Mapiranje zvučnih informacija s pokretima lica afektivnog virtualnog agenta

Gregov, Petra

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Petra Gregov

Zagreb, 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:
Izv. prof. dr.sc. Tomislav Stipančić, dipl. ing.

Student:
Petra Gregov

Zagreb, 2023.

Izjavljujem da sam ovaj rad izradila samostalno koristeći stečena znanja tijekom studija i navedenu literaturu.

Petra Gregov

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
Proizvodno inženjerstvo, inženjerstvo materijala, industrijsko inženjerstvo i menadžment, mehatronika i robotika, autonomni sustavi i računalna inteligencija

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: **602 - 04 / 23 - 6 / 1**	
Ur.broj: **15 - 23 -**	

DIPLOMSKI ZADATAK

Student: **Petra Gregov** JMBAG: 0035215291

Naslov rada na hrvatskom jeziku: **Mapiranje zvučnih informacija s pokretima lica afektivnog virtualnog agenta**

Naslov rada na engleskom jeziku: **Mapping sound information with facial movements of an affective virtual agent**

Opis zadatka:

Strojno i duboko učenje kao grane umjetne inteligencije fokusiraju se na razvoj računalnih modela koji su sposobni automatski otkriti značajne obrasce iz analiziranih podataka. U području virtualne stvarnosti i afektivne robotike često je fokus istraživanja razvoj virtualnih modela ljudi. Na virtualne modele se potom implementiraju različiti obrasci ponašanja i djelovanja ljudi kako bi se ostvarilo njihovo što vjernije djelovanje.
Afektivni virtualni agent PLEA se razvija u sklopu Laboratorija za projektiranje izradbenih i montažnih sustava te se koristi za proučavanje interakcije ljudi i robota. PLEA omogućuje analizu i procjenu emocionalnog stanja osobe za vrijeme interakcije što potom koristi kako bi vodila interakciju.
U radu je potrebno uskladiti pokrete usta virtualnog agenta sa zvučnim informacijama koje PLEA koristi prilikom verbalne komunikacije.
U sklopu toga je potrebno:

- istražiti koje su metode strojnog učenja najbolji izbor za klasifikaciju zvučnih podataka, odnosno analizu govora,
- odrediti prikladnu razvojnu okolinu za povezivanje s 3D modelom lica virtualnog bića,
- prikupiti vizualne podatke osoba koje govore te stvoriti bazu podataka za treniranje računalnog modela ili koristiti postojeću, te
- izraditi računalni model koji će povezati zvučne ulazne podatke s odgovarajućim izražajima lica virtualnog bića u realnom vremenu.

Rad računalnog modela je potrebno evaluirati koristeći testni i neovisni skup podataka kako bi se pronašli odgovarajući parametri učenja, minimizirale greške predikcije te ubrzao rad modela.
U radu je potrebno navesti korištenu literaturu te eventualno dobivenu pomoć.

Zadatak zadan:	Datum predaje rada:	Predviđeni datumi obrane:
28. rujna 2023.	30. studenoga 2023.	4. – 8 . prosinca 2023.

Zadatak zadao: Predsjednik Povjerenstva:

Izv. prof. dr. sc. Tomislav Stipančić Prof. dr. sc. Ivica Garašić

Sadržaj

Popis slika

Popis tablica

Popis kratica

Kratica	Značenje
A2F	Audio2Face
AI	*Artificial Intelligence*
ANN	*Artificial Neural Network*
ASR	*Automatic Speech Recognition*
AU	*Action Unit*
bps	bitova po sekundi
CNN	*Convolutional Neural Networks*
DCT	*Discrete Cosine Transform*
DFT	*Discrete Fourier Transform*
DL	*Deep Learning*
DNN	*Deep Neural Network*
DSP	*Digital Signal Processing*
FACS	*Facial Action Coding System*
FC	*fully connected*
FFT	*Fast Fourier transform*
FNN	*Feedforward Neural Network*
FT	*Fourier Transform*
HCI	*Human-Computer Interaction*
LPC	*Linear Predictive Coding*
LSTM	*Long Short-Term Memory*
MDL	*Multimodal Deep Learning*
MFCCs	*Mel-Frequency Cepstral Coefficients*
MH	*MetaHuman*
MLP	*Multilayer Perceptron*
MMSE	*Minimum Mean-Square Error*

NLP	*Natural Language Processing*
RNN	*Recurrent Neural Networks*
SPL	*Sound Pressure Level*
STFT	*Short Time Fourier Transform*
UE	*Unreal Engine*

Sažetak

Tema ovog rada je generiranje lica vođeno zvukom u stvarnom vremenu kroz primjenu dubokog učenja. Rad se u teorijskom dijelu fokusira na pravilnu obradu ulaznih podataka za trening modela i analizu najčešće korištenih neuronskih mreža. Dodatno se razmatraju različita rješenja zadatka u sklopu aktualnih istraživanja. U praktičnom dijelu rada se istražuje implementacija gotovog modela putem Audio2Face aplikacije. Rad aplikacije se dodatno obrađuje u sklopu virtualnog okruženja unutar Unreal Engine platforme s naglaskom na prijenos uživo.

Ključne riječi: duboko učenje, obrada zvučnog signala, točke lica, generiranje lica vođeno zvukom, Audio2Face, Unreal Engine

Summary

The theme of this paper is real-time sound-driven face generation utilizing deep learning. In the theoretical part, the emphasis is on proper processing of input data for model training and analysis of the most commonly used neural networks. Furthermore, diverse approaches to the task are investigated within the context of ongoing research. In the practical part, exploring the implementation of a pre-trained model through the Audio2Face application takes focus. The application's functionality is further analyzed within a virtual environment on the Unreal Engine platform, with a focus on live streaming.

Key words: deep learning, audio signal processing, facial landmarks, sound-driven face generation, Audio2Face, Unreal Engine.

1. Uvod

U suvremenom računalnom okruženju, integracija dubokog učenja postaje neizostavan dio razvoja interaktivnih aplikacija i tehnologija, posebice kada je riječ o kompleksnijim zadacima. U ovom radu se nastoji prodrijeti u osnovne principe i izazove generiranja animacije lica vođene zvukom, prepoznajući potencijal takvih sustava u transformaciji područja interakcije čovjeka i računala. Generiranje animacije lica vođeno zvukom otvara vrata novim dimenzijama korisničkog iskustva, razvoja video igara i filmske animacije, te raznim poboljšanjima u edukaciji i medicini.

Prvi dio rada usmjerava se na definiranje osnovnih pojmova te pravilnu pripremu i obradu ulaznih podataka. Obrada podataka obuhvaća pravilne reprezentacije zvuka te različite tehnike mapiranja ključnih anatomsko-fizioloških točaka lica. Detaljno istraživanje ovog segmenta osigurava kvalitetan temelj za daljnje razmatranje specifičnosti u implementaciji dubokog učenja.

Drugi dio rada fokusira se na primjenu dubokog učenja te se razmatraju najčešće korištene neuronske mreže. Moguće implementacije i rješenja prethodno opisanih metoda se pružaju kroz pregled relevantnih istraživanja. Time se daje uvid u aktualna postignuća u tom području i ističe se raznolikost pristupa u generiranju animacije pomoću zvuka.

Treći dio rada posvećen je praktičnoj implementaciji prethodno opisanih metoda preko Audio2Face aplikacije koja predstavlja inovativan pristup korištenju dubokog učenja za stvaranje dinamične i autentične animacije lica. Analiziraju se sučelje i mehanizmi pokretanja animacije te se naposljetku opisuje prijenos animacije u Unreal Engine u stvarnom vremenu.

2. Umjetne neuronske mreže

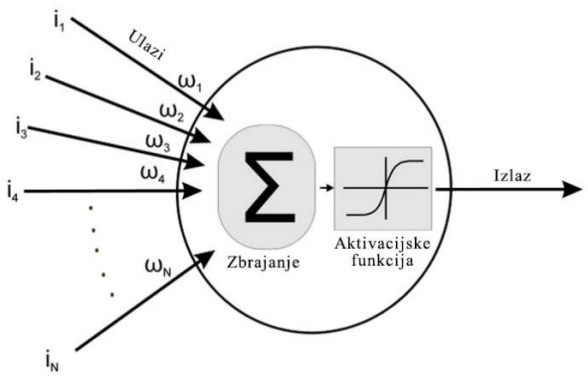
Umjetna inteligencija (eng. *Artificial Intelligence*, krat. AI) je znanost kojom se razvijaju neživi sustavi sposobni za snalaženje u nenaučenim situacijama s ciljem imitiranja ljudske inteligencije. No, definiranje umjetne inteligencije se pokazalo dvojakim problemom. S jedne strane, pojam inteligencije se ne može jednoznačno objasniti, te postoji velik broj približno sličnih definicija inteligencije od kojih nijedna ne obuhvaća njen svaki aspekt. S druge strane, ne postoji univerzalna metoda umjetne inteligencije primjenjiva za svaki problem, već metoda ovisi o vrsti problemi dok se neki problemi mogu riješiti s više različitih metoda. Navedena problematika je rezultirala grananjem umjetne inteligencije i nedostatkom točno utvrđenih metoda i formalizama [1].

Današnju tehnologiju karakterizira velika količina podataka iz velikog broja izvora te nastaje pojam „Big dana“ koji ujedinjuje količinu podataka, brzinu kojom ta količina raste i raznolikost samih podataka. Rukovanje tako opsežnim i raznolikim skupovima podataka je postao nedostižan zadatak za čovjeka te nastaje potreba za automatizaciju ekstrakcije bitnih podataka i pripadajućih atributa. Naime, razlog zašto je čovjek iznimno dobar u obradi podataka je adaptabilnost, odnosno sposobnost donošenja odluka s obzirom na velik broj promjenjivih faktora kroz proces učenja. Grana umjetne inteligencije koja se bavi razvojem algoritama koji su sposobni „učiti“ je strojno učenje (eng. *machine learning*). Učenje u tom kontekstu podrazumijeva prepoznavanje bitnih obrazaca iz podataka te primjena naučenog na novi set podataka [2].

Sredinom 20. stoljeća nastaju umjetne neuronske mreže (eng. *Artificial Neural Network*, krat. ANN), algoritmi čija se struktura temelji na radu neurona. Iako se jako malo zna o tome kako mozak funkcionira, neka osnovna pravila se mogu primijeniti na razvoj takvih algoritama. Primjerice, prema Hebbianovoj teoriji sinaptička povezanost neurona u mozgu je varijabilna te ovisi o aktivnosti

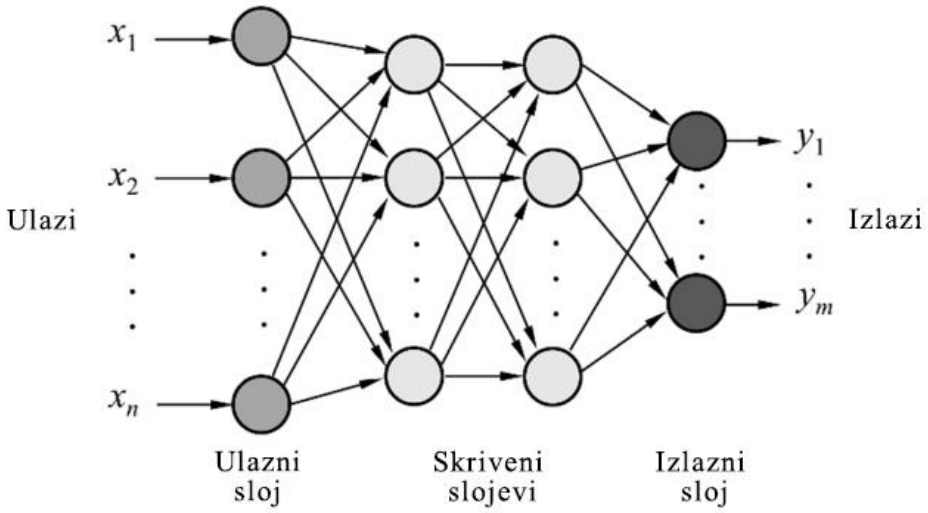
neurona. Varijabilnost snage povezanosti neurona je jedna od osnova učenja. Odnosno, iako su svi neuroni aktivni, određeni neuroni prenose bitnije informacije s obzirom na namjenu neuronske mreže. Na temelju toga, ANN između svojih neurona postavlja težine (eng. *weights*) koje određuju snagu povezanosti dvaju neurona te se mijenjaju dok se mreža uči, odnosno trenira. No, iako ANN postoje od prošlog stoljeća, njihova primjena je bila ograničena zbog manjka podataka na kojima može učiti i zbog manjka računalnih resursa. Razdoblje *Big data* omogućuje algoritmima bolju generalizaciju, odnosno kvalitetnije učenje, a time i širu primjenu. Dok, s druge strane, razvojem tehnologije i dostupnosti iste omogućuje rad s puno kompleksnijim i većim modelima [3].

Neuron je sastavna jedinica neuronskih mreža koje prenose informacije o atributima podataka. Na Slika 1 je prikazana osnovni način rada neurona. Svaki neuron je povezan sa svakim neuronom. Informacije iz prethodnih neurona u obliku vrijednosti se množe sa pripadajućim težinama ω_i koje su drukčije za svaku vezu. Tako pomnožene vrijednosti se zbrajaju te se na sumu primjenjuje aktivacijska funkcija kojom se uvodi linearnost u mrežu. Na taj način se smanjuje broj parametara iz kojih mreža uči i postiže se rjeđa povezanost mreže. Najčešća funkcija aktivacije je ReLU funkcija koja mijenja negativne vrijednosti u nulu [4].



Slika 1 Osnovni princip rada neurona ANN [4]

Najčešća arhitektura ANN je vidljiva na Slika 2 gdje su neuroni poslagani u slojevima. Sve operacije u jednom sloju se događaju simultano. Naime, ulazni i izlazni sloj su „vidljivi“ slojevi jer su poznati, dok su unutarnji slojevi „skriveni“ jer promjene koje se događaju unutar njih se ne mogu izravno predvidjeti promatranjem ulaznog i izlaznog sloja. Naime, transformacije i komputacije koje se događaju unutar skrivenih slojeva su inherentne mreži. Ovakav oblik ANN u kojem tok informacije ide u jednom smjeru, od ulaza prema izlazu, se nazivaju unaprijednim neuronskim mrežama (eng. *Feedforward Neural Network*, krat. FNN). Stariji naziv za FNN koji se može pojaviti u literaturi je višeslojni perceptron (eng. *Multilayer Perceptron,* krat. MLP). Uvođenjem više od jednog skrivenog sloja mreža postaje „duboka“ te spada u duboke neuronske mreže (eng. *Deep Neural Network*, krat. DNN). Područje strojnog učenja koje se bavi s DNN je duboko učenje (eng. *Deep Learning*, krat. DL) [4].



Slika 2 Osnovna FNN arhitektura [4]

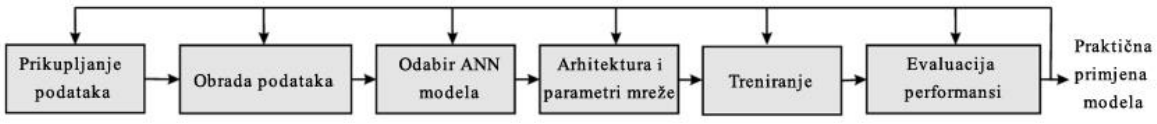
U sklopu ovog rada će se obraditi zadatak generiranja točaka lica vođenim zvukom u stvarnom vremenu, odnosno osnovna ideja je dobiti niz mapa koordinata točaka lica pogodnih za animaciju 3D modela iz zvuka. Ulazi DL

modela koji uči predviđati opisani zadatak su zvuk i niz mapa točaka lica dobivenih iz pripadajućeg videa. Mape točaka lica su željeni izlaz te predstavljaju označenu vrstu podataka za treniranje mreže. Vrsta učenja mreže koja se bazira na označenim podacima se zove nadzirano učenje, a najčešća tehnika nadziranog učenja je algoritam propagacije unatrag (eng. *backpropagation algorithm*). Prvo mreža predviđa neki izlaz koji je produkt svih neurona i svih težina. Zatim se računa greška između predviđenih vrijednosti i označenih podataka za trening. Greška se računa s funkcijama specifičnim zadatku. Nakon toga algoritam putuje unatrag, od izlaza prema ulazu mreže, i mijenja težine koristeći gradijentni spust. Gradijent je smjer rasta neke funkcije te se u kontekstu neuronskih mreža se računa kao derivacija greške s obzirom na težine. Gradijentni spust je optimizacijski algoritam kojim se traži minimum funkcije, odnosno težine se mijenjaju u suprotnom smjeru od smjera gradijenta prema izrazu:

$$\boldsymbol{W}_x^* = \boldsymbol{W}_x - a\left(\frac{\partial Greška}{\partial \boldsymbol{W}_x}\right), \tag{2.1}$$

gdje su $\boldsymbol{W}_x$ matrica težina između dva sloja, $\boldsymbol{W}_x^*$ promijenjena matrica težina i a stopa učenja. Postupak se iterira dok mreža ne zadovolji neki prag greške. Nakon toga se trenirani model testira na testnom skupu podataka čime se evaluira performansa modela [3].

