ-a. Uz to, u sklopu platforme postoji i rječnik hrvatskoga znakovnog jezika kao i aspekt društvene mreže, gdje se korisnici mogu međusobno dodati kao prijatelji te pratiti međusobni napredak i čak natjecati. Aplikacija je prvotno izrađena u sklopu LUMEN Development natjecanja (2022. godine) studentske udruge eSTUDENT gdje je i osvojila prvo mjesto. Na razvoju aplikacije se surađuje s Hrvatskim savezom gluhih i nagluhih (HSGN) kojima se sviđela ideja i koji pružaju kurikulum, videosadržaje o pravilnom izvođenju gesta i povratne informacije.

Razlog je nastanka samog projekta slaba informiranost o gluhoj i nagluhoj kulturi kao i neznanje osnova hrvatskoga znakovnoga jezika. U radu projekta sudjeluju studenti Fakulteta elektrotehnike i računarstva u Zagrebu od druge i pete godine studija te studentica Arhitektonskog fakulteta u Zagrebu četvrte godine studija. Platforma bi koristila svim osobama voljnim učenju osnova hrvatskoga znakovnog jezika kao i pomoć pri edukaciji u slučaju tečajeva ili nastavnih predmeta na fakultetima.

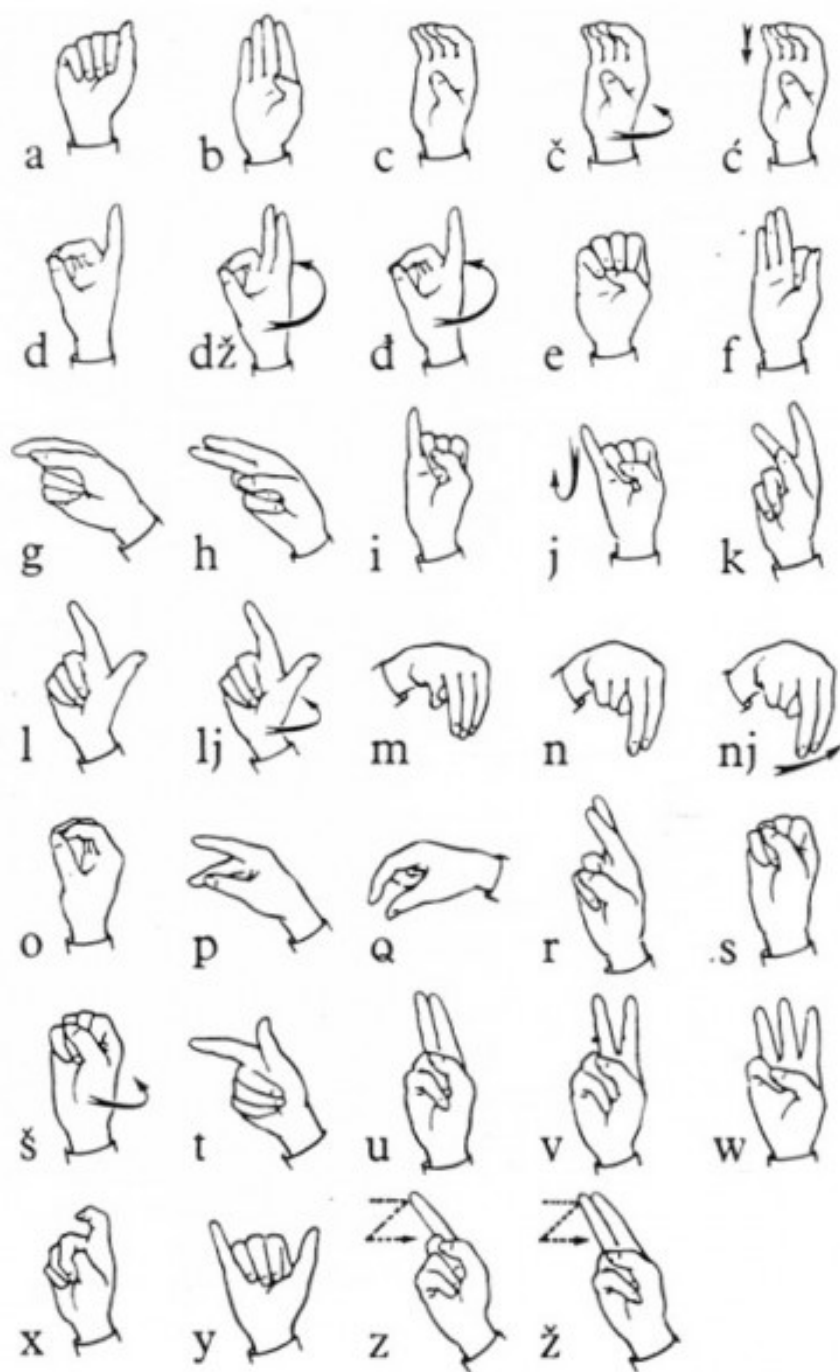
2. Hrvatski znakovni jezik

2.1. Jednoručna znakovna abeceda

Zajednica gluhih i nagluhih osoba u RH upotrebljava dvije vrste ručne abecede, jednoručnu i dvoručnu. Hrvatska jednoručna abeceda je zapravo predefinirana inačica međunarodne (američke) jednoručne abecede s nadodanim slovima poput č, ć, đ, š i ž. Slova nj i lj nemaju poseban znak već su kombinacija znakova l i j, odnosno n i j, dok se slova poput q, w, y i x ne koriste. U jednoručnoj ne razlikujemo velika i mala slova već na sve znakove gledamo kao mala tiskana slova. Jednoručna abeceda većinom se koristi u školama za gluhu i nagluhu djecu. Budući da na svijetu ima više dešnjaka izvođenje jednoručne abecede izvodi se iznad polovice desnog trupa.



Slika 2.1: Primjeri izvođenja slova "a" i "b" jednoručne abecede.



Slika 2.2: Prikaz hrvatske jednoručne znakovne abecede.

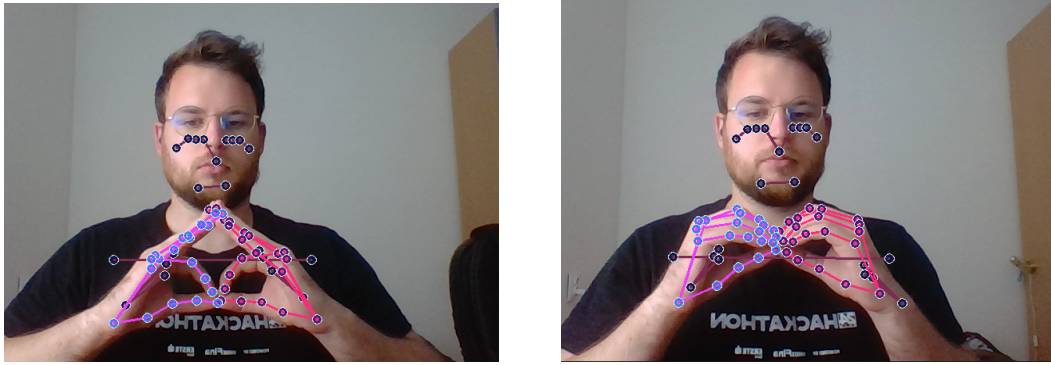
2.2. Dvoručna znakovna abeceda

Za razliku od jednoručne hrvatska dvoručna abeceda proširena je s nekim slovima stranih abeceda (npr. Q, W, X i Y). Suprotno jednoručnoj abecedi znakovi dvoručne abecede prate oblik velikih tiskanih slova Hrvatske abecede. Isto kao i kod jednoručne

abecede nemamo posebne znakove sa slova lj i nj. Dvoručnu abecedu prati duga tradicija u zajednici gluhih, nagluhih te gluhoslijepih osoba. Gluhoslijepe osobe ponekad koriste dvoručnu abecedu iz razloga što se koriste veliki pokreti ruku pa je osobama sa slabim poremećajem vida moguće razumijeti znakove.



Slika 2.3: Prikaz hrvatske dvoručne znakovne abecede.



Slika 2.4: Primjeri izvođenja slova "a" i "b" dvoručne abecede.

2.3. Geste hrvatskoga znakovnog jezika

Za razliku od znakovne abecede geste se izvode cijelim tijelom i licem. Geste su oblik neverbalne ili nevokalne komunikacije kod koje tjelesnim radnjama prikazujemo određene poruke. Geste se dijele u dvije glavne skupine po tipu na komunikativne i informativne. Informativne geste su pasivne geste koje nam govore o govorniku kao osobi, npr. podizanje ruke u školi, dok komunikativne geste koristimo u razgovoru. Prilikom izvođenja samih gesta koristi se cijelo tijelo uz fokus na 6 faktora:

- **usta**, kod kojih je bitno da li se sama riječ izgovara ili se oponaša neki specifičan oblik usnama
- **pogled**, naime smjer pogleda je vrlo bitan kod komunikacije s gluhim i nagluhim osobama, bitno je da se tokom razgovora sugovornici međusobno gledaju
- **položaj obrva** (spuštene ili podignute)
- **treptanje** (broj treptaja)
- **pokreti glave**
- **rotiranje i naginjanje tijela** (uključuje i slijeganje ramenima)

Kod izvođenja gesta izuzetno je bitna i brzina izvođenja te sam stav govornika, jer to puno ukazuje kako se govornik osjeća te s kojom važnošću prikazuje gestu.



Slika 2.5: Primjeri izvođenja geste "gluh".