Tipičan proces nastanka ANN modela je prikazan na Slika 3.



Slika 3 Proces izrade ANN modela [4]

Jedan od bitnijih koraka pri izradi kvalitetnog modela je pravilna priprema i obrada podataka na kojima se mreža trenira. Naime, previše nepotrebnih atributa unutar ulaznih podataka smanjuje preciznost mreže i zahtjeva veću kompleksnost modela. Stoga će se u ovom radu, za početak, razraditi pravilna reprezentacija ulaznih podataka u kontekstu generiranja lica vođeno zvukom.

3. Analiza govora

U posljednja dva desetljeća metode digitalne obrade signala (eng. *Digital Signal Processing*, krat. DSP) i metode dubokog učenja doživljavaju nagli napredak. Povećana zainteresiranost i njihovo bolje razumijevanje utječu na discipline umjetne inteligencije vezane za govor i jezik. Metode automatskog prepoznavanja govora[1] (eng. *Automatic Speech Recognition*, krat. ASR) i obrade prirodnog jezika[2] (eng. *Natural Language Processing*, krat. NLP) se sve češće integriraju u zadatke umjetne inteligencije, a veći fokus je doveo do velikog broja istraživanja koja će omogućiti kompleksniju primjenu navedenih metoda [5], [6].

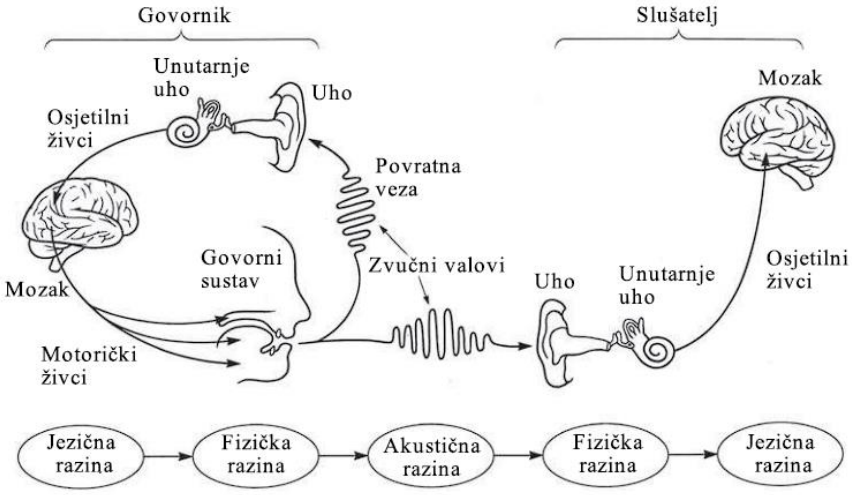
U ovom poglavlju će se detaljnije obraditi proces obrade zvučnog signala koji sadrži govor te će se opisati narav zvučnih podataka kao ulaznih podataka za daljnju uporabu.

3.1 Zvučni signal govora

Zvuk putuje kroz zrak u obliku akustičnih zvučnih valova koji uzrokuju titranje čestica, odnosno mehaničke vibracije. Mehaničke vibracije se zatim pomoću specifičnog organa u unutarnjem uhu pretvaraju u električni impuls koji mozak interpretira kao zvuk (Slika 4). Na sličan način funkcioniraju i mobilni uređaji koji zvučne valove pretvaraju u električne signale koji sadrže informacije o zvuku. Električni signal se može manipulirati, nakon čega se ponovo vraća u akustičnu formu, odnosno u zvučne valove [6].

[1] Sustavi koji omogućuju prepoznavanje i interpretaciju govora (npr. zapis govora kao teksta): *https://www.britannica.com/technology/speech-recognition*

[2] Sustavi koji omogućavaju razumijevanje govora i konteksta oponašajući ljudsku sposobnost: *https://www.britannica.com/technology/natural-language-processing-computer-science*

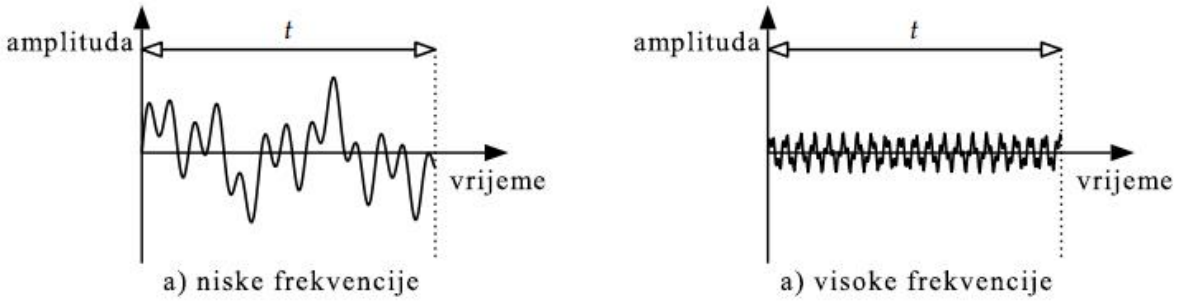


Slika 4 Proces stvaranja i percepcije zvuka [6]

Prilikom analogno digitalne pretvorbe akustičnog vala zvuka, kvaliteta očuvanja zvuka ovisi o brzini prijenosa informacija. Prema teoriji informacija, zvuk se može opisati kroz informacije koje prenosi i brzinom kojom ih prenosi, gdje su informacije podaci koji opisuju zvuk. Govor se na najjednostavnijoj razini može podijeliti na diskretne elemente sastavljene od određenog broja simbola koji predstavljaju najmanje jedinice jezika – foneme. Broj i vrsta fonema ovisi o jeziku, primjerice hrvatski jezik sačinjava 32 fonema od kojih je 30 poznato, a preostala dva su slogotvorno r i glas „ie“.. Brzina prijenosa informacija ovisi o: brzini govora, broju i vrsti fonema, prozodiji izgovora, artikulaciji te o kontrolnim signalima koji će definirati artikulaciju. Naposljetku, normalna brzina prijenosa se kreće između 64 tisuće i 700 tisuća bitova po sekundi (krat. bps). Veća brzina prijenosa podrazumijeva veće očuvanje kvalitete zvuka, primjerice mobilna kvaliteta zvuka je 64,000 bps dok je brzina „CD kvalitete“ 705,600 bps. U slučaju mobilne kvalitete, može se prepoznati primjetna razlika između prenesenog i originalnog zvuka, dok je za CD kvalitetu ta razlika za čovjeka skoro nepostojeća. Prilikom pripreme i obrade zvučnog signala za uporabu, potrebno je očuvati poruku koja se prenosi i maksimalno zadržati kvalitetu zvuka [6].

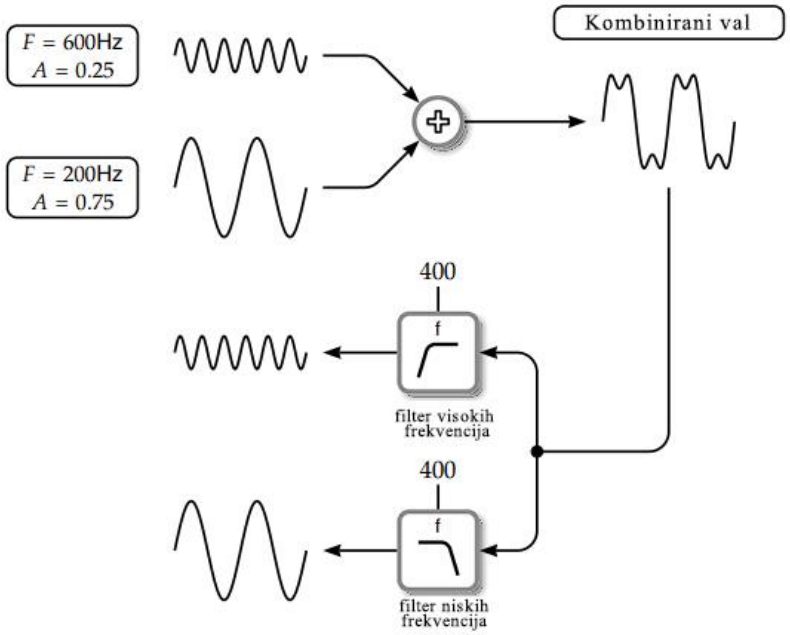
3.2 Priprema zvučnog signala

Zvučni val je određen svojim frekvencijama i amplitudama. Frekvencija označava broj oscilacija u sekundi gdje se veća frekvencija povezuje s višim tonovima, dok niža frekvencija s nižim (Slika 5). Amplituda u kontekstu zvučnog vala označava varijaciju tlaka zraka za koju se nelinearno veže razina jakosti zvuka (eng. *Sound Pressure Level*, krat. SPL). Jakost zvuka je mjera za intenzitet zvuka te linearno raste s eksponencijalnim povećanjem tlaka zraka. Bitno je napomenuti da razina jakosti zvuka nije isto što i glasnoća jer je glasnoća složena karakteristika zvuka koja ovisi o više faktora, ali su usko povezani [7].



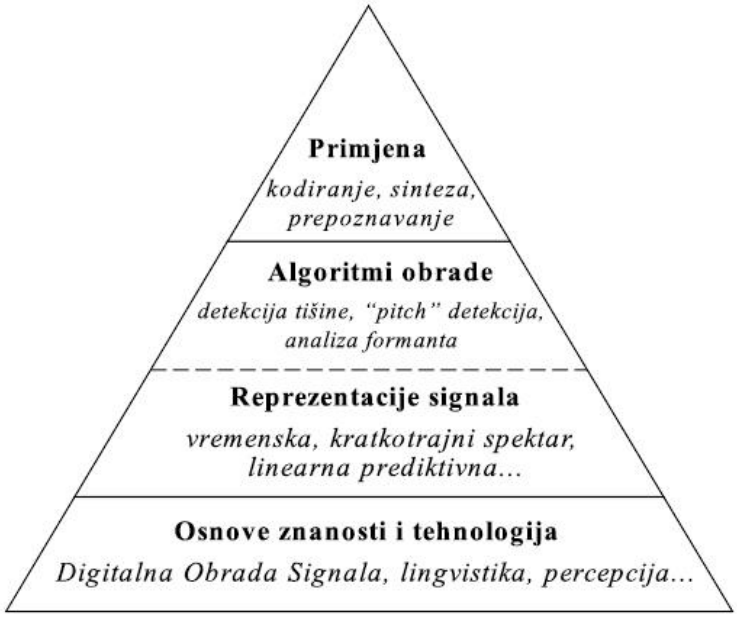
Slika 5 Izgled zvučnih signala niskih i visokih frekvencija [7]

Zvuk je kompleksni kontinuirani zvučni val u vremenskoj domeni koji se može opisati s više sinusoida gdje svaka ima pripadajuću frekvenciju i amplitude [7]. Na Slika 6 je vidljivo kako zvučni val može nastati sumiranjem dvaju jednostavnih zvučnih valova te se jednako tako može razdvojiti na više valova filtriranjem.



Slika 6 Konstrukcija i dekonstrukcija zvučnog vala pomoću sinusoida [7]

Kako bi zvuk bio uporabljiv za strojno učenje, potrebno ga je transformirati u oblik koji je pogodan za vrstu zadatka. Na Slika 7 prikazan je proces računalne uporabe zvuka od teorijskih temelja do primjene.



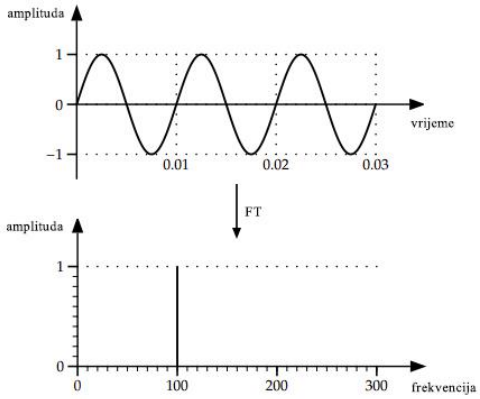
Slika 7 Računalna uporaba zvučnog signala od teorije do primjene [6]

Za početak je bitno prikupiti znanje o zvuku i njegovoj digitalnog pretvorbi i obradi. Nakon toga se zvučni val transformira u oblik pogodan za daljnju analizu, odnosno potrebna je njegova odgovarajuća reprezentacija. Algoritmima obrade se zatim mogu izdvojiti bitni atributi ili ukloniti nebitni. Granica između drugog i trećeg sloja nije stroga iz razloga što algoritmi obrade ovise o kvaliteti reprezentaciji signala koja se može po potrebi izmijeniti. Naposljetku se obrađena reprezentacija zvuka može primijeniti za različite zadatke [6].

3.2.1 Osnovni koncepti reprezentacije zvuka

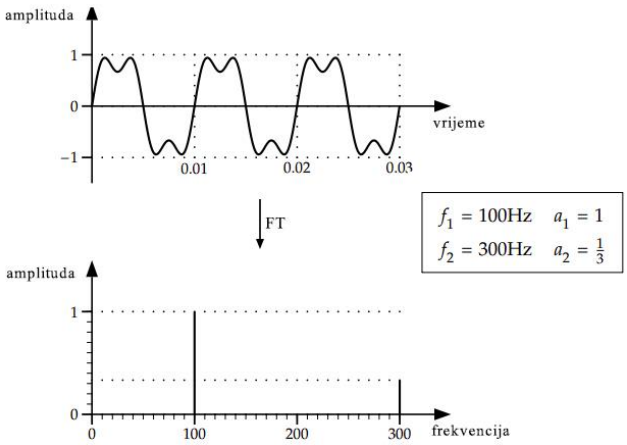
Mnogi dijelovi zvuka su nepotrebni prilikom analize govora i otežavaju iščitavanje bitnih atributa. Različite transformacije zvuka stavljaju naglasak na različite atribute govora te se odabir transformacije vrši prema vrsti zadatka. Prije transformacije zvuka, zvuk je potrebno uzorkovati (eng. *sampling*) na frekvenciju koja će biti dovoljna da očuva bitne informacije o zvuku. Po Nyquistovom teoremu, frekvencija uzorkovanja mora biti barem dvostruko veća od najviše frekvencije signala kako bi se osigurao kvalitetni prijenos signala [7].

Zvuk se u svojoj najosnovnijoj formi može prikazati frekvencijskim spektrom pomoću Fourierove transformacije (eng. *Fourier Transform*, krat. FT). FT je matematički postupak koji mjeri sličnost između originalne funkcije vremena i kompleksne harmonijske funkcije. Odnosno, zvučni signal u vremenskoj domeni opisan s promjenom amplitude kroz vrijeme se transformira u frekvencijsku domenu opisan s amplitudom za određenu frekvenciju.



Slika 8 FT jednostavnog signala [7]

Na Slika 8, signal je jednostavna sinusoidna funkcija. U slučaju složenije funkcije, koja je rezultat sumiranja više različitih funkcija, amplituda nije jednako uočljiva. Na Slika 9 prikazana je FT složene funkcije sastavljene od dvije sinusoide različitih frekvencija i amplituda.



Slika 9 FT složenog signala [7]

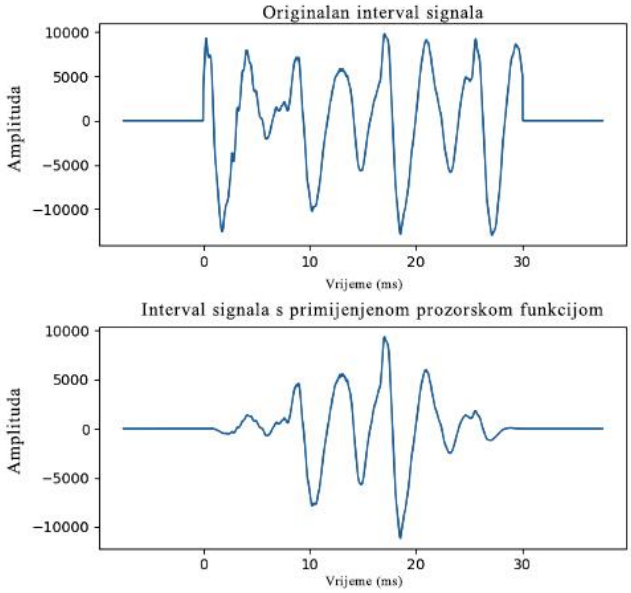
FT se računa prema:

$$X_f = \int_{-\infty}^{\infty} \mathrm{x}_t e^{-i2\pi f t}\, dt, \tag{3.1}$$

iz čega je vidljivo da se primjenjuje na beskonačne kontinuirane signale. S obzirom da je zvuk najčešće konačan signal s određenim vremenskim intervalom, može se koristiti diskretna FT (eng. *Discrete Fourier Transform*, krat. DFT) koja dijeli signal na N diskretnih intervala, odnosno prozora (eng. *window*) prema sljedećem izrazu:

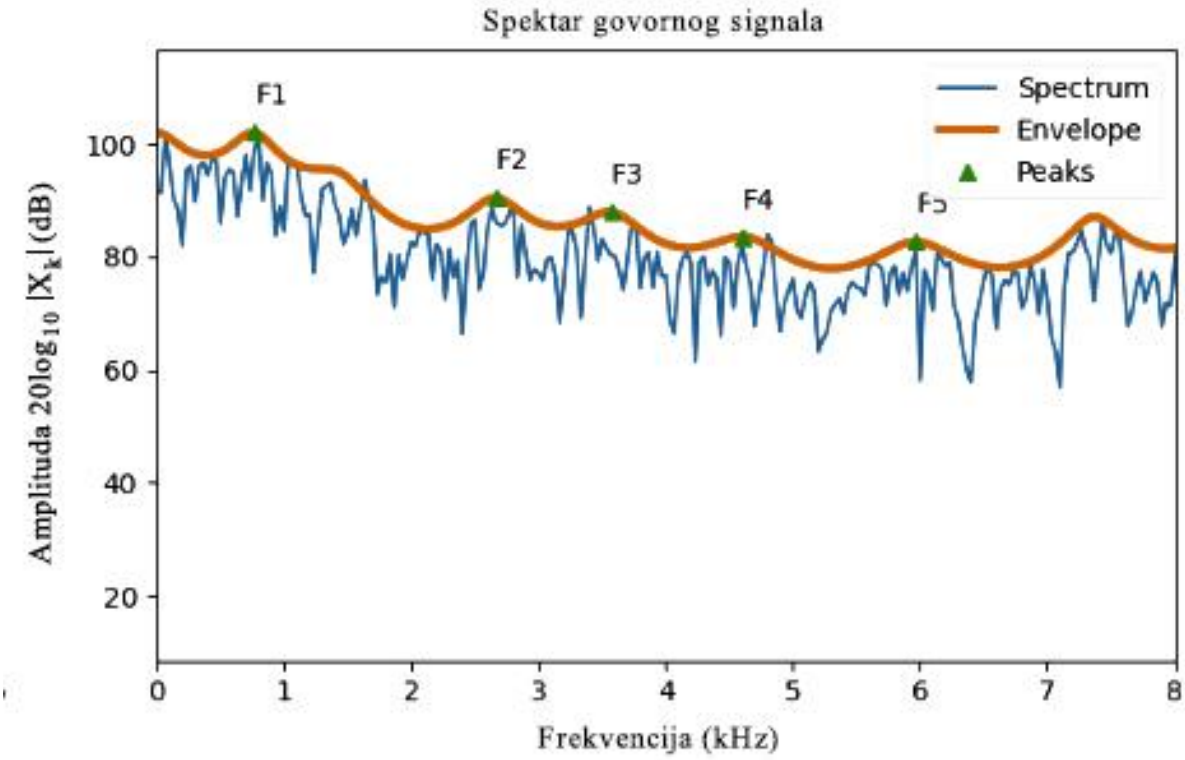
$$\tilde{X}_k = \sum_{n=0}^{N-1} \tilde{x}_n e^{-i2\pi kn/N}. \tag{3.2}$$

Jedan prozor signala kao mali dio ukupnog signala naizgled ima stacionarne karakteristike. S obzirom da se DFT funkcija primjenjuje na približno stacionaran dio zvuka, a ukupni zvuk nije stacionaran, pojavljuje se problem na krajevima signala gdje zvuk naglo počinje, odnosno naglo završava. Na taj način kraj transformiranog prozora se neće podudarati s početkom sljedećeg transformiranog prozora, te se takvi frekvencijski diskontinuiteti nazivaju razmazivanjem spektra (eng. *spectral leakage*). Frekvencije razmazivanja spektra se ne pojavljuju u originalnom zvuku te stvaraju problem prilikom analize i konstrukcije zvučnog signala [6]. Jedno od rješenja tog problema je prozorska funkcija (eng. *windowing*) koja na krajevima zaglađuje signal prema nuli unutar jednog prozora (Slika 10) [8].



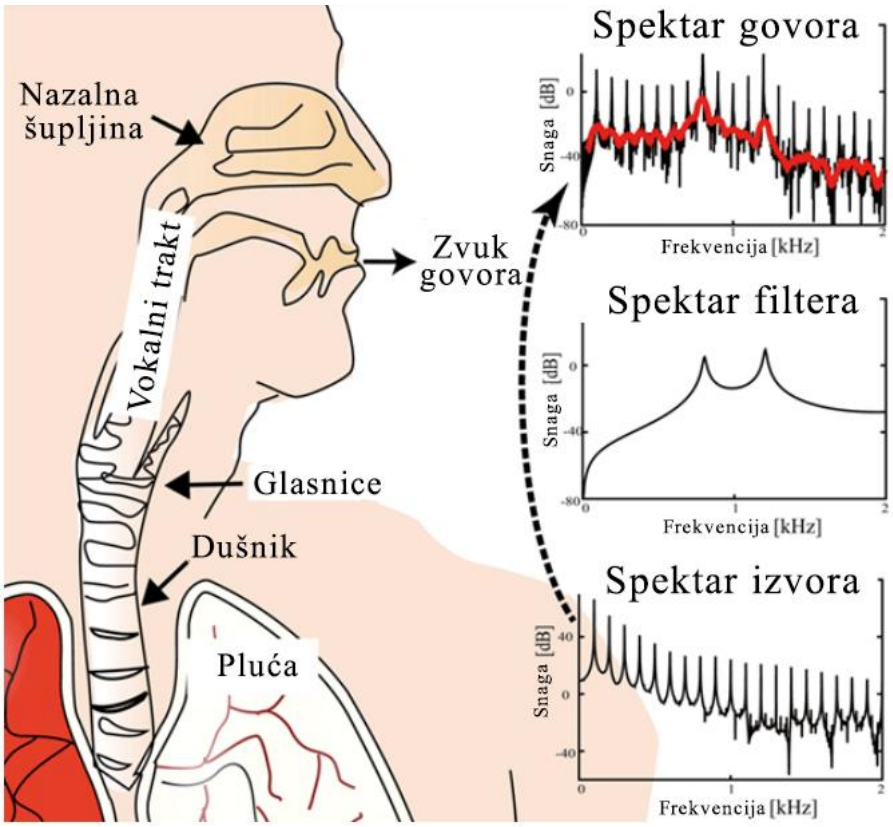
Slika 10 Primjena prozorske funkcije [8]

Izlaz DFT metode primijenjene na signal zvuka su DFT koeficijenti u obliku kompleksnih brojeva. Njihova apsolutna vrijednost opisuje amplitudu, a kvadratna vrijednost opisuje snagu, odnosno energiju signala. Primjenom logaritamske skale, podaci daju puno veći uvid u atribute signala te približno odgovaraju načinu na koji ljudi percipiraju glasnoću zvuka za pojedine frekvencije. Makro element spektra koji prati vrhove logaritamskih amplitudnih vrijednosti se naziva spektralni omotač. Vrhovi omotača se nazivaju formantima. Formanti su visoko-energetska područja spektra koja se najčešće vežu za samoglasnike, a samoglasnici povezuju suglasnike [8]. Iako svaki fonem zasebno ima svoju artikulaciju, uzastopni fonemi mogu imati drukčiju artikulaciju s obzirom na njihov redoslijed. Prepoznavanjem formanta i njihovog redoslijeda, mogu se odrediti ostale karakteristike zvuka koje se vežu za njih i pripadajući redoslijed [9]. Na Slika 11 je prikazan spektar govora sa spektralnim omotačima i označenim pozicijama formanta. Primjena DFT na sekvencu se još naziva i brza FT (eng. *Fast Fourier transform*, krat. FFT).



Slika 11 Prikaz spektralnog omotača i označeni formanti [8]

Nadalje, sredinom 20. stoljeća nastaje izvor-filter model govora. Naime prema tom modelu se mehanizam proizvodnje govora može podijeliti na izvor zvuka i na filter zvuka. Izvor zvuka bi bio protok zraka iz pluća koji titra glasnice. Izvorni zvuk tada prolazi prema vokalnom traktu koji sačinjavaju jezik, zubi, usne, ždrijelo i nepca. Vokalni trakt oblikuje spektralnu strukturu protoka zraka. Procesom filtracije pomoću vokalnog trakta se pojačavaju, odnosno smanjuju frekvencije. Izvor će određivati osnovnu harmoniju i frekvencije glasa, dok filter oblikuje ukupnu zvučnu strukturu. Na Slika 12 su vidljivi osnovni elementi u proizvodnji govora te kako bi izgledao spektar akustične snage i frekvencije za izvor, filter i govor [10].



Slika 12 Koncept teorije izvor-filter [10]

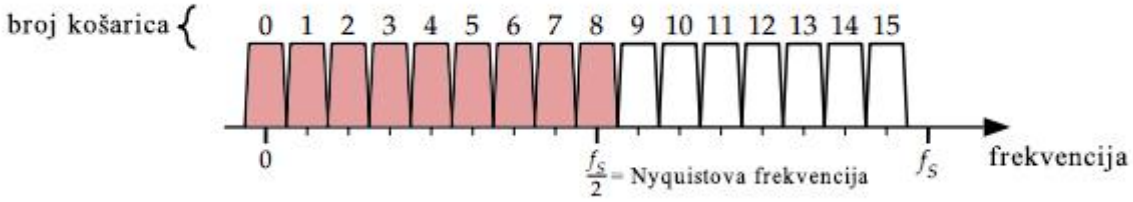
Opisani model je temelj mnogih metoda za ekstrakciju atributa iz govora koji se koriste u sustavima analize govora.

3.2.2 Reprezentacije signala govora

Zvučni signal je u svojoj prvotnoj formi redundantan stoga se teži učinkovitim reprezentacijama govornog signala. Odnosno, zvučni signal se oblikuje na efikasne načine koji zauzimaju manje resursa, ali i dalje omogućavaju kvalitetnu rekonstrukciju i analizu govora. U nastavku će se opisati najčešće korištene transformacije zvučnog signala u relevantnim domenama kako bi se dobilo dublje razumijevanje karakteristika zvučnog signala.

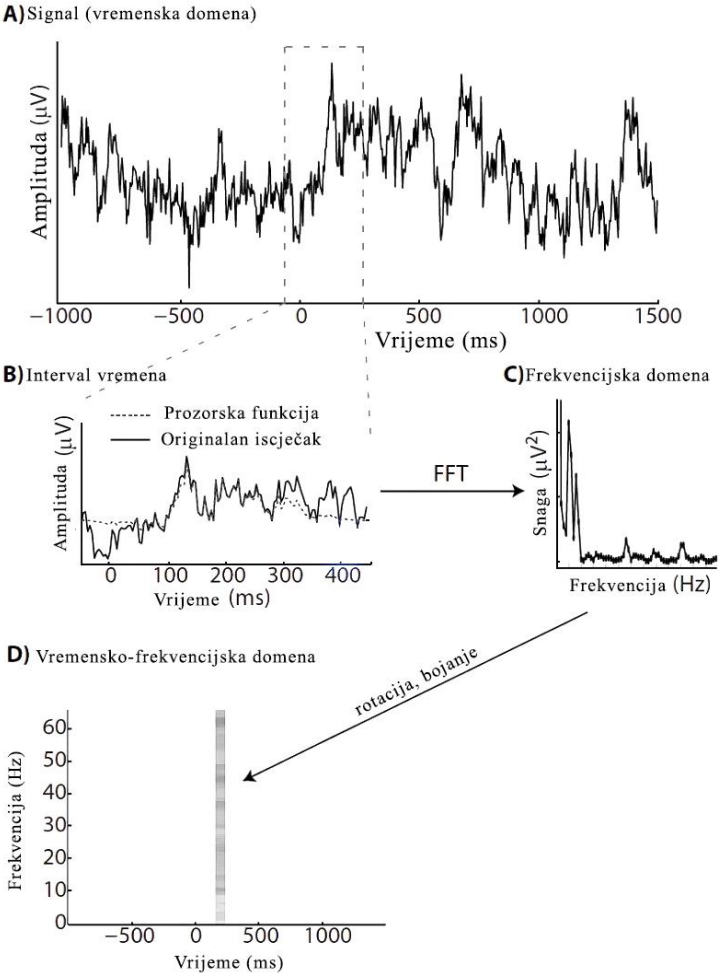
3.2.2.1 Kratkotrajna Fourierova transformacija

Transformacije zvučnih signala u frekvencijskoj domeni su se pokazale kvalitetnim načinom za njihovu reprezentaciju te su računalno efikasnije u usporedbi s oblicima u vremenskoj domeni. Jedna od implementacija FT je kratkotrajna FT (eng. a *Short Time Fourier Transform*, krat. STFT) koja daje informacije o frekvenciji za lokalnu točku vremena. Duljina prozora se definira na početku transformacije kao potencija broja 2. Prozor zatim „kliže" po signalu te se za svaki vremenski trenutak transformira u frekvencijsku domenu kao jedan okvir (eng. *frame*) jednake duljine kao prozor. Okvir sastavljaju košarice (eng. *bin*) s informacijama o frekvencijama u kojima se frekvencija prozora segmentira na frekvenciju uzorkovanja. Broj košarica odgovara duljini prozora, ali će uporabljive košarice frekvencija biti do Nyquistove frekvencije, odnosno $N / 2 + 1$ košarica (Slika 13) dok su ostale preslikane vrijednosti [7].



Slika 13 Upotrebljiv broj košarica [7]

Duljina košarice je time jednaka frekvenciji uzorkovanja podijeljenoj sa duljinom prozora. Dodatna bitna značajka STFT metode je korak pomaka (eng. *hop size*) za koji se prozor pomiče, koji najčešće iznosi pola duljine prozora i njime se postiže preklapanje, čime se zaglađuju rubovi i zaobilazi problem razmazivanja spektra. Osim toga, može se primijeniti prethodno opisana prozorska funkcija [7]. Pojednostavljen vizualan prikaz STFT metode dat je na Slika 14.