2.4. Hrvatski savez gluhih i nagluhих

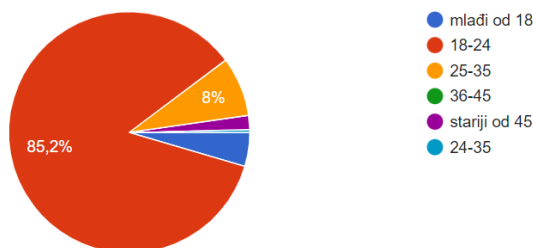
Počeci organiziranih udruživanja gluhih i nagluhих na području RH datira još od 1885. godine, kada je gluhi učitelj Adalbert Lampe zajedno s doseljenicima iz austrougarskih zemalja otvorio školu za gluhe u Zagrebu. Prvo društvo gluhih 'Dobrotvor' registrirano je 1921. godine, a prvi predsjednik bio je gluhi slikar Ivan Smole. To je prvo društvo gluhih na prostoru južno od Beča i Budimpešte. Društvo je postojalo pod navedenim imenom sve do 1945. godine. Poslije rata osnivaju se dvije nove udruge na području Splita udruga i Zagreba, te se kroz godine organiziraju brojni kulturni i sportski događaji za gluhe i nagluhe. Danas postoji Hrvatski savez gluhih i nagluhих s centrom u Zagrebu s kojim smo još u veljači 2022. godine stupili u kontakt. Početkom ožujka predstavili smo im ideju o izradi aplikacije za učenje HZJ te ih upitali za pomoć pri izradi same aplikacije. Dogovorili smo se kako će nam oni izraditi kurikulum po kojem će korisnici naše aplikacije učiti znakovni jezik te kako će nam kroz vrijeme dostavljati prijekopotrebne video snimke gesta, na temelju kojih treniramo naš AI model za validaciju ispravnosti izvođenja gesta.

3. Anketa

Kako bi se ispitala upućenost u hrvatski znakovni jezik kao i interes o aplikaciji za učenje osnova hrvatskoga znakovnog jezika proveli smo anketu pod nazivom: *Istraživanje o interesu o aplikaciji za učenje Hrvatskog znakovnog jezika*. Uzorak koji smo ispitali su bili uglavnom mlađa dobna skupina, slika 3.1, te se pokazalo da malo više od 25% ispitanika poznaje gluhu ili nagluhu osobu, slika 3.2. Malo veći broj ispitanika također zna i neke osnove hrvatskoga znakovnog jezika, bila to hrvatska znakovna abeceda ili pak neke geste HZJ-a, slika 3.3. Najbitniji zaključak koji smo izvukli iz ankete je da postoji veliki interes za nekakvom aplikacijom gdje bi se olakšalo sporazumijevanje s gluhom ili nagluhom osobom, slika 3.4, što nam ukazuje da postoji područje i potreba za našom platformom.

Dobna skupina

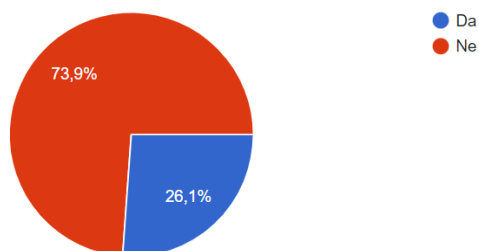
264 odgovora



Slika 3.1: Dobne skupine ispitanika.

Poznajete li gluhu ili nagluhu osobu?

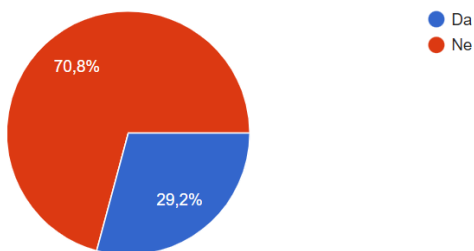
264 odgovora



Slika 3.2: Poznavanje gluhe ili nagluhe osobe.

Znate li Hrvatsku znakovnu abecedu ili neke geste Hrvatskog znakovnog jezika (HZJ)?

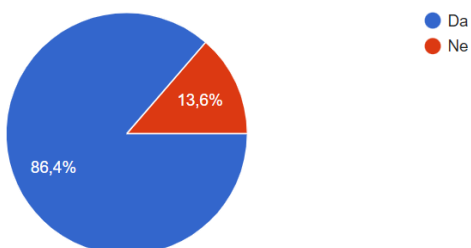
264 odgovora



Slika 3.3: Poznavanje nekih osnova hrvatskoga znakovnoga jezika.

Biste li bili zainteresirani za aplikaciju koja bi Vam omogućila lakše sporazumijevanje s gluhom ili nagluhom osobom?

264 odgovora



Slika 3.4: Zainteresiranost za aplikaciju olakšavanje komunikacije s gluhom ili nagluhom osobom.

4. Implementacija rješenja

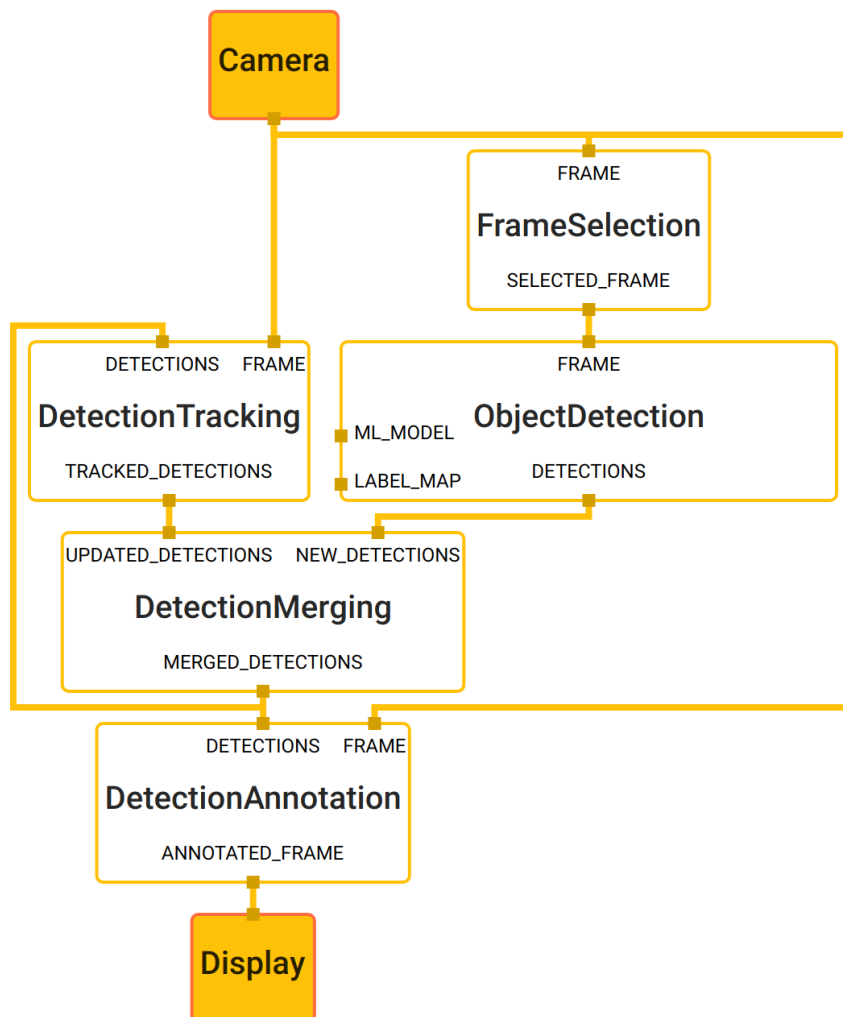
Pri razvoju platforme koristile su se razne tehnologije:

- MediaPipe - Googleova biblioteka za procjenu poze. Korišteno je rješenje *Holistic*.
- Python - programski jezik
- OpenCV - biblioteka programskih funkcija koja je uglavnom usmjerena na računalni vid u stvarnom vremenu te služi za sve funkcionalnosti s kamerom
- NumPy - biblioteka za programski jezik Python, koja dodaje podršku za velike, višedimenzionalne nizove i matrice, zajedno s velikom zbirkom matematičkih funkcija visoke razine za rad na tim nizovima
- Keras - biblioteka koja pruža Python sučelje za umjetne neuronske mreže
- TensorFlow - biblioteka za strojno učenje i umjetnu inteligenciju
- PostgreSQL - besplatan sustav za upravljanje bazama podataka otvorenog koda
- Liquibase - biblioteka s neovisnom bazom podataka otvorenog koda za praćenje, upravljanje i primjene promjena sheme baze podataka
- Java - programski jezik
- Spring Boot - radni okvir pisan je u programskom jeziku Java koji služi za izradu backenda web aplikacija
- Figma - uređivač vektorske grafike i alat za izradu prototipa koji se prvenstveno temelji na webu
- ReactJS - Javascript biblioteka korištena za izradu korisničkog sučelja

U nastavku ovoga poglavlja će najbitnije korištene tehnologije biti detaljnije objašnjene.

4.1. Mediapipe

MediaPipe je okvir za izgradnju cjevovoda za izvođenje zaključivanja preko proizvoljnih senzornih podataka. S MediaPipeom se percepcijski cjevovod može izgraditi kao graf modularnih komponenti, uključujući zaključivanje modela, algoritme za obradu medija i transformacije podataka itd. Sensorni podaci kao što su audio i video tokovi ulaze u graf, a percipirani opisi kao što su lokalizacija objekta i tokovi orijentira na lice izlaze iz grafa. Primjer je prikazan u slici 4.1.



Slika 4.1: Detekcija objekta pomoću MediaPipe biblioteke.