Slika 14 STFT metoda [7]

STFT se računa na sljedeći način:

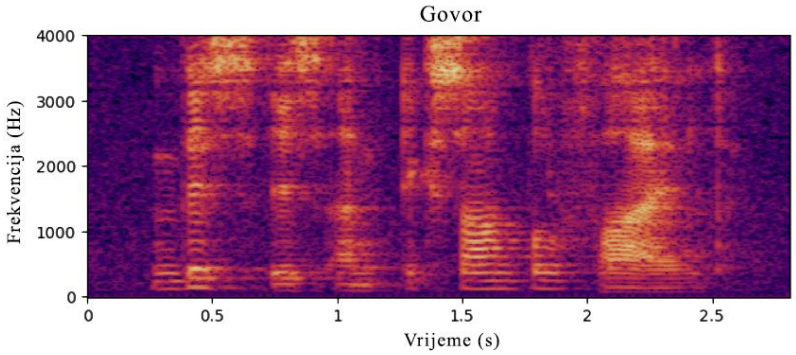
$$STFT\{x_n\}(h,k) = X(h,k) = \sum_{n=0}^{N-1} x_{n+h} w_n e^{-i2\pi\frac{kn}{N}}, \qquad (3.3)$$

gdje je w_n prozor, a h vremenski pomak. Ostatak izraza je jednak kao DFT. Izlaz STFT je matrica kompleksnih brojeva koji opisuju frekvencije i amplitude za svaku točku vremena te se može vizualizirati pomoću spektrograma[3]. Ono što

[3]Spektrogram zvuka – grafički prikaz triju parametara zvuka na jednom dijagramu gdje je na apscisi vrijeme, na ordinati frekvencija, a jakost zvuka se označava obojenjem. Izvor: *https://tl.lzmk.hr/clanak/spektrogram-zvuka*

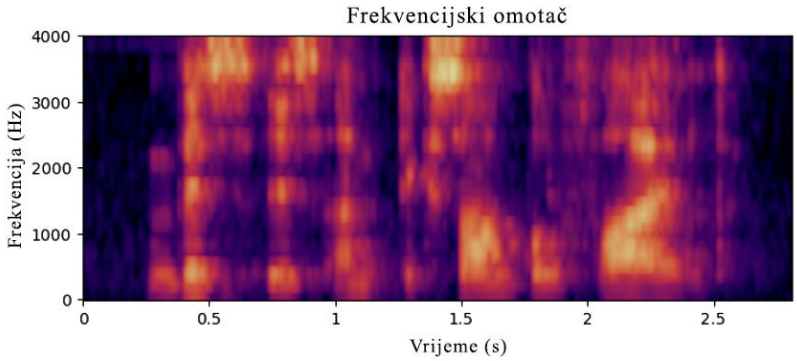
razlikuje STFT od TF je svojstvo klizanja prozora za svaki vremenski trenutak zbog čega je izlaz matrica [11].

Na sljedećim slikama je prikazan proces ekstrakcije formanta preko STFT spektrograma. Na Slika 15 je prikazan cjelovit spektrogram govora.

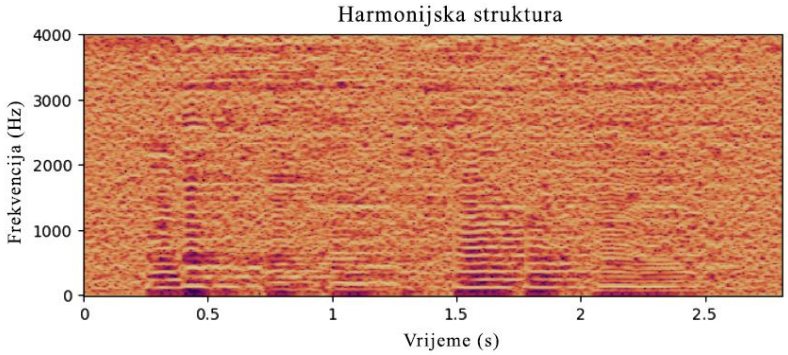


Slika 15 Spektrogram govora [8]

Zatim se, kako je opisano prema izvor-filter modelu, odvoji harmonijski dio (Slika 17) od spektralnog omotača koji sadrži informacije o formantima (Slika 16).

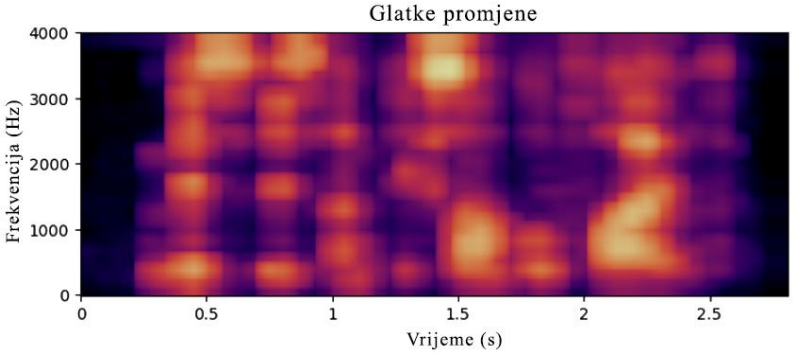


Slika 16 Spektrogram izdvojenog frekvencijskog omotača govora [8]

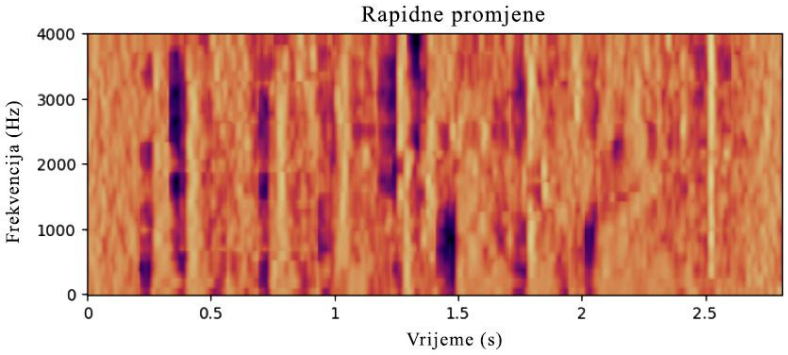


Slika 17 Spektrogram harmonijske strukture govora [8]

Na spektralnom omotaču su uočljive glatke (Slika 18) i rapidne promjene (Slika 19) kroz vrijeme [8].



Slika 18 Spektrogram formanta [8]



Slika 19 Spektrogram rapidnih promjena govora (npr. glas "k") [8]

3.2.2.2 Linearno prediktivno kodiranje

Linearno prediktivno kodiranje (eng. *Linear Predictive Coding*, krat. LPC) je jedna od popularnijih metoda za procjenu parametara govora te se često koristi za reprezentaciju govora za automatsko prepoznavanje govora. Važnost ove metode leži u jednostavnosti računanja i točnosti predikcije. Metoda se temelji na činjenici da se govor može aproksimirati linearnom kombinacijom prethodnih uzoraka govora. Lokalnim minimiziranjem greške se može odrediti jedinstveni skup koeficijenata predikcije, a jednadžbom ravnoteže između koeficijenata predikcije i koeficijenata razlike dobiva se pouzdana metoda za procjenu parametara koji karakteriziraju linearno vremenski promjenjiv model proizvodnje govora. LPC se odnosi na različite formulacije modeliranja govornog signala koje su u osnovi ekvivalentne. Razlike između formulacija se uglavnom odnose na detalje izračuna koji se koristi za dobivanje koeficijenata prediktora. Najčešće formulacije su:

- metoda kovarijance (eng. *covariance method*),
- formulacija autokovarijance (eng. *autocorrelation formulation*),
- metoda mrežaste strukture (eng. *lattice method*),
- formulacija inverznog filtera (eng. *inverse filter formulation*),
- formulacija estimacije spektra (eng. *spectral estimation formulation*),
- formulacija maksimalne izglednosti (eng. *maximum likelihood formulation*) i
- formulacija unutarnjeg produkta (eng. *inner product formulation*) [6].

Naime, govor je kontinuirani signal gdje su uzorci signala međusobno ovisni, odnosno povezani. U jednom segmentu zvuka, ti uzorci su relativno blizu jednog drugog. Na taj način se prema prethodnim uzorcima x_n može pretpostaviti sljedeći uzorak $\hat{x}_n$. Takav linearan prediktor koji koristi M prethodnih uzoraka bi glasio:

$$\hat{x}_n = -\sum_{k=1}^{M} a_k x_{n-k} \tag{3.4}$$

gdje su a_k težinski koeficijenti. Ostatak, odnosno greška predikcije bi onda glasila:

$$e_n = x_n + \sum_{k=1}^{M} a_k x_{n-k} = \sum_{k=0}^{M} a_k x_{n-k} = a_n x_n \tag{3.5}$$

gdje je $a_0 = 1$. U matričnom zapisu taj izraz glasi:

$$\begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{N-1} \end{bmatrix} = \begin{bmatrix} x_0 & \cdots & x_M \\ \vdots & \ddots & \vdots \\ x_{N-1} & \cdots & x_{N-M} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_M \end{bmatrix} \tag{3.6}$$

gdje je izračunata greška, odnosno ostatak do N uzorka signala. Takav oblik $\boldsymbol{X}$ matrice uzoraka je pogodna za metodu kovarijance[4]. Pretpostavljajući da je signal jednak nuli van određenih granica, matricu se može izmijeniti na način da su uzorci za $N - M < k < 0$ jednaki nuli te poprima sljedeći oblik:

$$\boldsymbol{X} = \begin{bmatrix} x_0 & 0 & \ldots & 0 \\ x_1 & x_0 & \ddots & \vdots \\ x_2 & x_1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \vdots & \vdots & & x_0 \\ x_{N-M} & x_{N-M-1} & & \vdots \\ 0 & x_{N-M} & \ddots & \vdots \\ \vdots & \ddots & \ddots & x_{N-M-1} \\ 0 & \ldots & 0 & x_{N-M} \end{bmatrix}.$$

[4] Kovarijanca – zbroj umnožaka odstupanja vrijednosti dviju varijabli od njihovih prosjeka te se koristi za mjerenje stupnja statističke povezanosti između pojava, izvor: *https://www.enciklopedija.hr/natuknica.aspx?id=33541*

Takav oblik matrice je pogodan za metodu autokovarijance[5] koja se najčešće koristi za pronalaženje koeficijenata predikcije. Dobivanje koeficijenata proizlazi iz greške predikcije, odnosno iz minimiziranja greške koristeći metodu minimalne srednje kvadratne pogreške (eng. *Minimum Mean-Square Error,* krat. MMSE). Očekivanje srednje greške u metodi autokovarijance se računa prema sljedećem izrazu:

$$E[\|e\|^2] = E[a^T \boldsymbol{X}^T \boldsymbol{X} a] = a^T E[\boldsymbol{X}^T \boldsymbol{X}] a = a^T \boldsymbol{A}_x a, \tag{3.7}$$

gdje je E operator očekivanja, a matrica $\boldsymbol{A}$ je dijagonalno-konstantna i simetrična Toeplitzova matrica oblika:

$$\boldsymbol{A} = \begin{bmatrix} a & b & c & d & e \\ f & a & b & c & d \\ g & f & a & b & c \\ h & g & f & a & b \\ i & h & g & f & a \end{bmatrix}.$$

Izraz (3.4) se svede na sustav normalnih jednadžbi[6] nakon uvođenja Lagrangeovih multiplikatora. Takav sustav se zatim može rekurzivno riješiti Levinson-Durbinovim algoritmom[7] namijenjenim za autoregresivne modele[8]. Time se omogućuje stabilnost metode kada se signal iterativno predviđa [8].

Jedna od primjena LPC transformacije je ta što inverznom filtracijom signala dobiveni polinom predikcije, sastavljen od LPC koeficijenata, predstavlja

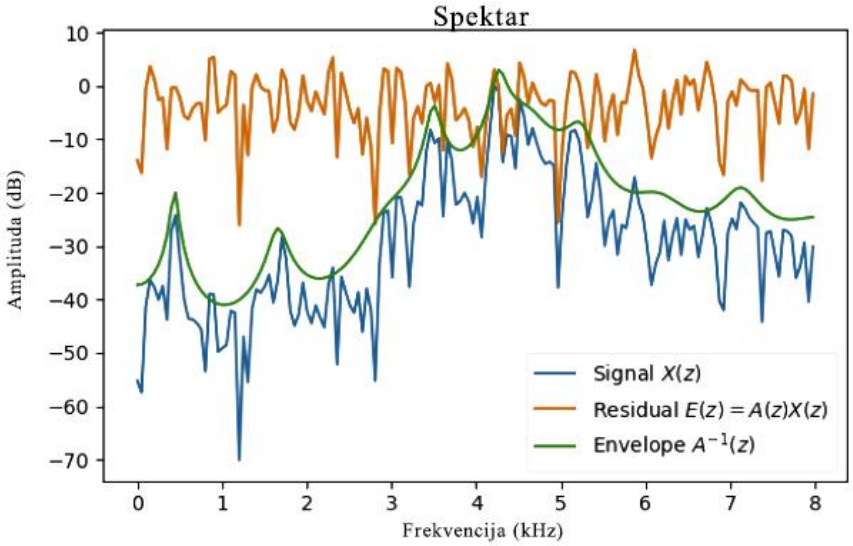
[5] Autokovarijanca - opisuje mjeru kovarijance procesa sa samim sobom u različitim trenucima vremena, izvor: *https://en.wikipedia.org/wiki/Autocovariance*

[6] Normalne jednadžbe – parcijalne derivacije sume kvadratnih greška su jednake nuli, izvor: *https://link.springer.com/referenceworkentry/10.1007/978-0-387-32833-1_286*

[7] Levinson-Durbinov algoritam – rekurzivno rješavanje problema reprezentacije vremenskih serija podataka, izvor: *https://en.wikipedia.org/wiki/Levinson_recursion*

[8] Autoregresivan model – model u kojem su izlazne vrijednosti rezultat vlastitih prethodnih vrijednosti, zvor: *https://en.wikipedia.org/wiki/Autoregressive_model*

spektralni omotač signala vokalnog trakta, odnosno filtera (Slika 20). Narančasto označen ostatak na slici predstavlja signal izvora odvojen od formanta [6].



Slika 20 Primjena LPC metode na zvučni signal s inverznom filtracijom [8]

3.2.2.3 Mel-frekvencijski kepstralni koeficijenti

Ekstrakcija mel-frekvencijskih kepstralnih koeficijenta (eng. *Mel-Frequency Cepstral Coefficients*, krat. MFCCs) je među najdominantnijim metodama ekstrakcija atributa govora zbog toga što se temelji na istraživanjima čovjekove slušne percepcije. Priprema zvučnog signala za ekstrakciju MFCCs započinje primjenom TTF na segmentiran zvučni signal gdje je duljina prozora najčešće 20-40 milisekundi. Zatim se računa periodogram na način da se izračuna spektralna gustoća (eng. *spectral density*) koja opisuje prosječnu energiju svakog prozora prema sljedećem izrazu:

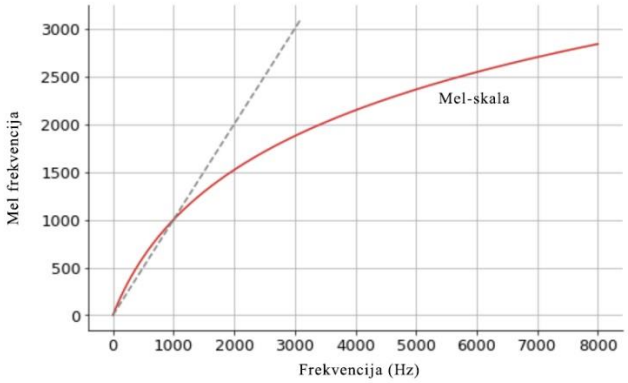
$$P(k) = \frac{1}{N}|S(k)|^2, \tag{3.8}$$

gdje je N broj uzoraka unutar prozora a k su cijeli brojevi od 1 do N. Daljnji postupak dobivanja MFCC se pretežno temelji na pojmu mel-skale. Mel-skala je perceptivna skala frekvencije zvuka iz koje se može zaključiti da ljudsko uho nije jednako osjetljivo na sve frekvencije, odnosno čovjek bolje percipira niže frekvencije. Skala je nastala eksperimentalno na način da su slušatelji morali namjestiti tonove tako da budu udaljeni u jednakim intervalima, no preciznost tih intervala je opadala s višim tonovima [12].

Mel-skala približno prati izraz:

$$M(f) = 1125\, ln\left(1 + \frac{f}{700}\right), \tag{3.9}$$

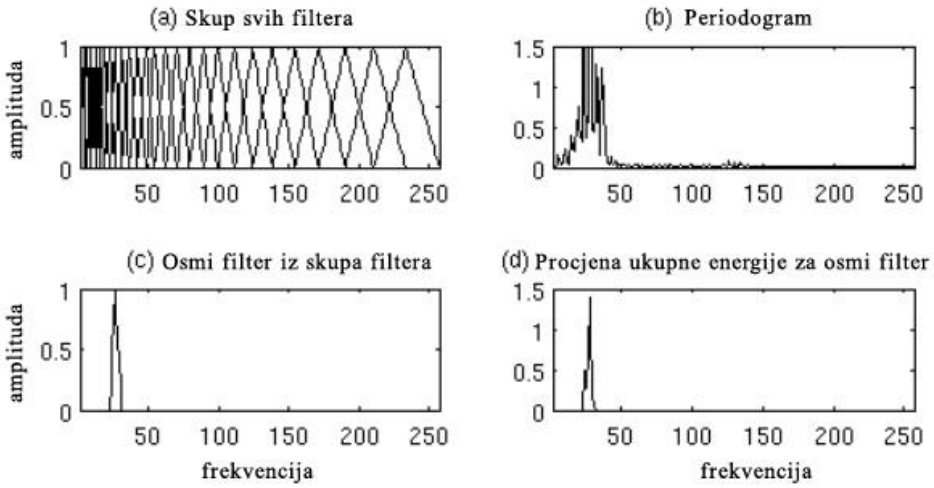
kako je prikazano na Slika 21.



Slika 21 Prikaz mel-skale [12]

Zatim se izrađuje skup mel-filtera u setu od 20-40 trokutastih filtera čija gustoća odgovara mel-skali. Bitno je napomenuti da je velika većina frekvencijskih komponenti govora u rangu do 8kHz, stoga je govor prema Nyquistovom teoremu dovoljno uzorkovati na 16kHz. Normalna duljina prozora je 512 uzoraka, što znači da je upotrebljivo 257 košarica. Broj filtera varira između 20 i 40, dok je

normalni broj 26. Svaki filter se u obliku vektora duljine 257 množi sa periodogramom te se dobiveni koeficijenti zbrajaju. Rezultat je procjena ukupne energije za svaki filter [11]. Proces je vizualno predočen na Slika 22.

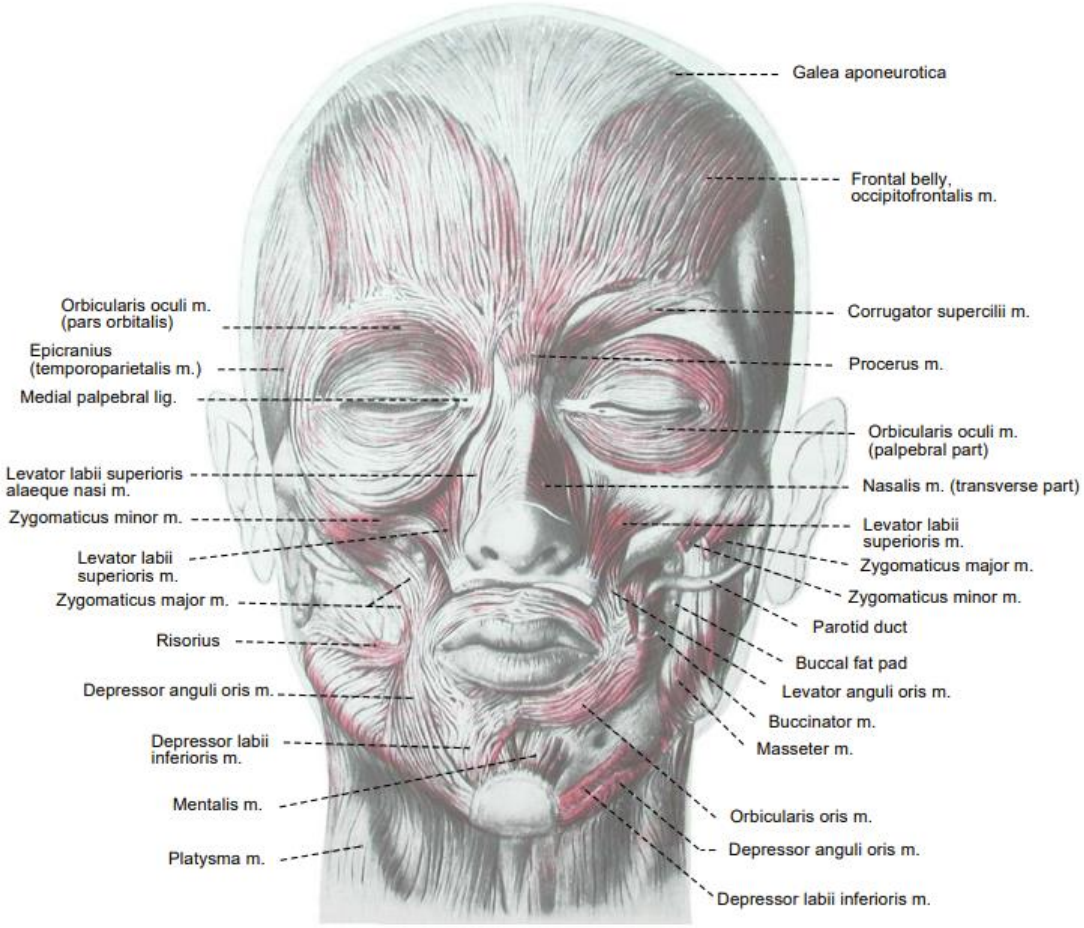


Slika 22 Prikaz primjene jednog filtera na periodogram [13]

Nakon toga se svih 26 procjena energije logaritamski skalira te se primijeni diskretna kosinusna transformacija (eng. *Discrete Cosine Transform*, krat. DCT). DCT izražava niz podataka kao zbroj kosinusnih funkcija s različitim frekvencijama, a primjenjuje se kako bi se smanjila korelacija dobivenih energija. Rezultat te transformacije je dijagonalna matrica kovarijance s kojom se mogu modelirati atributi govora. Dobiveni koeficijenti su MFCC od kojih se koristi prvih 12 jer su dovoljni za analizu govora [11].

4. Mapiranje točaka lica

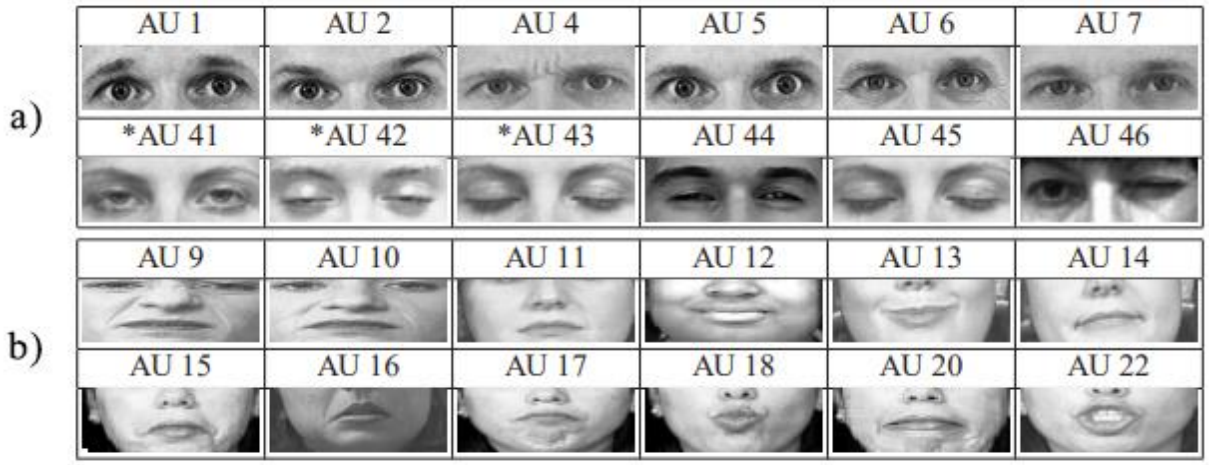
Ljudsko lice sačinjava 40 mišića koji upravljaju izrazima lica od kojih je većina vidljiva na Slika 23 [14]. Naime, velik broj kombinacija aktiviranih mišića zajedno s karakteristikama kože (npr. elastičnost kože) i jedinstvenosti pojedinaca predstavlja velik problem prilikom animacije lica. Dodatan aspekt problema je i ljudska priroda nasumičnih pokreta i trzaja, poput treptanja i mikro pokreta [15].



Slika 23 Mišići lica; lijevo su površni mišići, desno su unutarnji, dublji mišići [14]

Od sredine 20. stoljeća bilo je puno pokušaja kodiranja pokreta lica, odnosno opisivanja pokreta lica. Jedan od uspješnijih projekata je sustav kodiranja pokreta

lica (eng. *Facial Action Coding System*, krat. FACS). FACS se temelji na pokretima lica koje promatrač može raspoznati te ga sačinjava 44 elemenata pokreta (eng. *Action Unit*, krat. AU) od kojih se neke može vidjeti na Slika 24 [15].



Slika 24 AU sustava FACS; a) gornji dio lica, b) donji dio lica [15]

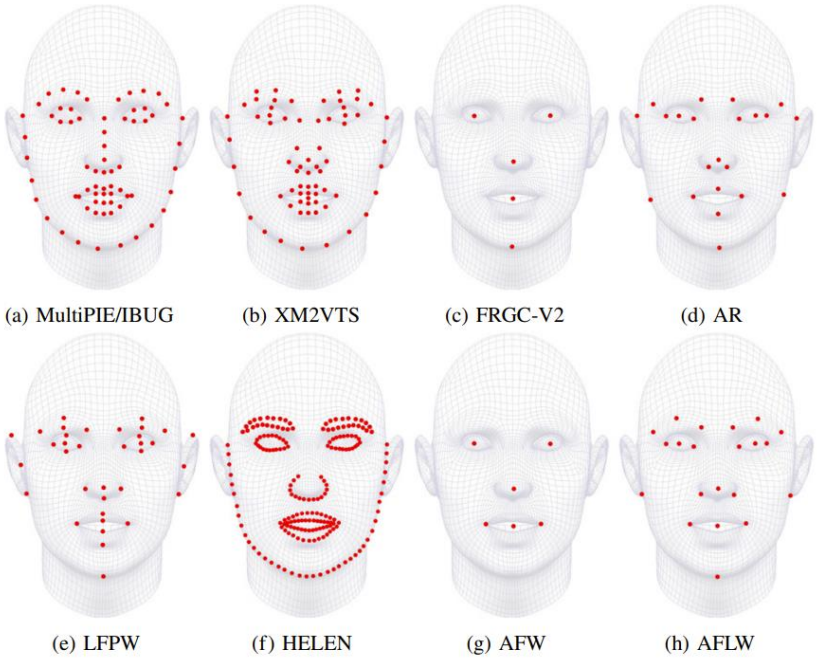
AU u različitim kombinacijama mogu opisati emociju (Slika 25).

Slika 25 Redom neutralan, tužan i ljut izraz lica kao kombinacija AU [15]

Na temelju opisanog sustava i sličnih koji će nastati nakon njega, provode se mnoga istraživanja o ključnim anatomskim točkama lica, aktivnosti facijalnih mišića i praćenju promjena u izrazima lica. Izrada modela za praćenje točaka lica je postao relativno lagan posao prvenstveno jer postoji velik broj različitih baza podataka označenih slika lica s različitim brojem i mjestima oznaka. U Tablica 1 su opisane neke od navedenih baza podataka koja prati Slika 26 [16].

Tablica 1 Karakteristike baza podataka za treniranje modela mapiranja točaka lica

Oznaka na slici	Baza podataka	Broj slika	Broj subjekata	Broj točaka	Poza	Osvjetljenje	Izraz lica
a)	Multi-PIE	750,000	337	68	15 poza	19 različitih osvjetljenja	6 izraza lica
b)	XM2VTS	2360	295	68	Frontalna	Jednako	Neutralno
c)	FRGC-V2	4950	466	5	Frontalna	Kontrolirano i nekontrolirano	Neutralno i nasmiješeno
d)	AR	+4000	126	22	Frontalna	Različita osvjetljenja	Različiti izrazi lica
e)	LFPW	1287	(Slike s interneta)	35	Različite poze	Različita osvjetljenja	Različiti izrazi lica
f)	HELEN	2330	(Slike s *flickr.com* stranice)	194	Različite poze	Različita osvjetljenja	Različiti izrazi lica
g)	AFW	250	468 različitih lica	6	Različite poze	Različita osvjetljenja	Različiti izrazi lica
h)	AFLW	25,993	(Slike s *flickr.com* stranice)	21	Različite poze	Različita osvjetljenja	Različiti izrazi lica
a)	IBUG	135	(Slike s interneta)	68	Različite poze	Različita osvjetljenja	Različiti izrazi lica



Slika 26 Konfiguracije točaka lica za različite baze podataka [16]

Osim toga, danas je moguće primijeniti širok spektar metoda umjetne inteligencije za izradu vlastitih modela koji prate točke lica, no također postoji veliki broj gotovih modela koji se mogu primijeniti u sklopu različitih programskih jezika. Najpopularnije knjižnice otvorenog izvora su navedene u Tablica 2.

Tablica 2 Popularne knjižnice za detekciju točaka lica

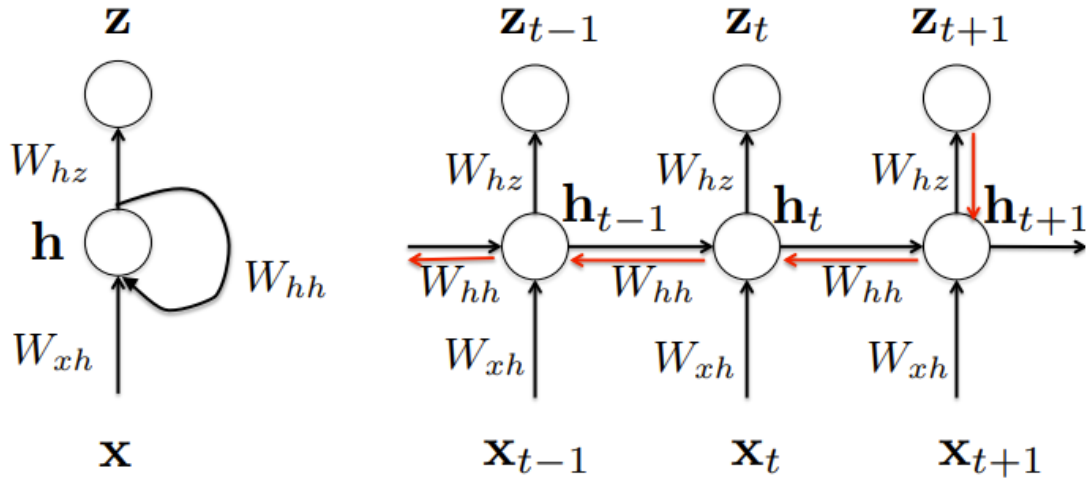
Knjižnica	**Alat**	**Jezik**
Dlib	face_landmark_detection	C++, Python
OpenCV	face module	C++, Python
MediaPipe	FaceMesh	Python (TensorFlow)
PyTorch	Pytorch Face Landmark Detection	Python
TensorFlow	TensorFlow Face Landmark Detection	Python

5. Generiranje animacije lica vođeno zvukom

Do sada su se obradili podaci koji će biti ulaz za metode dubokog učenja, specifičnije za neuronske mreže. Vrsta i arhitektura same neuronske mreže ovisi o zadatku za koji je namijenjena. Kako je već navedeno, u ovom radu se koncentrira na zadatak generiranja točaka lica vođeno zvukom. Ta tehnologija se prvenstveno primjenjuje u interakciji čovjeka i računala (eng. *Human-Computer Interaction,* krat. HCI). Konkretnija primjena uključuje virtualne agente u različitim aspektima ljudskog života od uslužnih djelatnosti do edukacije. Osim poboljšanja korisničkog iskustva, takva tehnologija bi mogla i poboljšati kvalitetu života ljudi s raznim invaliditetima. Napredak bi se osjetio i primjerice u industriji video igara te prilikom animiranja ljudi u filmskoj industriji. Uspješan projekt takve vrste zahtjeva ekspertna znanja iz područja računalnih znanosti, psihologije, strojnog učenja i medicine [17]. U nastavku će se opisati temeljni principi najkorištenijih neuronskih mreža za opisani zadatak. Naime, osim dubokog učenja postoje starije metode korištene za ovakav tip zadataka, no razvojem i primjenom metoda dubokog učenja se postižu bolji rezultati na kompleksnijim razinama.