MediaPipe je dizajniran za korištenje za osobe u području strojnog učenja (engl. Machine learning - ML), uključujući istraživače, studente i programere, koji implementiraju ML aplikacije spremne za proizvodnju, objavljuju istraživačke radove i izrađuju tehnološke prototipove. Glavni slučaj upotrebe MediaPipea je brza izrada prototipa percepcijskih cjevovoda s zaključivanjem modela i druge komponente za više-

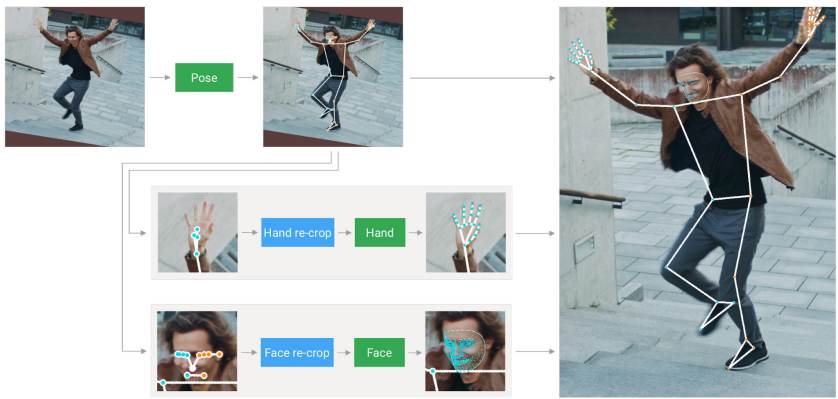
kratnu upotrebu. MediaPipe također olakšava implementaciju tehnologije percepcije u demonstracije i aplikacije na širokom rasponu različitih hardverskih platformi.

Postoje razna rješenja strojnog učenja u MediaPipeu, kao što su detekcija lica (engl. Face detection), mreža za lice (engl. Face mesh), detekcija šarenice, detekcija ruku, detekcija poze, holistička detekcija, segmentacija kose, detekcija objekata i mnogi drugi¹.

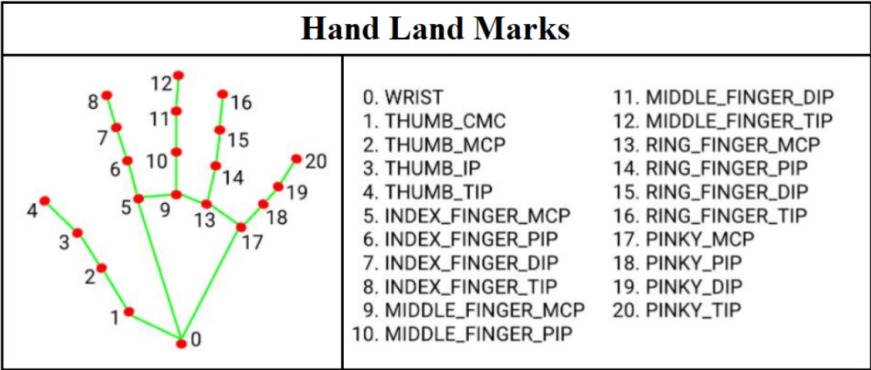
MediaPipe već nudi brza i točna, a opet odvojena rješenja za simultanu detekciju lica, ruku i poze. Kombiniranje svih njih u stvarnom vremenu u semantički konzistentno end-to-end rješenje je jedinstveno težak problem koji zahtijeva istovremeno zaključivanje više, ovisnih neuronskih mreža. MediaPipe rješenje koje je najadekvatnije za dani problem je *MediaPipe Holistic* rješenje[11]. MediaPipe Holistic cjevovod integrira zasebne modele za komponente poze, lica i ruku, od kojih je svaki optimiziran za svoju određenu domenu. Međutim, zbog njihove različite specijalizacije, ulaz u jednu komponentu nije dobro prikladan za druge. Model procjene poze, na primjer, uzima niži videookvir fiksne rezolucije (256x256) kao ulaz. Ali ako bi netko izrezao regije šake i lica s te slike kako bi ih prešao na njihove odgovarajuće modele, rezolucija slike bila bi preniska za točnu artikulaciju. Stoga je MediaPipe Holistic dizajniran kao višestupanjski cjevovod koji tretira različite regije koristeći prikladnu razlučivost slike za regiju.

Prvo se procjenjuje ljudska poza na vrhu slike 4.4 s BlazePose detektorom poze i naknadnim modelom orijentira. Zatim, koristeći zaključene orijentire poza, izvode se tri regije područja interesa (engl. Region of Interest - ROI) za svaku ruku (dva puta) i lice, te se koristi model ponovnog obrezivanja kako bi se poboljšao ROI. Zatim se izrezuje ulazni okvir pune razlučivosti na te regije interesa i primjenjuju se specifični modeli lica i ruku za procjenu njihovih odgovarajućih orijentira. Konačno, spaja se sve orijentire s onima iz modela poze kako bi se dobilo punih 543 orijentira (33 orijentira poza, 468 orijentira lica i 21 ručnih orijentira po ruci), odnosno karakterističnih točaka. Svaki orijentir sadrži kartezijske koordinate (x, y, z) te informaciju o vidljivosti.

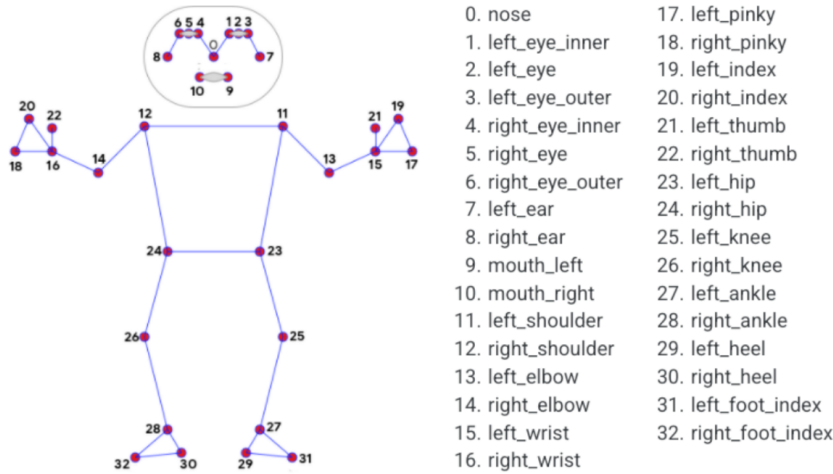
¹<https://google.github.io/mediapipe/#ml-solutions-in-mediapipe>



Slika 4.2: Primjer MediaPipe Holistic cjevovoda.



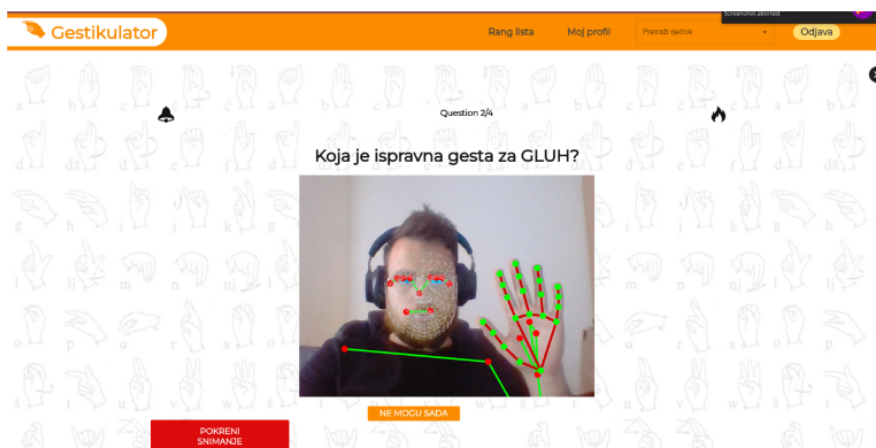
Slika 4.3: Karakteristične točke ruku.



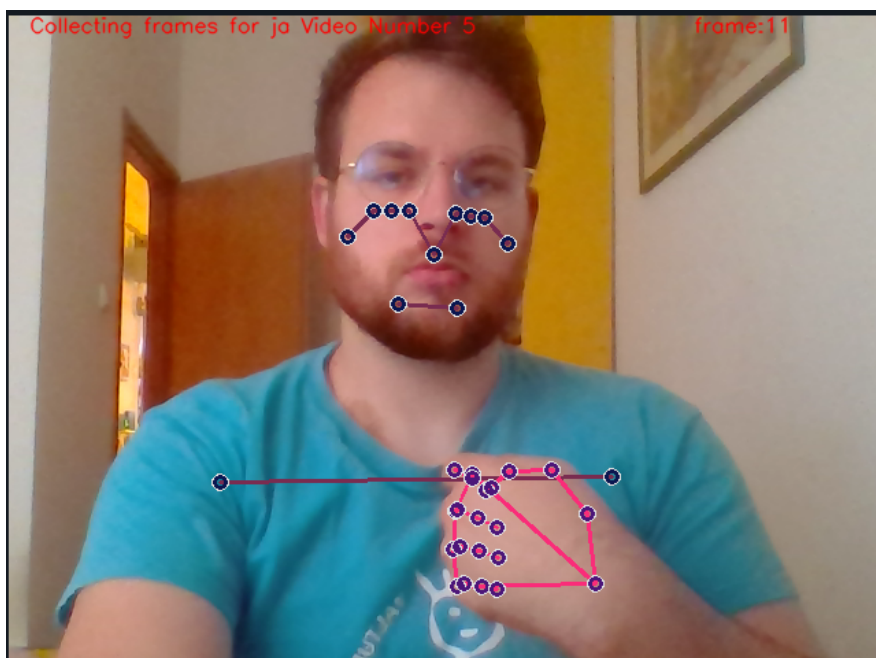
Slika 4.4: Karakteristične točke poze.

Jedna od glavnih odlika ove Googleove biblioteke je što ima podršku na raznim

platformama, od mobilnih uređaja, do stolnih i prijenosnih računala, pa sve do web aplikacija². Iz tog razloga je MediaPipe bio glavni kandidat za biblioteku koja će detektirati lice, ruke i pozu. U platformi se MediaPipe koristi u dvije instance: jedna za detekciju karakterističnih točaka na web aplikaciji (primjer slika 4.5) te druga koja je služila pri treniranju modela gdje su se detektirale karakteristične točke kroz Python kôd koji se izvodio na prijenosnom računalu (primjer slika 5.1).



Slika 4.5: Detekcija karakterističnih točaka na webu.



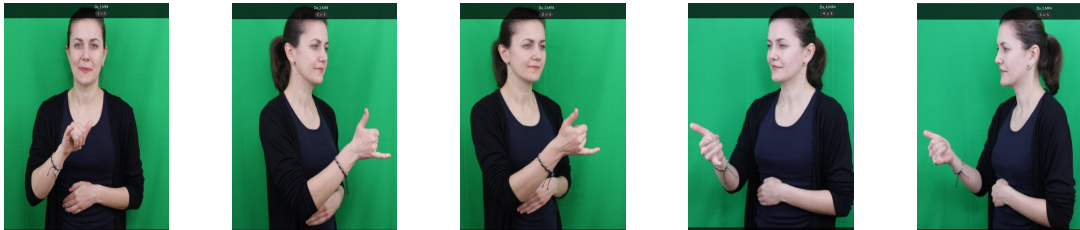
Slika 4.6: Detekcija karakterističnih na desktop aplikaciji kroz Python.

²https://google.github.io/mediapipe/getting_started/getting_started.html

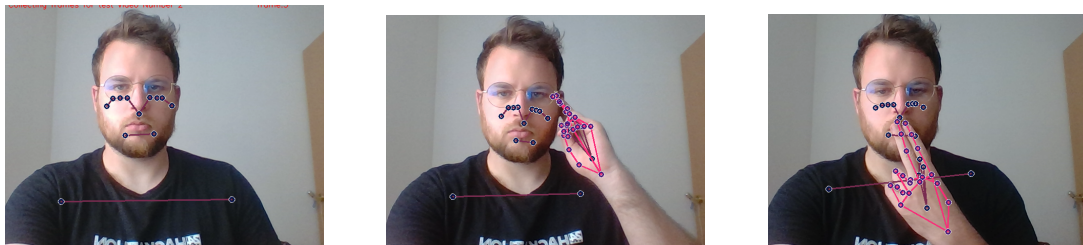
4.2. Treniranje modela

4.2.1. Skup podataka

Da bi se model mogao trenirati potrebno je prvo imati skup podataka. U ovom slučaju, za hrvatski znakovni jezik, takav skup podataka ne postoji te smo ga morali sami izgraditi. Kako bismo to uradili ispravno, surađivali smo s Hrvatskim savezom gluhih i nagluhih, koji su nam slali videozapise ispravo izvedenih gesta iz više kuteva (prikazano na slici 4.7) na temelju kojih smo se snimali više puta (100 - 200 puta po gesti) kako bi izgradili skup podataka. Primjer jednog takvog izvođenja gesta vidljiv je na slici 4.8. Skup podataka zapravo nisu videozapisi izvođenja gesta, nego skup ranije opisanih orijentira kroz svaki okvir videozapisa. Ti orijentiri se spremaju u NumPy datoteke za svaki snimljeni okvir.



Slika 4.7: Primjer geste "Da" iz raznih kuteva



(a) Početak geste gluh

(b) Sredina geste gluh

(c) Kraj geste gluh

Slika 4.8: Izvođenje geste gluh

4.2.2. Model dubokog učenja

Sada kada napokon imamo skup podataka, možemo izgraditi model dubokog učenja. Budući da je problem raspoznavanja geste problem koji se odvija u vremenu ne možemo se osloniti na jednostavne neuronske mreže, a ni na konvolucijske neuronske mreže. Stoga se za glavnu arhitekturu rješavanja problema uzimaju povratne neuronske mreže.

Ulaz u mrežu će biti dvodimenzionalan, jedna dimenzija će biti skup koordinata i vidljivosti orijetira (sveukupno 1662 podataka), a druga dimenzija će biti 30 budućih da je to vremenski period u okvirima koji smo promatrali. Nakon toga će se nalaziti par slojeva s LSTM ćelijama ReLU aktivacijskim funkcijama nakon kojih slijede gusto povezani slojevi te na kraju kao izlaz mreže imamo jednodimenzionalni vektor čija je veličina ovisna o broju kategorija gesta nad kojima smo trenirali model. Taj jednodimenzionalni vektor nam označuje s kolikom mjerom dani ulaz pripada kojoj gesti. Primjer tako definirane mreže vidljiv je u kodu 4.2.2.

```
model = Sequential()
model.add(LSTM(64, return_sequences=True,
              activation='relu', input_shape=(30, 1662)))
model.add(LSTM(128, return_sequences=True,
              activation='relu'))
model.add(Dropout(0.2))
model.add(LSTM(64, return_sequences=False,
              activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(actions.shape[0],
              activation='softmax'))
```

Kod 1: Isječak koda definiranja neuronske mreže

Zbog nedovoljno računalnih resursa nije bilo moguće trenirati model s potpunim skupom podataka, stoga se uzimalo nasumično određen broj gesta nad kojima je bio treniran model. Najbolji rezultati su se dobili kada je model bio treniran na 7 gesta, što naravno ovisi o veličini podatkovnog skupa, odabiru gesta te nasumičnosti treniranja. Istreniran model je na kraju spremljen te se uz njega sprema i tekstualna datoteka u kojoj se nalaze informacije nad kojim je gestama model istreniran. Također je bitno naglasiti da taj istreniran model nije savršen za sve geste te će najčešće dobro funkcionirati za pet od sedam gesta bilo to zbog neproporcionalnosti skupa podataka ili kompleksnosti geste stoga treba voditi evidenciju o informaciji koji model radi uspješno za koju gestu. Iz tog je razloga definirana *models_config.json* (čiji je dio sadržaja vidljiv u kodu 4.2.2) datoteka koja u sebi sadrži informacije koji modeli uspješno raspoznaju koju gestu.

```
{
  "gluh": {
```

```

    "name": "../models/model_112.h5",
    "model": null,
    "info": "../models/model_112.txt"
  },
  "godina": {
    "name": "../models/model_74.h5",
    "model": null,
    "info": "../models/model_74.txt"
  },
  "ime": {
    "name": "../models/model_112.h5",
    "model": null,
    "info": "../models/model_112.txt"
  },
  "nista": {
    "name": "../models/model_74.h5",
    "model": null,
    "info": "../models/model_74.txt"
  },
  ...
}

```

Kod 2: Isječak *models_config.json* datoteke koja sadrži informacije koji modeli uspješno raspoznaju koje geste.

4.3. Python API

Kako bi web ili mobilne aplikacije mogle iskoristiti prethodno objašnjeni istrenirani model, potrebno je omogućiti komunikaciju s Python kodom koji vrši predikciju geste. Iz tog razloga je također razvijena *Flask* aplikacija koja služi kao "most" prema modelu, odnosno aplikacijsko programsko sučelje ili sučelje za programiranje aplikacija (engl. Application programming interface - API). Ova Flask aplikacija pri paljenju učitava sve modele za sve geste iz *models_config.json* datoteke te pokrene server koji na kranjoj točki (engl. Endpoint) */predict* sluša na *POST* zahtjeve koji u svom tijelu sadrže naziv tražene geste te skup karakterističnih točaka. Nakon što zaprimi zahtjev procesiraju se ulazni karakteristične točke u format koji je prikladan za model dubokog učenja

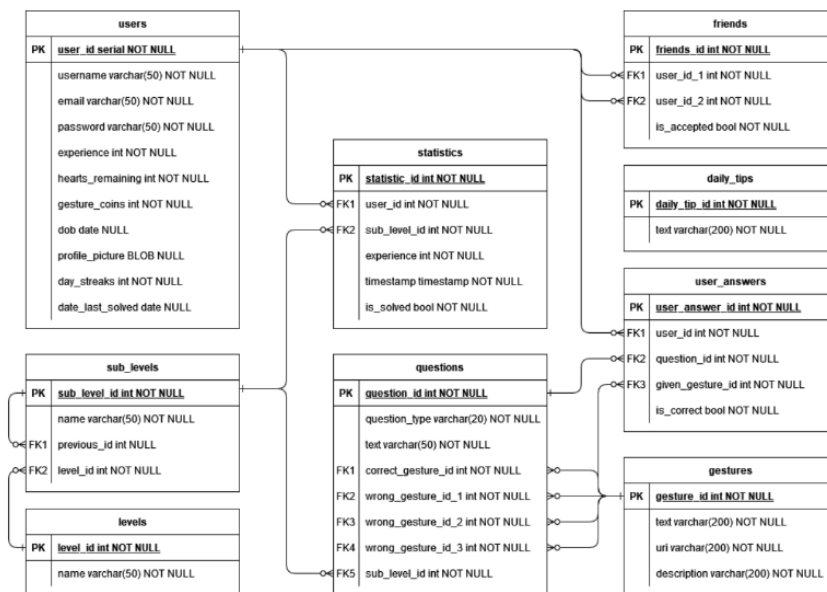
te nakon što se to obavi, poziva se funkcija predviđanja koja nam za određeni skup karakterističnih točaka kroz 30 okvira odredi o kojoj se gesti radi. Ukoliko je predviđena gesta jednaka nazivu geste poslanom u zahtjevu, server šalje pozitivan odgovor na upit čime servis koji je poslao zahtjev zna da je tražena gesta ispravno izvedena.

4.4. Baza podataka

U bazi podataka pohranjene su informacije o:

- Registriranim korisnicima
- Pitanjima organiziranim u razine i podrazine
- Gestama (korištene u pitanjima i u rječniku)
- Rezultatima rješavanja kvizova

ER model baze podataka je vidljiv na slici 4.9.