5.1 RNN

Zvuk i video su sekvence, odnosno niz podataka, gdje sljedeći podatak x_n ovisi o prethodnom x_{n-1} podatku. Rekurentne neuronske mreže (eng. *Recurrent Neural Networks*, krat. RNN) su vrsta neuronskih mreža koje su pogodne za obradu sekvencijalnih podatka. Ključna razlika između RNN i prethodno već opisanih klasičnih FNN je način na koji informacije prolaze kroz mrežu. Točnije, RNN vraća vlastite vrijednosti nazad u ciklus protoka informacija.



Slika 27 Pojednostavljen primjer RNN; lijevo – rekurzivni prikaz, desno – proširena („odmotana“) struktura [18]

Za početak je potrebno definirati pojam vremenske oznake t (eng. *time-stamp*) koja označava poziciju unutar sekvence. Promatrajući Slika 27, $\boldsymbol{x}$ je ukupna sekvenca koja se promatra, $\boldsymbol{h}$ je vektor prethodnih skrivenih stanja, $\boldsymbol{z}$ je predikcija, a $\boldsymbol{W}$ su matrice težina s time da su $\boldsymbol{W}_{yh}$ i $\boldsymbol{W}_{yh}$ analogne matricama iz FNN, a $\boldsymbol{W}_{hh}$ određuje težinu komponenta skrivenih stanja. Skriveno stanje za neki t se može prikazati kao nelinearna ovisnost dijela sekvence $\boldsymbol{x}_t$ i prethodnog skrivenog stanja prema izrazu:

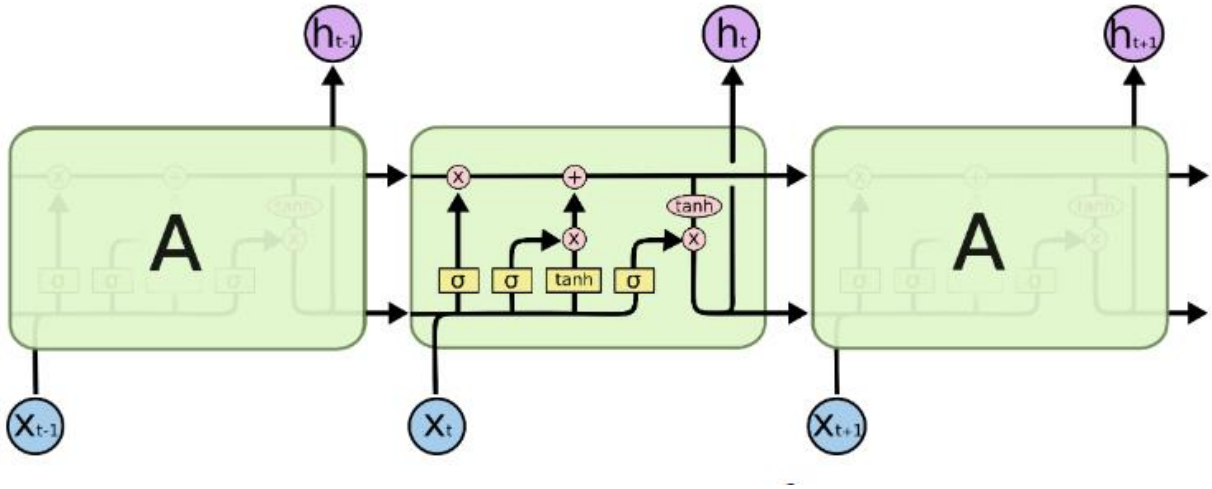
$$\boldsymbol{h}_t = f(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t), \tag{5.1}$$

prema čemu $\boldsymbol{h}_t$ sadrži znanje o cijeloj sekvenci te služi kao dugotrajna memorija. Na ta način mreža uči uzimajući u obzir prethodne podatke [18].

Problem RNN se pojavljuje prilikom učenja mreže primjenom algoritma propagacije unatrag. Algoritmom propagacije kroz mnogo slojeva, gradijent može nestati ili eksplodirati. U RNN se to događa jer svi dijelovi sekvence $\boldsymbol{x}_t$ dijele matrice težina kroz mrežu, odnosno matrice težina su rekurentne. Naime, algoritmom propagacije unazad, pri izračunu gradijenta događa se uzastopno se

množenje s derivacijom matrice $\boldsymbol{W}_{hh}$. Ako su vrijednosti derivacije jako male gradijent će se kontinuirano približavati nuli. Kako se težine mijenjaju s obzirom na gradijent, zbog jako malog gradijenta težine bi se sve manje mijenjale što se algoritam „dublje“ vraća u mrežu. S druge strane, velike vrijednosti derivacija bi rezultirale dosta većim gradijentom i time bi se znatno mijenjale težine. Na taj način mreža ne može učiti, odnosno pravilno optimizirati težine [19].

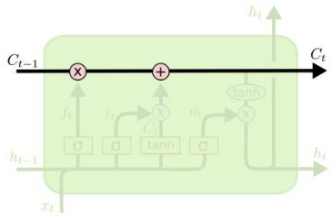
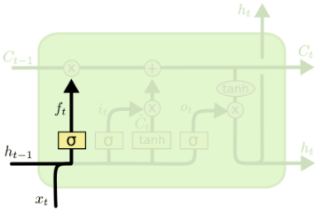
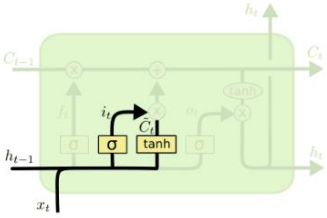
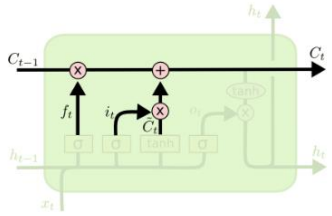
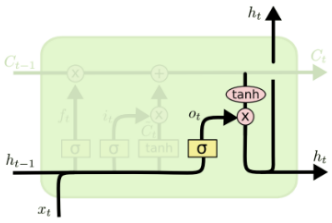
Jedno od rješenja navedenog problema je varijacija RNN s ćelijama dugoročne kratkotrajne memorije (eng. *Long Short-Term Memory,* krat. LSTM) koje se koriste umjesto perceptrona. Osnovna ideja LSTM je dugoročno učenje s time da se ne uči sve, već se kontroliraju podaci koji se uče, odnosno zaboravljaju. U nizu s drugim ćelijama, struktura mreže bi izgledala kao na Slika 28.



Slika 28 Arhitektura LSTM mreže [20]

U Tablica 3 je razložen proces kontrole toka informacija.

Tablica 3 Proces upravljanja informacijama LSTM ćelije [20]

Korak	Prikaz	Opis
Glavni tok informacija		Slobodan protok informacija s minimalnim promjenama
Propusnica zaboravljanja (eng. *forget gate*)		Sigmoidni sloj kojim se odlučuje koje informacije se zadržavaju u kojoj mjeri, gdje je 0 potpuno odbacivanje, a 1 potpuno zadržavanje
Propusnica novog ulaza (eng. *candidate cell state*)		Sigmoidni sloj odlučuje koja vrijednosti će se ažurirati, dok se hiperboličnim tangesom stvara vektor novih kandidata
Ažuriranje odluka		Ažuriranje prethodno donesenih odluka operacijama množenja i zbrajanja
Izlaz		Prvo se sigmoidnim slojem odlučuje koji dijelovi ćelije će izaći, zatim se hiperboličnim tangesom normalizira vrijednost između -1 i 1

Gradijent LSTM mreže je dosta kompleksniji, no krajnje sređen izraz se može zapisati u obliku zbroja (za razliku od umnoška kod RNN) te će rezultat često biti

iznad 1 ili između 0 i 1. U slučaju da gradijent počne konvergirati prema 0, dovoljno je povećati vrijednosti propusnice zaboravljanja i drugih dijelova ćelije kako bi se gradijent približio jedinici.

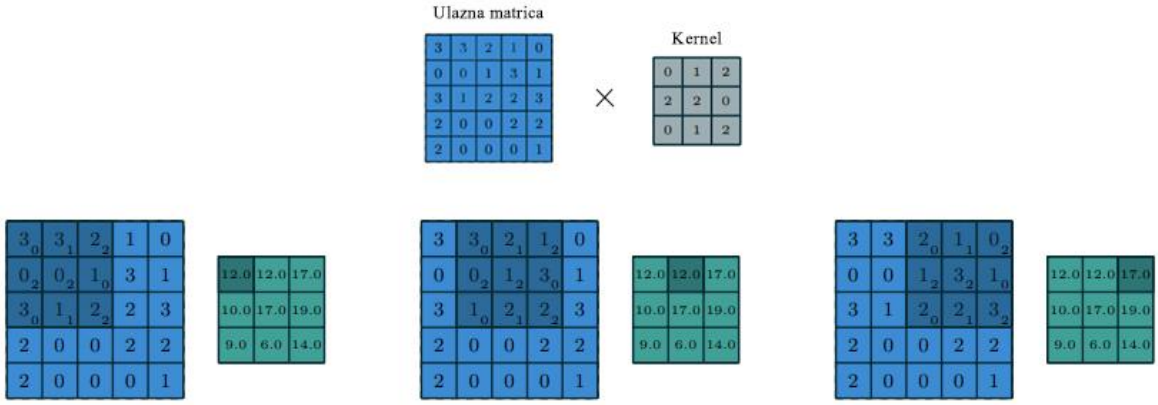
5.2 CNN

Konvolucijske neuronske mreže (eng. *Convolutional Neural Networks*, krat. CNN) se temelje na matematičkoj operaciji konvolucije koja se ponaša kao filter. Konvolucija izražava utjecaj jedne funkcije na drugu u obliku treće, nove funkcije. Matematički se može opisati kao integral produkta dviju funkcija na njihovom intervalu. Kao filter, ta operacija je iznimno jednostavna i učinkovita što je jedan od razloga zašto su CNN među najpopularnijim metodama dubokog učenja [21].

Naime, većina vrsta podataka se, bez obzira na njihovu dimenzionalnost, može opisati kao matrica ili skup matrica. Primjerice slika je $width \times height$ matrica gdje su vrijednosti intenzitet, odnosno boja piksela. Ako je slika u boji, matrica poprima i dimenziju dubine gdje se jedna slika sastoji od tri matrice prema RGB modelu boja. Na primjer, broj piksela u obojanoj slici koja je visoka i široka 50 piksela bi se računao kao 50 piksela × 50 piksela × 3 kanala boje (crveno, zeleno i plavo). Često se, zbog bržeg i jednostavnijeg računanja, slike konvertiraju u crno-bijelu verziju (eng. *grayscale*) čime dimenzija dubine postaje 1 [19].

Filter konvolucije će najčešće predstavljati 3×3 matrica koja se naziva kernel. Na Slika 29 je vidljivo kako se odvija proces filtracije konvolucijom. Kernel sa svojim vrijednostima kliže s nekim korakom po ulaznoj matrici vrijednosti (koja predstavlja sliku) te se poklapajuće vrijednosti množe i naposljetku zbroje. Dobivena vrijednost će biti element nove matrice čija pozicija odgovara poziciji kernela na ulaznoj matrici. Novo dobivena matrica se naziva izlazna mapa

značajki (eng. *output feature maps*). Opisani primjer je 2D konvolucija, u slučaju da podaci imaju i dubinu, kernel bi također imao dubinu te bi dodatno klizao i na osi dubine. Izlaz bi bile 2D mape značajki (3 mape za RGB model) koje se na kraju zbrajaju [21].

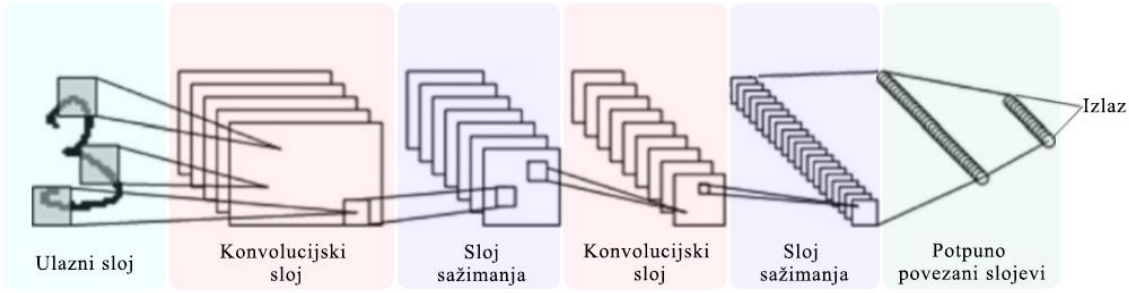


Slika 29 Primjer konvolucije [21]

Filteri u prvim slojevima mreže uče opće karakteristike kao što su primjerice rubovi (Slika 30). Što su slojevi dublje unutar mreže, to su sposobniji otkrivati kompleksnije atribute [19].

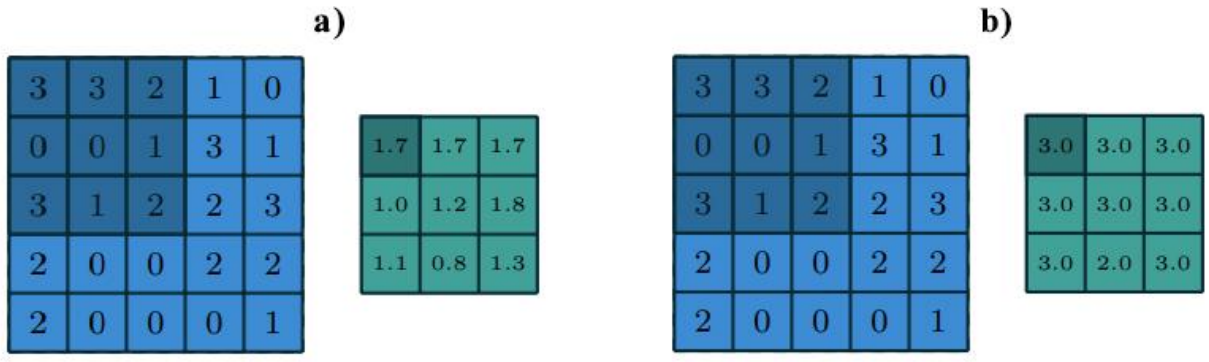
Slika 30 Primjer naučenih filtera CNN nakon prvog sloja konvolucije

Konvolucija se odvija u slojevima konvolucije kao ukupne arhitekture CNN prikazane na Slika 31.



Slika 31 Arhitektura CNN

Osim slojeva konvolucije, slojevi sažimanja (eng. *pooling layer*) predstavljaju bitan aspekt CNN. Njihova uloga je smanjenje veličina mapi značajki s time da se zadrže bitni atributi. Proces je sličan procesu konvolucije gdje funkcija sažimanja klizi preko mape značajki te nakon primjene operacije sažimanja rezultira mapom značajki s manjim dimenzijama. Najčešće operacije sažimanja (Slika 32) su izbor maksimalne vrijednosti i sažimanje pomoću prosječne vrijednosti [21].



Slika 32 Operacije sažimanja; a) prosječna vrijednost, b) izbor maksimalne vrijednosti [21]

Prije izlaza se nalaze potpuno povezani slojevi (eng. *fully connected*, krat. FC) sastavljeni od neurona koji su potpuno povezani s prethodnim slojem.