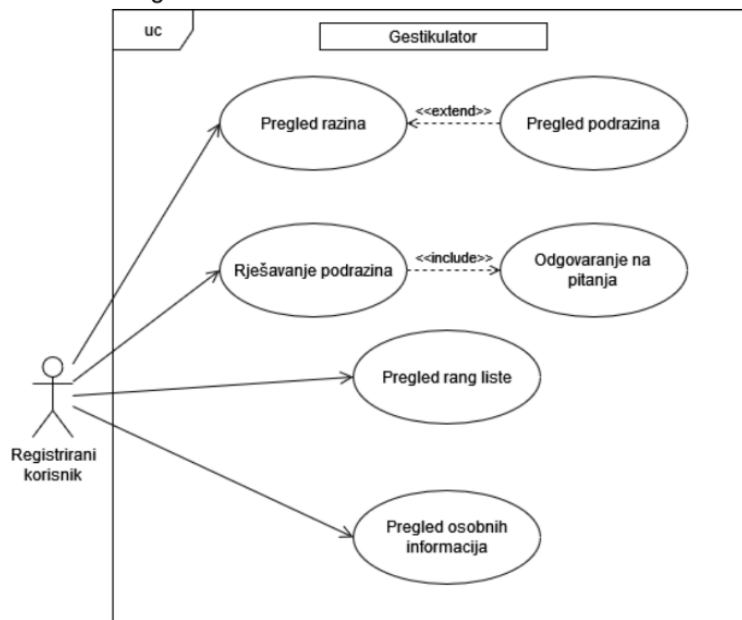


Slika 4.9: ER model baze podataka.

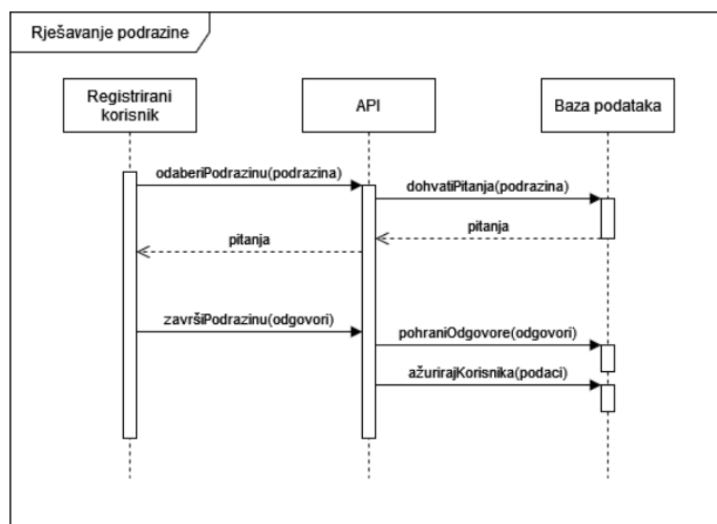
4.5. Backend

Backend aplikacije oblikovan je višeslojnom arhitekturom kojom se razdvajaju zadaci i ograničavaju pristupi slojevima. Na taj način se postiže “separation of concerns”. Korišteni slojevi su:

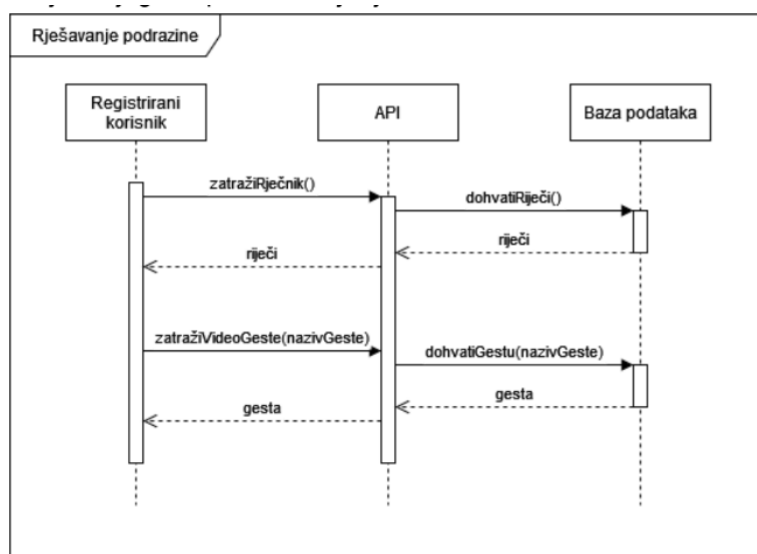
- Podatkovni sloj - provodi komunikaciju s bazom podataka i izvodi naredbe dohvata, pohrane, izmjene i brisanja podataka. Ovaj sloj ostvaren je sučeljima koja se zovu repozitoriji. Podaci korišteni u aplikaciji oblikovani su u entitete, odnosno objektivne reprezentacije pojedine relacije baze podataka.
- Logički sloj - sloj poveznica između podatkovnog i prezentacijskog sloja u kojem se provodi napredna obrada podataka i priprema ih se na posluživanje korisniku ili na pohranu u bazu podataka. Ovaj sloj oblikuje se sučeljima i klasama koje implementiraju ta sučelja te ih zajedno zovemo servisima.
- Prezentacijski sloj - sadrži pristupne točke za klijentsku aplikaciju na kojima backend dio poslužuje i prihvaća podatke od i za korisnika. Ovaj sloj izveden je klasama koje zovemo kontrolerima.



Slika 4.10: Use case dijagram.



Slika 4.11: Sekvencijski dijagram rješavanja kviza.



Slika 4.12: Sekvencijski dijagram pretraživanja rječnika.

Postoji nekoliko pristupnih točaka koje API opskrbljuje (u testnoj fazi API se nalazi na adresi <http://localhost:8080>):

- levels/main - pristupna točka na koju se šalje identifikacijski broj korisnika te API vraća informacije o korisniku, razine i podrazine zajedno s informacijom o riješenosti. Dodatno, API vraća izazove i aktivnosti korisnika
- levels/summary - pristupna točka kojoj se šalju podaci o korisnikovim odgovorima na pitanja u kvizu. API na osnovu primljenih podataka ažurira korisnikove podatke (životu, iskustvo) i vrši evidenciju odgovora na pitanja

- gestures - pristupna točka za dohvat rječnika gesti
- gestures/gestureText - pristupna točka za dohvat videa geste *gestureText*

4.6. Dizajn aplikacije

Dizajn aplikacije razvijen je u alatu Figma. Figma je uređivač vektorske grafike i alat za izradu prototipa koji se prvenstveno temelji na webu, s dodatnim izvanmrežnim značajkama koje omogućuju desktop aplikacije za macOS i Windows. Aplikacija je dizajnirana *user friendly* kako bi bila pristupna svim uzrastima, jednostavna i intuitivna. Raspored komponenti podijeljen je na važnije i malo manje važne odvajanjem svakog u svoje okvire. Svaki okvir nam nudi nove opcije koje su tekstualno i vizualno opisane iznad svakog kratkim naslovom. Gornji dio je sastavljen od logotipa, dodatnih opcija, search bara te odjavnog gumba. Srednji lijevi dio služi kao manja komponenta koja nas potiče na veće korištenje aplikacije prikazujući nadolazeće izazove te aktivnost naših prijatelja. U srednjem desnom djelu nalazimo glavni dio koji je istaknut dvjema trakama izabirući razinu i kategoriju koju želimo učiti. Odabir razine i kategorije je jednostavan; klikom na opciju odabiremo, a klikom na strelicu koja se nalazi na traci gledamo ostale ponuđene opcije.



Slika 4.13: Prikaz početnog ekrana u alatu Figma.

Započevši kviz traka nam naglašava, uz brojku, na kojem smo pitanju prikazujući njen početak i kraj. Klikom odabiremo svoje odgovore te klikom na "dalje" krećemo na sljedeće pitanje.



Slika 4.14: Prikaz jednog od ekrana za kviz u alatu Figma.

Boje su izabrane prema sagledavanjem uočljivih no neometajućih nijansi. Također, narančasta je boja koja u UI dizajnu označava samouvjerenost i strast te je poznata po tome da budi uzbuđenje.

Logotip je nastao na temelju samog znakovnoga jezika prikazujući slovo "g" kao prvo slovo naziva same aplikacije. On je prenesen, no pojednostavljen, sadržeći boje same aplikacije kako bi sve postalo jednom cjelinom.



Slika 4.15: Dizajn logotipa Gestikulatora.

4.7. Frontend

Klijentski dio sustava je razvijen pomoću ReactJSa. React je JavaScript biblioteka otvorenog koda za izradu korisničkog sučelja koju je napravio i održava Facebook. Dizajn samog sučelja ostvaren je uporabom komponenti iz MUI biblioteke komponenata koju nudi Google, te je njihovom prilagodbom za našu aplikaciju dobiven konkretan dizajn.

Kao što je već bilo prije opisano, MediaPipe ima i podršku za web. To nam je omogućilo stvarnom vremenu pomoću korisnikove kamere detektiramo karakteristične točke te ih spremimo u listu kako bi ih kasnije mogli poslati na Python Flask aplikaciju koja sadrži model dubokog učenja koji služi za predikciju geste. Primjer prikaza detekcije karakterističnih točaka na web aplikaciji je vidljiv na slici 4.5. Izgled ekrana i glavnih komponenti bit će prikazan u sljedećem poglavlju.

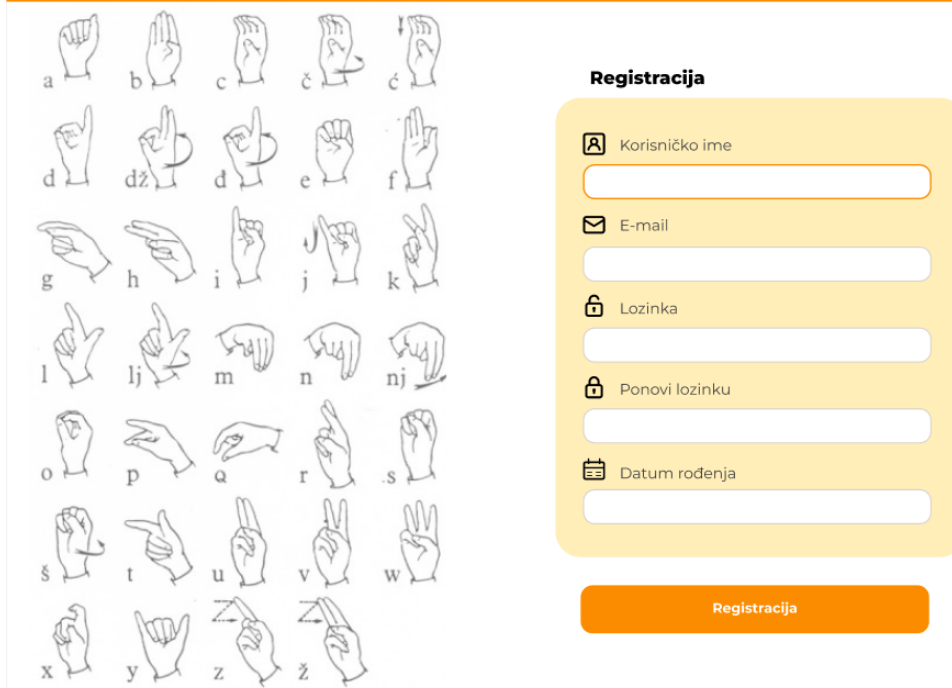
5. Rezultati

5.1. Prijava i registracija

Kako bi korisnik mogao prvi put pristupiti platformi potrebno je registrirati se. Trenutno je registracija omogućena preko standardnih polja za popunjavanje kao što su:

- Korisničko ime
- E-mail adresa
- Lozinka
- Ponovljena lozinka
- Datum rođenja.

Prozor za prijavu izgleda slično kao i za registraciju samo korisnik tu mora unijeti *korisničko ime ili e-mail adresu i lozinku.*



Slika 5.1: Prikaz ekrana za registraciju.

5.2. Početni ekran

Početni se ekran (prikazan na slici 5.2) sastoji od više dijelova. U središnjem dijelu se nalaze izbornici za odabir razine i podrazine gdje klikom na razinu mijenjanju podrazine. S lijeve strane se nalaze podaci o korisniku gdje je pri vrhu prikazana slika korisnika, ime te *daily streak*, odnosno broj koji označuje koliko je dana za redom korisnik rješio neki kviz na platformi. Ispod toga se nalaze dijelovi s izazovima te aktivnostima, gdje se unio aspekt društvene mreže i gejmfikacije kako bi se korisnika poticalo da što više koristi platformu.

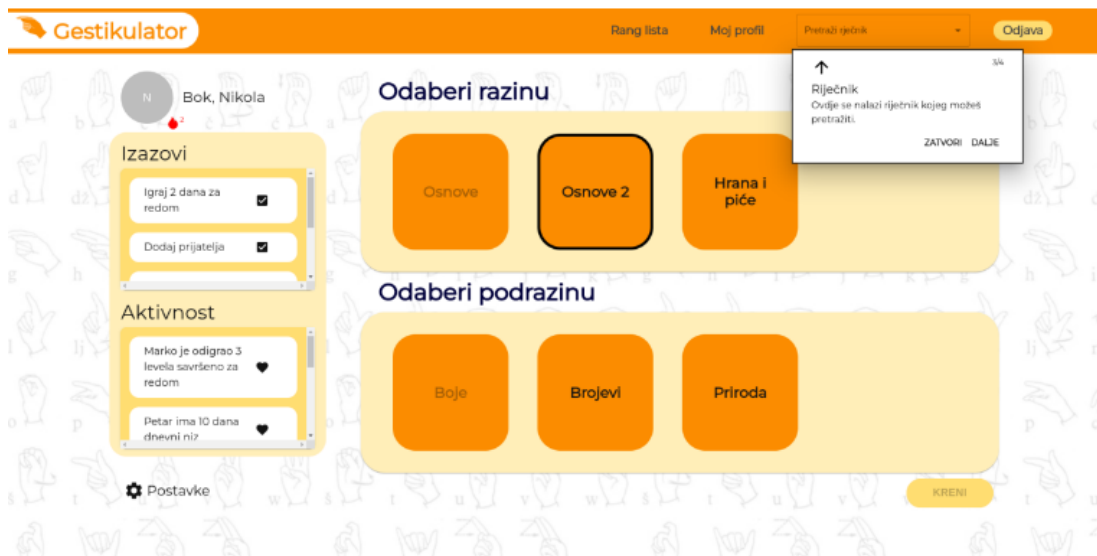


Slika 5.2: Prikaz ekrana za registraciju.

Iznad odabira razine s desne strane se nalaze gumb za pomoć, gdje novi korisnici mogu naučiti funkcije pojedinih dijelova platforme i kako je koristiti, a to je prikazano na slikama 5.3, 5.4, 5.5. Pored toga se nalazi gumb sa zanimljivosti dana, gdje se klikom na to otvara novi modal koji pokazuje današnju zanimljivost (prikazano na slici 5.6). Zanimljivost dana može biti informacija o gluhoj i nagluhoj kulturi, povijesti ili pak o nastanku pojedinih gesti.



Slika 5.3: Prikaz modala za pomoć.



Slika 5.4: Prikaz modala za pomoć.



Slika 5.5: Prikaz modala za pomoć..

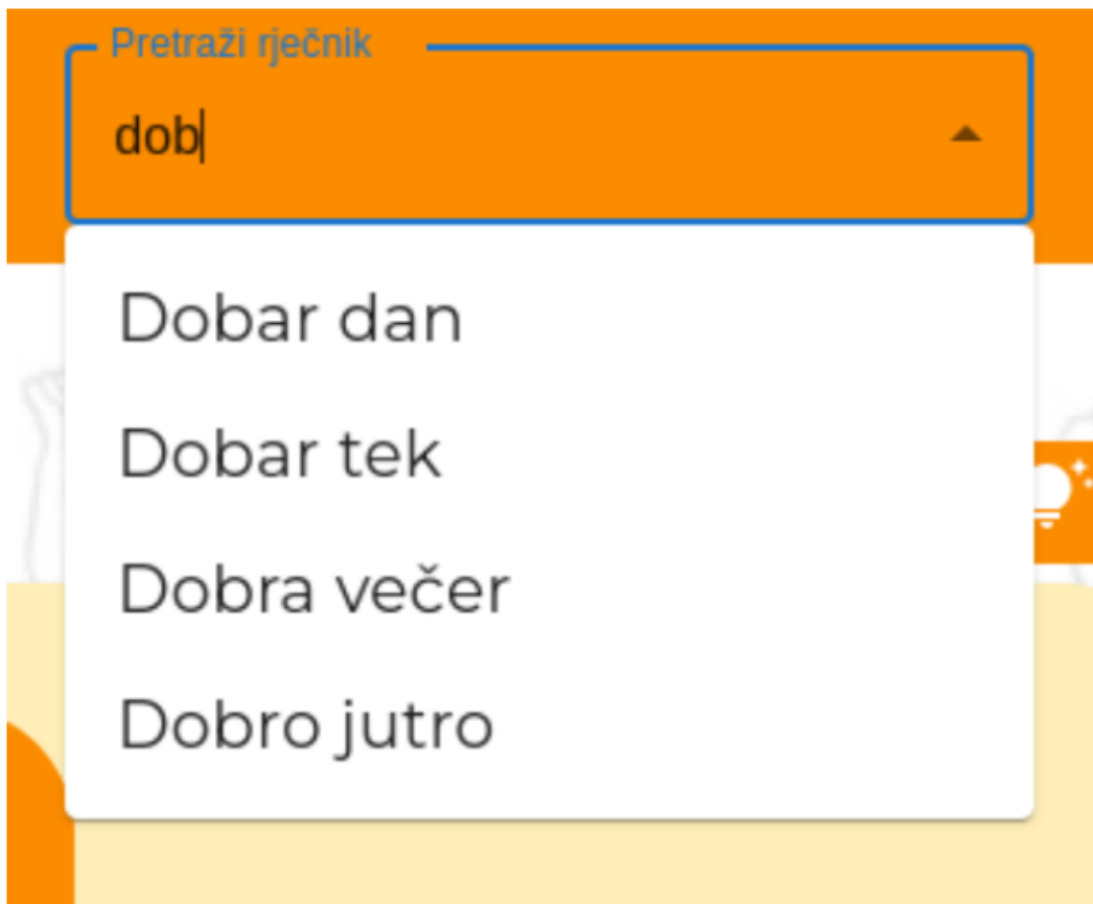


Slika 5.6: Prikaz zanimljivosti dana.

5.3. Rječnik

Zbog drugačijega modaliteta hrvatskoga znakovnog jezika i nedostatka standardiziranoga načina njegova bilježenja, pokušaji stvaranja tiskanih rječnika HZJ-a nisu urodili kvalitetnim radovima pa potreba za jednim kvalitetnim rječnikom još uvijek postoji. HZJ svojom modalnošću zahtijeva nov medij rječnika, bogat multimedijem te dodatnim podacima i sadržajima koji za rječnike govorenih jezika nisu potrebni[24]

U alatnoj traci na vrhu, osim rang lista i profila se nalazi pretraživač za rječnik. Polje za pretraživanje nudi mogućnost dovršavanja riječi (engl. Autocomplete) što je prikazano na slici 5.11. Odabirom geste koju želimo pretražiti otvara se novi model koji prikazuje tu gestu.