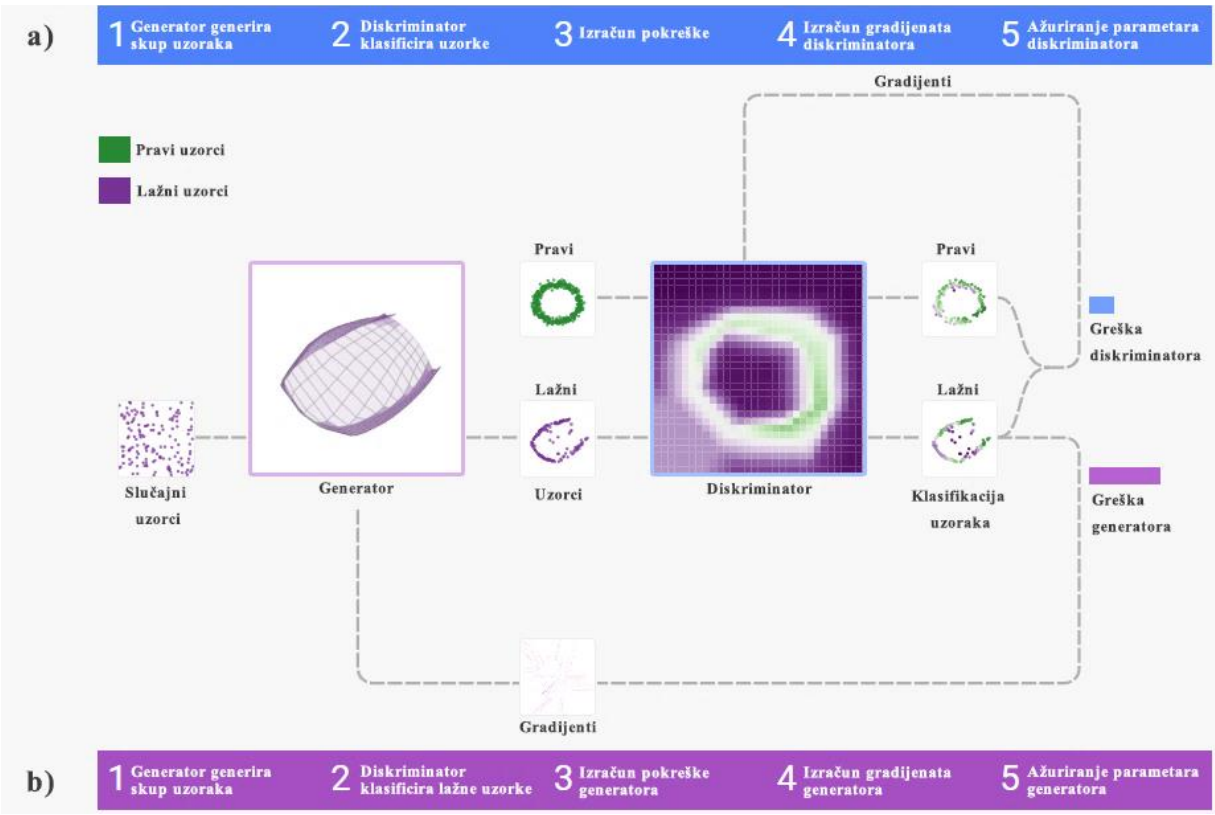
Naposljetku, izbor funkcije izlaza ovisi o samoj primjeni mreže, a treniranje CNN se izvršava algoritmom propagacije unazad [19].

5.3 GAN

Generativna suparnička mreža (eng. *Generative Adversarial Network,* krat. GAN) je sastavljena od dvije neuronske mreže s time da jedna ima ulogu generatora, dok druga ima ulogu diskriminatora. Generator generira umjetne primjere objekta prema datoj bazi podataka. Diskriminativna mreža zatim uzima ili prave primjere iz baze podataka ili umjetne, generirane primjere te pokušava odrediti je li je primjer pravi ili umjetan. Iteracijom tog procesa, obje mreže se usavršavaju te naizmjenično uče, generator sve bolje stvara umjetne slike, a diskriminator sve bolje pronalazi razlike između pravih i umjetnih slika. Odnosno, mreže se natječu u učenju [19].

Promatrajući Slika 33 se može opisati pojednostavljena arhitektura i učenje GAN. Proces se može podijeliti na treniranje diskriminatora i treniranje generatora:

a) Treniranje diskriminatora – generator prvo generira slučajnu varijablu sa sličnom distribucijom vjerojatnosti kao što je distribucija stvarnih podataka. Diskriminator zatim klasificira podatke između lažnih i pravih, odnosno između 0 i 1. Zatim se računa greška diskriminatora i pripadajući gradijenti. Parametri diskriminatora se zatim mijenjaju s obzirom na gradijente.
b) Treniranje generatora – generator generira slučajnu varijablu kao u prethodnom koraku. Zatim diskriminator klasificira samo lažne podatke nakon čega se računa greška generatora i gradijenti prema čemu se mijenjaju parametri generatora [22].



Slika 33 Arhitektura i proces treniranja GAN [22]

5.4 Duboko učenje s više modaliteta

Dosad su se obrađivale metode dubokog učenja koje su sposobne učiti iz jedne vrste ulaznih podataka. Generiranje lica vođeno zvukom se može opisati sa dvije vrste podataka: zvuk i video podaci. S obzirom na njihovu međusobnu ovisnost prilikom govora, proizlazi potreba za novim modelom koji je sposoban učiti iz obje vrste podataka kako bi predviđeni podaci bili rezultat njihove kombinacije. Čovjek ima sposobnost zaključivanja iz više osjetila odjednom s naglaskom na kombinaciju sluha i vida. Kako bi model bio sposoban učiti na sličan način, potrebno je izraditi takvu vrstu modela koja može uzajamno učiti iz više modaliteta, gdje je modalitet način na koji se nešto odvija ili doživljava [23]. Izrada modela dubokog učenja s više modaliteta (eng. *Multimodal Deep Learning*, krat. MDL) predstavlja kompleksan niz koraka te će se u nastavku opisati osnovni

izazovi i koraci koji se vežu za njih. S obzirom da svaki korak najčešće ima više pristupa u ovisnosti o vrsti zadatka, obraditi će se koraci relevantni za zadatak generiranja lica vođeno zvukom.

5.4.1 Reprezentacija podataka

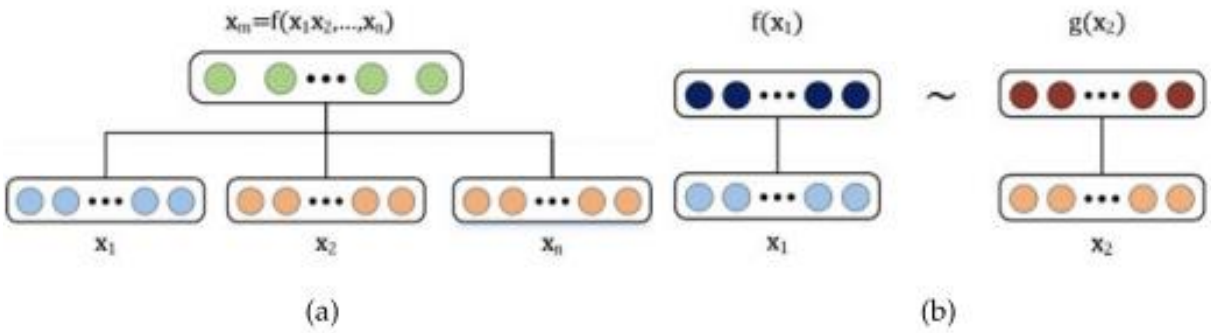
Prije nego što je moguće trenirati model dubokog učenja, podatke je potrebno konvertirati u format koji je razumljiv modelu. U slučaju MDL modela potrebno je na neki način povezati dvije vrste signala što je ujedno i osnovna ideja takvih modela. Na taj način izbori reprezentacija signala su međusobno ovisni. Postoje dva različita pristupa za reprezentaciju više modaliteta. Spojene reprezentacije (Slika 34.a) kombiniraju signale s jednim modalitetom u isti prostor reprezentacije prema izrazu:

$$x_m = f(x_1, \dots, x_n) \tag{5.2}$$

gdje je f funkcija koja računa spojenu reprezentaciju. S druge strane, koordinirane reprezentacije zadržavaju jednomodalni oblik ulaza, no također definiraju neke restrikcije na temelju sličnosti između signala kako bi ih se zadržalo u koordiniranom prostoru. Koordinirane reprezentacije (Slika 34.b) su date sa:

$$f(x_1) \sim g(x_2). \tag{5.3}$$

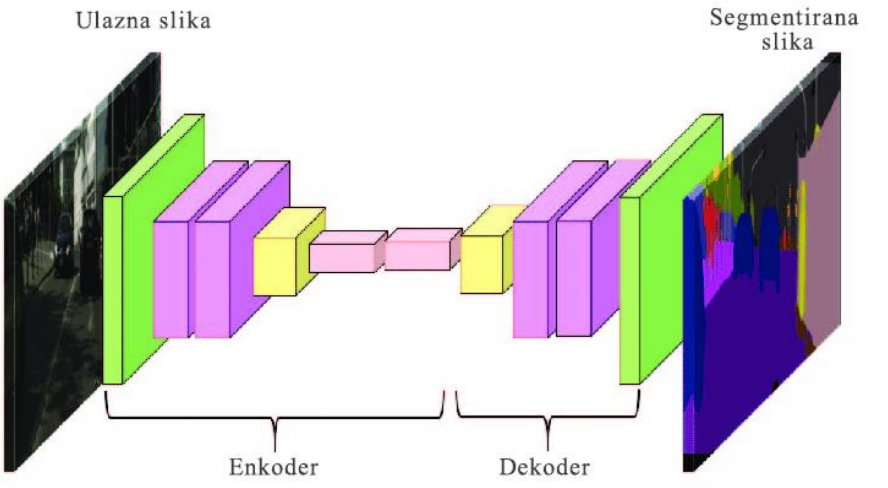
Spojene reprezentacije, koje se nazivaju i ranom fuzijom, se mogu dobiti dubokim neuronskim mrežama koje su pogodne za spajanje slika sa zvukom i tekstom. Za sekvencijalne vrste podataka se mogu koristiti RNN koje su korisne za spajanje zvuka i videa, te slika i teksta. Duboke neuronske mreže su popularan alat za učenje koordiniranih reprezentacija [23].



Slika 34 Reprezentacije podataka za MDL; a) spojene reprezentacije, b) koordinirane reprezentacije [23]

5.4.2 Mapiranje informacija s modaliteta na modalitet

Mapiranje informacija s jednog modaliteta na drugi je usko povezano sa zadatkom generiranja lica pomoću zvuka. Naime, za taj zadatak je potrebno mapirati informacije iz zvučnog modaliteta u vizualni modalitet. S obzirom da je izlaz modela sekvenca koordinata točaka lica, a ulaz zvuk, mapiranje se postiže kontinuiranim generativnim modelima. Oni su namijenjeni za procesiranje sekvenci te stvaraju neki izlaz za svaku vremensku oznaku. Problem takvog modela je održavanje temporalne konzistentnosti između modaliteta [23]. Jedan od popularnijih pristupa tom izazovu su kontinuirani enkoder-dekoder modeli sastavljeni od enkodera i dekodera. Enkoder prvo obrađuje ulaznu sekvencu i proizvodi „kontekst“ što je obično zadnje stanje skrivenih slojeva. Kontekst se bolje može opisati kao latentni atributi. Zatim dekoder koristi proslijeđeni kontekst i generira izlaznu sekvencu u željenom obliku. Posebnost takvog modela je to što duljine ulazne i izlazne sekvence ne moraju biti jednake [3]. Na Slika 35 je vidljiva pojednostavljena arhitektura modela za segmentaciju slika.



Slika 35 Enkoder-dekoder s ulogom segmentacije slika [24]

Uloge enkodera i dekodera najčešće imaju neke od opisani neuronskih mreža u prethodnim poglavljima rada.

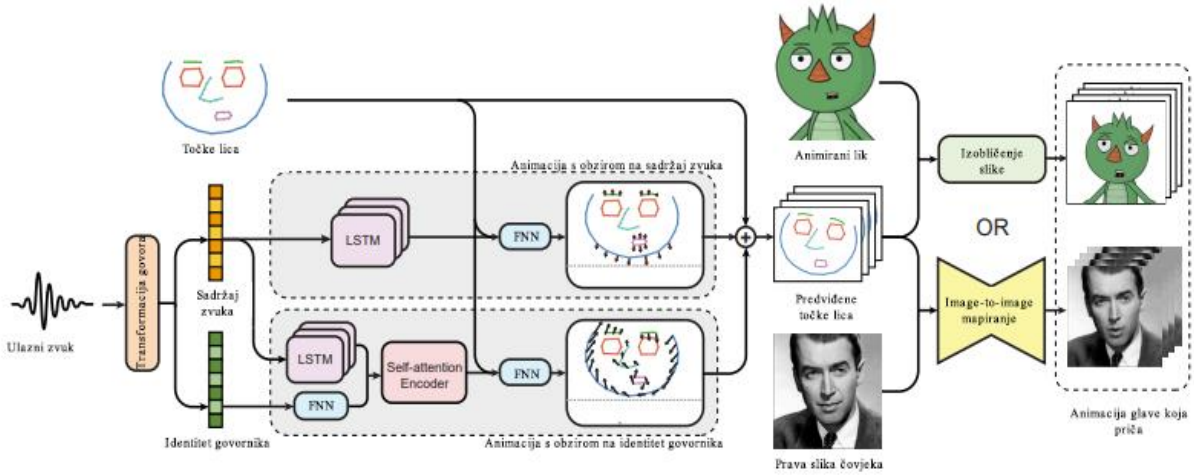
5.4.3 Fuzija informacija više modaliteta

Fuzija informacija podrazumijeva integraciju sakupljenih informacija iz više modaliteta s ciljem dobivanja predikcije. Fuzijom se zaobilazi problem kada informacije iz jednog modaliteta nedostaju (primjerice ako osoba šuti). Postoje dva pristupa izvršavanju fuzije. Prvi pristup nije ovisan o modelu te se dijeli na ranu i kasnu fuziju. Rana fuzija integrira značajke odmah na početku kako je već opisano, dok se kasna fuzija odvija nakon gotovog učenja svakog modaliteta. S druge strane, postoji pristup koji ovisi o modela gdje u kontekstu neuronskih mreža proces fuzije ovisi o vrsti mreže te je dio rada same mreže, primjerice prilikom primjene klasifikatora na izlazu iz mreže [23].

5.5 Pregled relevantnih istraživanja

U ovom dijelu rada će se navesti neka relevantna istraživanja s ostvarenim praktičnim implementacijama. Svi radovi su vezani za generiranje animacije lica pomoću zvuka. Princip rada za svaki model će biti opisan u kratkim crtama.

5.5.1 MakeItTalk

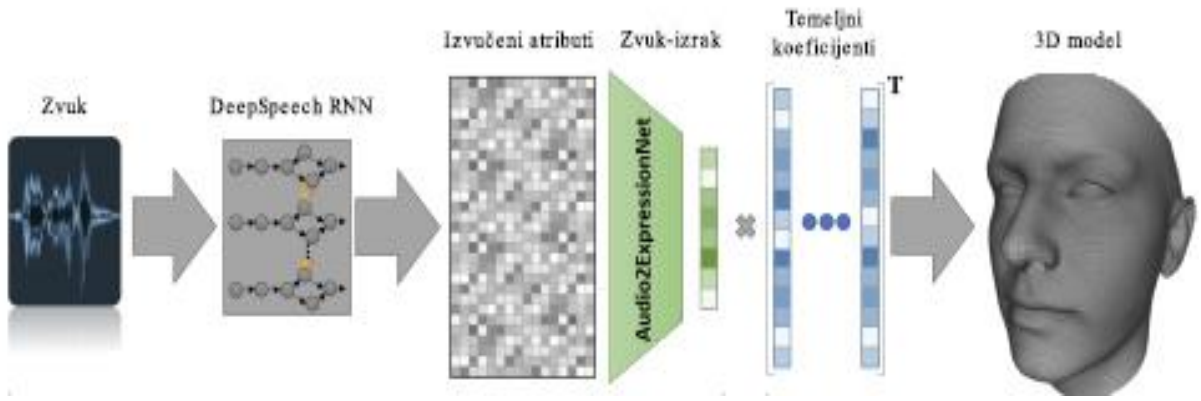


Slika 36 MakeItTalk model [25]

Model *MakeItTalk* (Slika 36) iz jedne slike i iz zvuka generira animaciju slike, s time da se može primijeniti na sliku prave osobe i nestvarnih likova. Za početak se zvuk komprimira u dvije reprezentacije – sadržajnu i reprezentaciju identiteta. Sadržajna reprezentacija ne ovisi o govorniku te je skup atributa zvuka koji su vezani za pokrete usana i obližnjih regija. Reprezentacija identiteta sadrži skup atributa vezanih za pojedinog govornika. Nakon što model generira točke lica potrebno ih je mapirati na sliku. Za nerealistične slike koristi se algoritam

izobličenja slika Delaunay triangulacijom[9] dok se za realistične slike koristi *image-to-image* mreža. Transformaciju zvuka izvršava LSTM enkoder, zatim se obje reprezentacije šalju u dekoder, FNN koja predviđa koordinate točaka. Naposljetku se generirani okviri spajaju te preko dodatnog enkoder-dekoder modela povezuju sa slikom koja se animira [25].