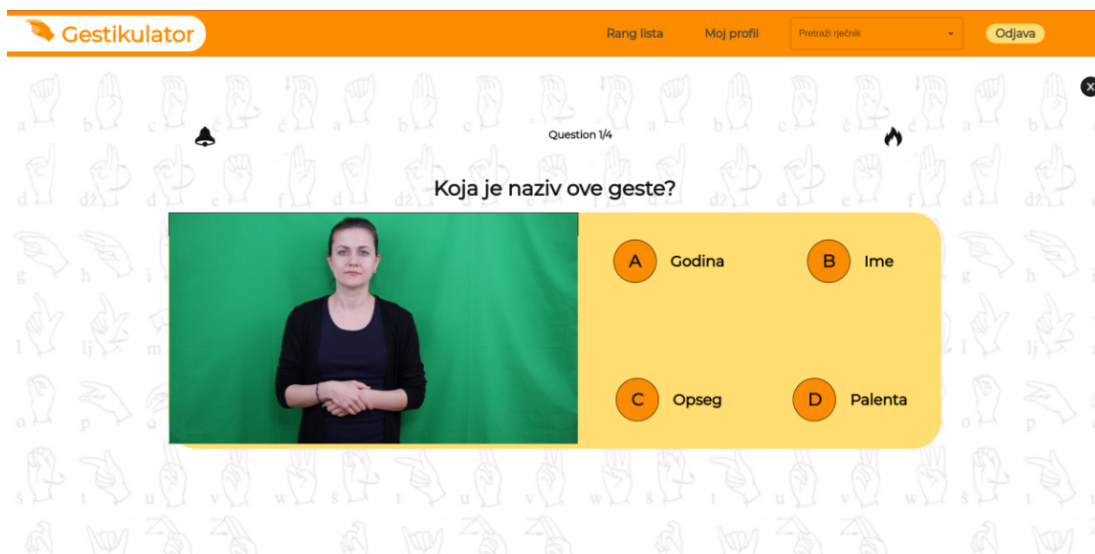
Slika 5.7: Prikaz pretraživanja rječnika.

5.4. Kviz

Glavni dio platforme je kviz. Nakon što korisnik na početnoj stranici odabere razinu i podrazinu koju želi proći i klikom na gumb *Kreni* pokreće se kviz. Kvizovi se sastoje od više pitanja (10 - 15), a ona su u jednom od 4 formata opisanih u sljedećem potpoglavlju. Kod kvizova nam je bitno da korisnik ima različite načine učenja, tj. formate pitanja/odgovora kako mu ne bi brzo dosadilo i kako bi se angažirao naučiti prepoznati gestu, prepoznati naziv geste te pravilno izvesti gestu.

5.4.1. Pitanje video - odgovor tekst

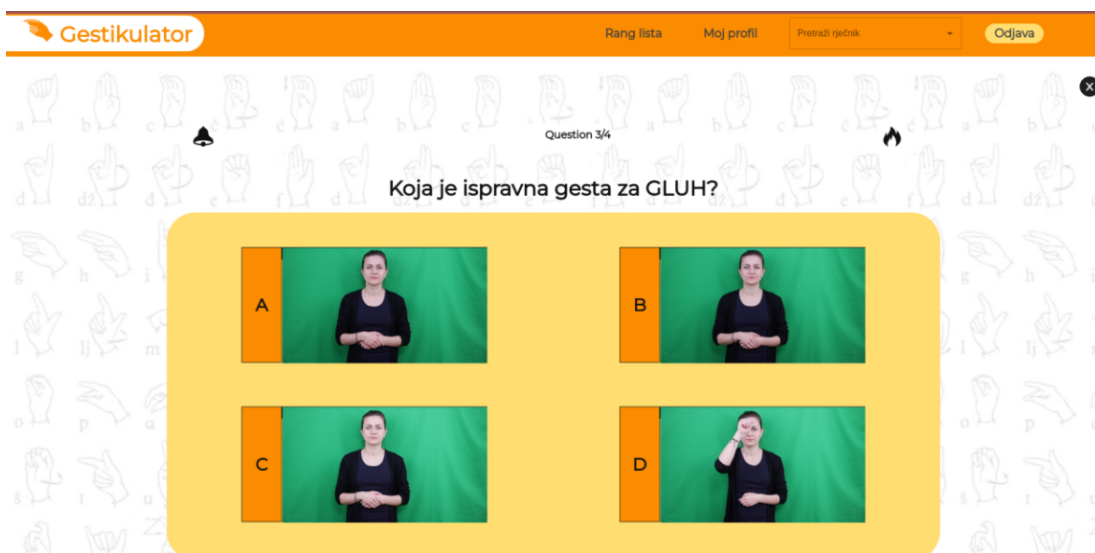
Uz pitanje se nalazi i videosadržaj koji korisnik treba pogledati te zaključiti o kojoj se gesti radi te odabrati ispravan naziv.



Slika 5.8: Pitanje video - odgovor tekst.

5.4.2. Pitanje tekst - odgovor video

U pitanju se nalazi naziv geste za koju korisnik mora odabrati pravilni video izvršavanja.

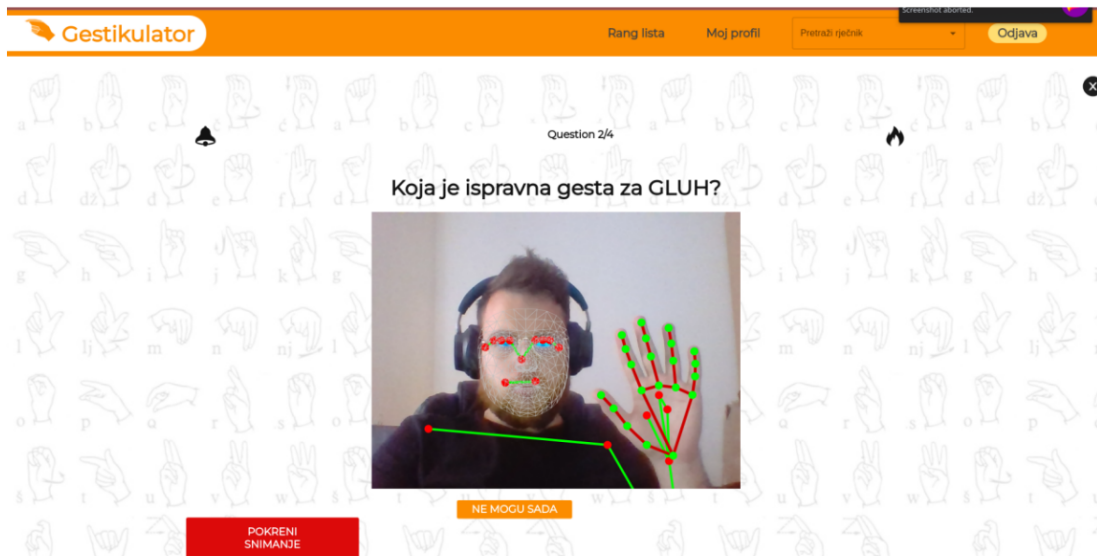


Slika 5.9: Pitanje tekst - odgovor video.

5.4.3. Pitanje tekst - odgovor kamera

Možda i najbitniji dio aplikacije, dio gdje korisnik na temelju danog naziva geste mora preko svoje kamere izvesti gestu na temelju čega mu model dubokog učenja validira

ispravnost izvođenja geste kako bi korisnik dobio povratnu informaciju izvodi li gestu ispravno. Korisnik mora sačekati kratko da se učitaju karakteristične točke preko JavaScriptove MediaPipe biblioteke, nakon čega pritiskom na gumb *POKRENI SNIMANJE* mora izvesti traženu gestu. Sustav nakon 30 - 60 okvira (engl. Frames) obuhvati sve karakteristične točke u tom intervalu te ih pošalje na ranije opisanu Python Flask aplikaciju gdje se vrši validacija te kao odgovor vraća povratna informacija je li gesta ispravno ili ne izvedena.

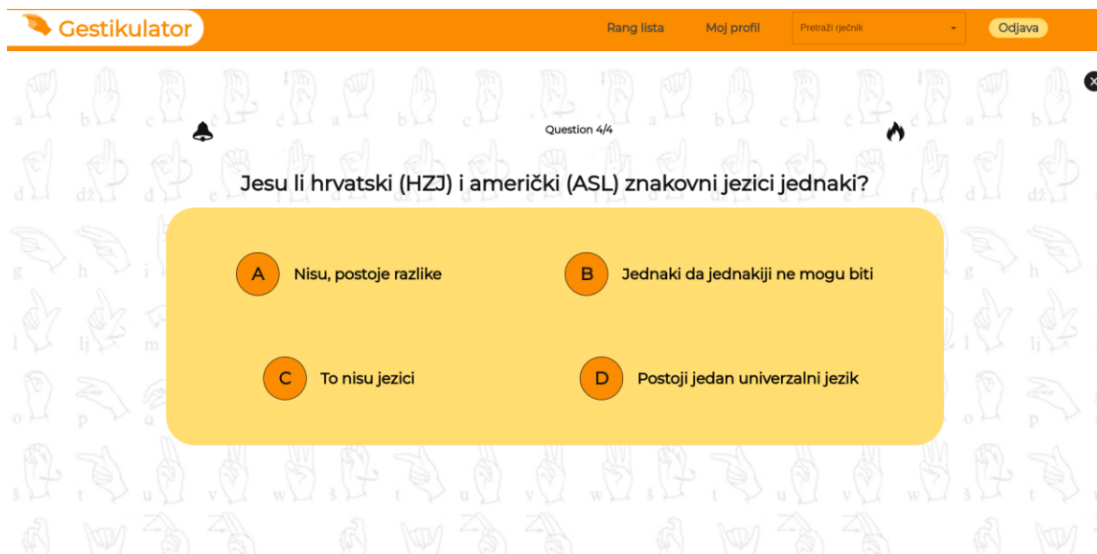


Slika 5.10: Pitanje tekst - odgovor kamera.

5.4.4. Pitanje o gluhoj i nagluhoj kulturi

Četvrta vrsta pitanja je zapravo kombinacija svih prethodnih, ali ona ne zahtjeva da korisnik odabere pravilno izvedenu gestu ili naziv, nego je više temeljena na kulturi i povijesti gluhe i nagluhe zajednice. Ovisno o pitanju korisnik će ili morati upaliti kameru i izvesti gestu, ili odabrati ispravan video koji odgovara na pitanje ili odbrati tekst koji ispravno odgovara na pitanje. Tako na primjer se tu mogu nalaziti pitanja:

- jesu li znakovni jezici univerzalni?
- koje je godine služben hrvatski znakovni jezik?
- kako je nastala gesta "petak"?
- kako se plješće u gluhoj i nagluhoj zajednici?



Slika 5.11: Pitanje o gluhoj i nagluhoj kulturi.

6. Upute za pokretanje

6.1. Pokretanje Python djela

Prije pokretanja je potrebno pravilno konfigurirati *config.ini* i *models_config.json* datoteke. Python model se može testirati zasebno, a neke od najbitnijih dijelova su:

- *add_gesture.py* - kreiranje podatkovnog skupa, gdje se preko korisnikove kamere snima korisnika kako izvodi gestu i spremaju karakteristične točke
- *train_model.py* - treniranje modela na temelju gesti spremljenih pomoću *add_gesture.py* programa
- *test_model.py* - program koji uzima iztreniran model u .h5 formatu te pomoću kamere korisnik može izvoditi geste na temelju čega će sustav validirati ispravnost izvođenja
- *api/server.py* - program koji na temelju *models_config.json* datoteke uzima određene modele za određene geste te pokreće program koji se vrti na adresi *http://localhost:5000* gdje se na endpointu */predict* mogu poslati naziv geste i karakteristične točke te sustav vrati je li gesta ispravno izvedena.

6.2. Pokretanje baze podataka i backenda

Prvo je potrebno pokrenuti bazu podataka, koja je u našem slučaju dockerizirana instanca PostgreSQL baze podataka. Nakon što je baza pokrenuta potrebno je pokrenuti Liquidbase skripte koje će kreirati potrebne tablice i napuniti ih podacima. Nakon toga pokretanjem Java Spring Boot aplikacije se pokreće backend koji će se spojiti na bazu podataka i servirati podatke na prije opisanim krajnjim točkama (engl. Endpoint).

6.3. Pokretanje frontenda

Pozicioniranjem u glavni direktorij Frontenda te pokretanjem naredbi `npm install` instalirat će se potrebni paketi te pokretanjem `npm start` pokrenut će se aplikacija na lokalnoj adresi `http://localhost:3000`.

7. Zaključak

Detekcija poze u stvarnom vremenu otvara vrata raznim primjenama. Jedna od takvih primjena je učenje znakovnoga jezika gdje bi se mogao razviti sustav koji bi na temelju prikaza s kamere mogao u stvarnom vremenu raspoznavati o kojoj se gesti radi ili je li gesta ispravno izvedena. Vođeni tom idejom razvili su Gestikator - platformu za učenje hrvatskoga znakovnog jezika, ali ne na dosadan način gledanja videa, nego interaktivnije, kroz igru, odnosno rješavanje kvizova gdje bi se poticalo korisnika da uz minimalni napor od pet do petnaest minuta dnevno rješava različite tipove pitanja i tako nauči osnove HZJ-a. U pozadini sve toga se nalazi model dubokog učenja koji pomoću Googleove MediaPipe biblioteke te pomoću povratnih neuronskih mreža (RNN) određuje ispravnost izvođenja određene geste te daje korisniku povratnu informaciju. U sklopu platforme postoji i rječnik hrvatskoga znakovnog jezika kao i aspekt društvene mreže, gdje se korisnici mogu međusobno dodati kao prijatelji te pratiti međusobni napredak i čak natjecati.

U radu su opisani postupci razvoja aplikacije, od backend i frontend dijela, kao i biblioteka MediaPipe te razvoj modela dubokog učenja. Smatramo da je ovako jedna aplikacija prijeko potrebna i u podizanju svijesti o gluhoj i nagluhoj kulturi kao i u učenju osnova sporazumijevanja s gluhom ili nagluhom osobu radi boljitka društva te zbog popularnosti aplikacija za učenje jezike smatramo da ova aplikacija i primjena ima veliki potencijal.

LITERATURA

- [1] Tarczay, S. i suradnici(2004), Znak po znak 1, udžbenik za učenje hrvatskoga znakovnog jezika, Hrvatska udruga gluhoslijepih osoba DODIR, Zagreb
- [2] Juriša, M.(2012), Oštećenje sluha, gluhoća/naglušost i savjeti za prevladavanje komunikacijskih barijera, Zbornik radova sa stručnih skupova, Hrvatski savez gluhih i nagluhih, Zagreb, str. 9-14.
- [3] https://hr.wikipedia.org/wiki/Hrvatski_znakovni_jezik
- [4] <https://hsgn.hr/>
- [5] Bačić, Monika, Hrvatski znakovni jezik u osnovnim školama, <https://urn.nsk.hr/urn:nbn:hr:147:762113>
- [6] Izvješće o osobama s invaliditetom u Republici Hrvatskoj, Hrvatski zavod za javno zdravstvo - https://www.hzjz.hr/wp-content/uploads/2021/10/Invalid_2021.pdf
- [7] <https://www.semanticscholar.org/paper/Deep-Learning-Goodfellow-Bengio/a4cec122a08216fe8a3bc19b22e78fbaea096256>
- [8] Toshpulatov, M., Lee, W., Lee, S. et al. Human pose, hand and mesh estimation using deep learning: a survey. J Supercomput 78, 7616–7654 (2022). <https://doi.org/10.1007/s11227-021-04184-7>
- [9] Kevin Lin, Lijuan Wang, i Zicheng Liu. Mesh graphormer, 2021. URL <https://arxiv.org/abs/2104.00272>.
- [10] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, i Matthias Grundmann. Mediapipe: A framework for building perception pipelines, 2019. URL <https://arxiv.org/abs/1906.08172>.

- [11] <https://google.github.io/mediapipe/solutions/holistic.html>
- [12] Majetić, Klara: Uvod u hrvatski znakovni jezik 2018. URL <https://repozitorij.ffzg.unizg.hr/islandora/object/ffzg>
- [13] Dmitry Matveev. OpenCV Graph API. Intel Corporation, 2018.
- [14] TensorFlow. TensorFlow Lite, 2017. <https://www.tensorflow.org/lite>
- [15] https://www.fer.unizg.hr/predmet/dubuce_a
- [16] <https://www.fer.unizg.hr/predmet/struce1>
- [17] Hardesty, Larry (14 April 2017). "Explained: Neural networks". MIT News Office. Retrieved 2 June 2022.
- [18] Bengio, Yoshua; LeCun, Yann; Hinton, Geoffrey (2015). "Deep Learning". *Nature*. 521 (7553): 436–444. Bibcode:2015Natur.521..436L. doi:10.1038/nature14539. PMID 26017442
- [19] Dupond, Samuel (2019). "A thorough review on the current advance of neural network structures"
- [20] Muthu Mariappan H, & Dr Gomathi V. (2021). Indian Sign Language Recognition through Hybrid ConvNet-LSTM Networks. *EMITTER International Journal of Engineering Technology*, 9(1), 182-203. <https://doi.org/10.24003/emitter.v9i1.613>
- [21] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg and Matthias Grundmann: MediaPipe: A Framework for Building Perception Pipelines 2019. URL <https://arxiv.org/pdf/1906.08172.pdf>
- [22] Udruga osoba oštećena sluha Bjelovarsko-bilogorske županije URL <http://uoosbbz.hr/>
- [23] Klara Majetić, Petra Bago: Model e-rječnika hrvatskoga znakovnog jezika 2018. URL <https://hrcak.srce.hr/file/294726>
- [24] Tomažin Nikola: Sustav za raspoznavanje gesti hrvatskoga znakovnog jezika 2022.

Gestikulator - Platforma za učenje hrvatskoga znakovnog jezika

Sažetak

Gestikulator je platforma za učenje hrvatskoga znakovnog jezika, ali ne na dosadan način gledanja videa, nego interaktivnije, kroz igru, odnosno rješavanje kvizova gdje bi se poticalo korisnika da uz minimalni napor od pet do petnaest minuta dnevno rješava različite tipove pitanja i tako nauči osnove HZJ-a. U pozadini sve toga se nalazi model dubokog učenja koji pomoću Googleove MediaPipe biblioteke te pomoću povratnih neuronskih mreža (RNN) određuje ispravnost izvođenja određene geste te daje korisniku povratnu informaciju. U sklopu platforme postoji i rječnik hrvatskoga znakovnog jezika kao i aspekt društvene mreže, gdje se korisnici mogu međusobno dodati kao prijatelji te pratiti međusobni napredak i čak natjecati.

Ključne riječi: hrvatski znakovni jezik, učenje, računalni vid, MediaPipe, gesta, web aplikacija.

Gestikulator - A learning platform for Croatian Sign Language

Abstract

Gestikulator is a platform for learning Croatian sign language, but not in a boring way of watching videos, but more interactively, through play, ie solving quizzes where the user would be encouraged to ,with a minimum effort of five up to fifteen minutes a day, to solve different types of questions and thus learn the basics of Croatian Sign Language. In the background of all this is a deep learning model that uses Google's MediaPipe library and recurrent neural networks (RNN) to determine the correctness of performing a particular gesture and give the user feedback. Within the platform, there is also a dictionary of Croatian Sign Language as well as an aspect of a social network, where users can add to each other as friends and monitor each other's progress and even compete.

Keywords: Croatian Sign Language, Learning, Computer Vision, MediaPipe, Gesture, Web Application.