5.5.2 Neural Voice Puppetry



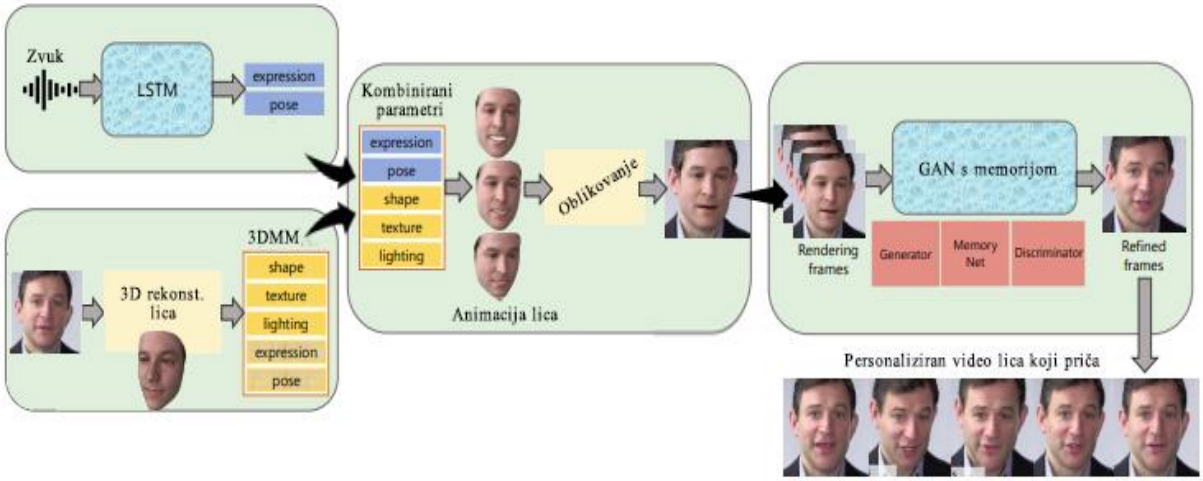
Slika 37 Neural Voice puppetry model [26]

Model *Neural Voice puppetry* (Slika 37) započinje izvlačenjem generalnog latentnog vektora izraza lica iz zvuka. Time se stvara zvuk-izraz prostor koji se dijeli između svih lica, odnosno omogućen je prijenos pokreta lica s osobe na osobu. Prvo se pomoću gotovog RNN modela *DeepSpeech* iz govora izvuku generalizirani atributi govora. Atributi se zatim prosljeđuju CNN koja smanjuje dimenzionalnost podataka govora. Izlaz zadnjeg FC sloja mreže je vektor u prethodno opisanom zvuk-izraz prostoru. Zatim se koeficijenti iz zvuk-izraz

[9] Delonayeva triangulacija - geometrijska tehnika koja podijeli skup točaka na troktue tako da se minimizira najveći kut. Mrežom trokuta se zatim može upravljati čime animacija izgleda kvalitetno, izvor: *https://gwlucastrig.github.io/TinfourDocs/DelaunayIntro/index.html*

prostora linearno preslikavaju na koeficijente koji opisuju 3D lice modela. Težine tog linearnog odnosa se dobivaju 1D konvolucijskim slojevima [26].

5.5.3 Audio-driven Talking Face Video Generation

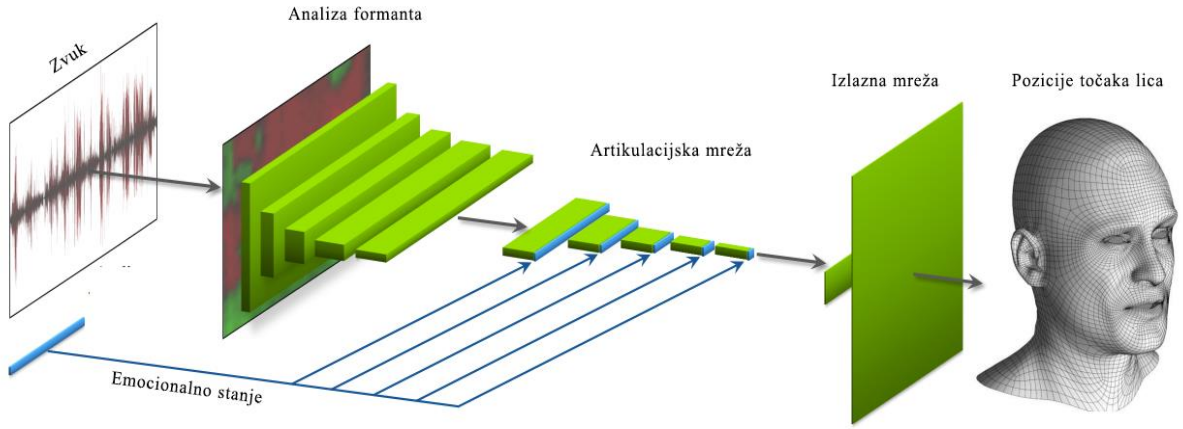


Slika 38 Audio-driven Talking Face Video Generation model

Model *Audio-driven Talking Face Video Generation* (Slika 38) započinje rekonstrukcijom 3D modela lica iz slike lica pomoću CNN. Iz zvuka se zatim izdvoje MFCC koji daju uvid u izraz lica te se pomoću LSTM mreže kombiniraju s 3DMM koeficijentima[10] iz 3D modela lica. Iz dobivenih koeficijenata se renderiraju slike lica s izrazima koji odgovaraju zvuku. Kako bi se dobivene slike poboljšale i učinile realističnijima, primjenjuje se GAN s memorijom. Memorija sadržava sparene spacijalne značajke sa značajkama identiteta te se koristi za dohvaćanje bliskih identiteta, pridonoseći personaliziranom generiranju slika.

[10] 3D Morphable Model koeficijenti – parametri koji omogućuju prilagodbu statističkog 3D modela lica kako bi se stvorio model koji odražava pojedinačne karakteristike i varijacije osobe

5.5.4 Audio2Face: Audio-Driven Facial Animation



Slika 39 Audio-Driven Facial Animation by Joint End-to-End Learning of Pose and Emotion model [27]

Istraživanje *Audio-Driven Facial Animation by Joint End-to-End Learning of Pose and Emotion* je praktično ostvareno u obliku aplikacije *Audio2Face* koja je odabrana za izvedbu praktičnog dijela ovog rada. Naime, temelj ovog modela je *end-to-end* mreža koja ujedno obrađuje podatke i izvršava svoj zadatak.

Model (Slika 39) započinje mrežom koja analizira formante iz LPC koeficijenata pomoću konvolucijskih slojeva. Na taj način se dobivaju kratkotrajne vremenske reprezentacije formanta koji su ključne za animiranje lica. Rezultat se zatim prosljeđuje u artikulacijsku mrežu koju također sastavljaju konvolucijski slojevi, no njihova je uloga analizirati vremenski razvoj atributa lica te je izlaz vektor atributa lica. Sekundarni ulaz u artikulacijsku mrežu je emocionalno stanje koje će predstavljati vektor. Izlaz artikulacijske mreže je spoj vektora atributa i atributa emocija koji skupa tvore izraz lica. Izlazna mreža su dva FC sloja koja računaju krajnju poziciju svih točaka lica [27].

6. Audio2Face aplikacija

Audio2Face (krat. A2F) je aplikacija izrađena od strane korporacije *Nvidia* kao dio kolaboracijske platforme *Omniverse.* Aplikacija je prvenstveno odabrana jer omogućuje interaktivnu animaciju u stvarnom vremenu. Osim toga, primjenjiva je za širok spektar jezika s time da je aplikacija u razvoju te se očekuje da će eventualno raditi za svaki jezik. Aplikacija dolazi s prethodno postavljenim likovima koji imaju svoj kontrole za pokretanje (eng. *rig*) koje se jednostavno mogu primijeniti na bilo koji 3D model, uključujući i neljudska lica. Uz animaciju lica vođenu govorom, korisniku je omogućeno i ručno upravljanje emocijama i njihovim intenzitetom. Naposljetku, moguće ju ostvariti prijenos animacije uživo što znači da ju je moguće primijeniti na vanjske softvere i programe [28]. U nastavku će se opisati mogućnosti i postupak korištenja A2F te primjena aplikacije za animaciju unutar softverske platforme *Unreal Engine.*

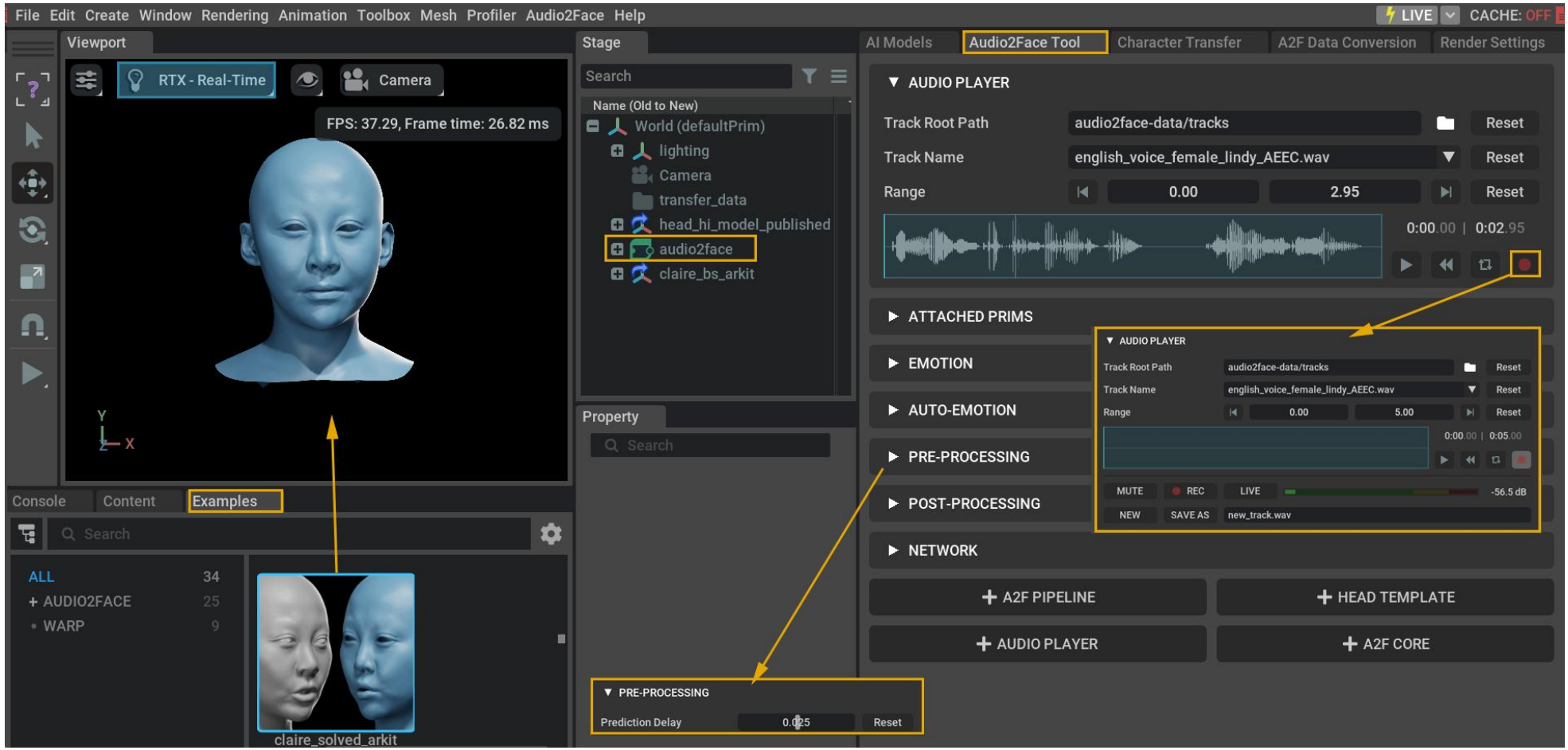
6.1 Sučelje i pokretanje animacije

Otvaranjem A2F aplikacije upali se sučelje kao na Slika 1. U radni prostor *Viewport* je potrebno unijeti model koji je unaprijed postavljen od strane kreatora. Naime, pokretanjem animacije see animira 3D model vođen od strane aplikacije. Kako bi se ta animacija mogla prenositi na druge modele potrebna je izrada *blendshape* modela koji opisuje kretanje 3D modela kao kombinaciju unaprijed definiranih pokreta. Na primjer, spajanje obrva može biti dio animacije i za ljutnju i za zbunjenost. Ukratko, *blendshape* model može generirati izraz lica linearnom kombinacijom *blendshape target-a.* Kombiniranjem više *blendshape target-a* i jačine njihovih intenziteta mogu se postići kompleksniji izrazi lica [29].

U sklopu *Examples* gotovih primjera, ponuđene su dva primjera modela i pripadajućeg *blend shape* modela vođenog A2F modelom: *claire_solved_arkit* i *mark_solved_arkit.* U radni prostor se dodao *Claire* model gdje je *blendshape* model plave boje, a A2F skriveni model je sive boje. U *Stage* panelu s desne strane *viewport-a* su vidljiva oba modela te ostali elementi scene kao što su osvjetljenje i kamera. Osim nabrojanih elementa, nalazi i se *Audio2Face Omnigraph*. *Omnigraph* je vizualni programski jezik koji omogućava korisnicima kodiranje raznovrsnih stvari pomoću blokova ili čvorova (eng. *nodes*).

Pored panela nalaze se alati aplikacije za različite vrsta upravljanja A2F podacima. U sklopu ovog rada potreban je *Audio2Face Tool* modul kojim se upravljaju pojedinosti animacije kao što su zvuk, parametri, emocije, procesiranje i slični. Svi parametri, uključujući emocije, se mogu namještati klizačima. S *Audio Player* kontrolerom je moguće namještati postavke zvuka kao što je dodavanje zvučnih datoteka i biranje glasa. No, moguće je i snimati glas s mikrofona klikom na crveni gumb za snimanje. Njime se otvaraju nove mogućnosti, kao prijenos zvuka uživo s gumbom *Live* te spremanje snimljenog zvuka. U slučaju *Live* opcije snimanja zvuka, lice se automatski animira sa ulazom iz mikrofona. Za potrebe ovog rada, potrebno je uživo prenositi animaciju, stoga nema potrebe za snimanjem i izvozom animacije. Dodatno, moguće je smanjiti vrijeme predikcije unutar kontrola za *Pre-processing* čime se postiže bolja sinkronizacija glasa i animacije s nešto manjom preciznošću.

Naposljetku je model potrebno pripremiti za prijenos animacije uživo za koji aplikacija ima ugrađeni *LiveLink* plugin (priključak) kojim se to omogućuje. Klikom na *audio2face Omnigraph* u *stage* panelu potrebno je odabrati *StreamLivelink* čvor te omogućiti opciju *Activate* (i po potrebi *Enable Audio Stream*). Time je model spreman za primjenu u vanjskim programima.



Slika 40 Sučelje aplikacije Audio2Face

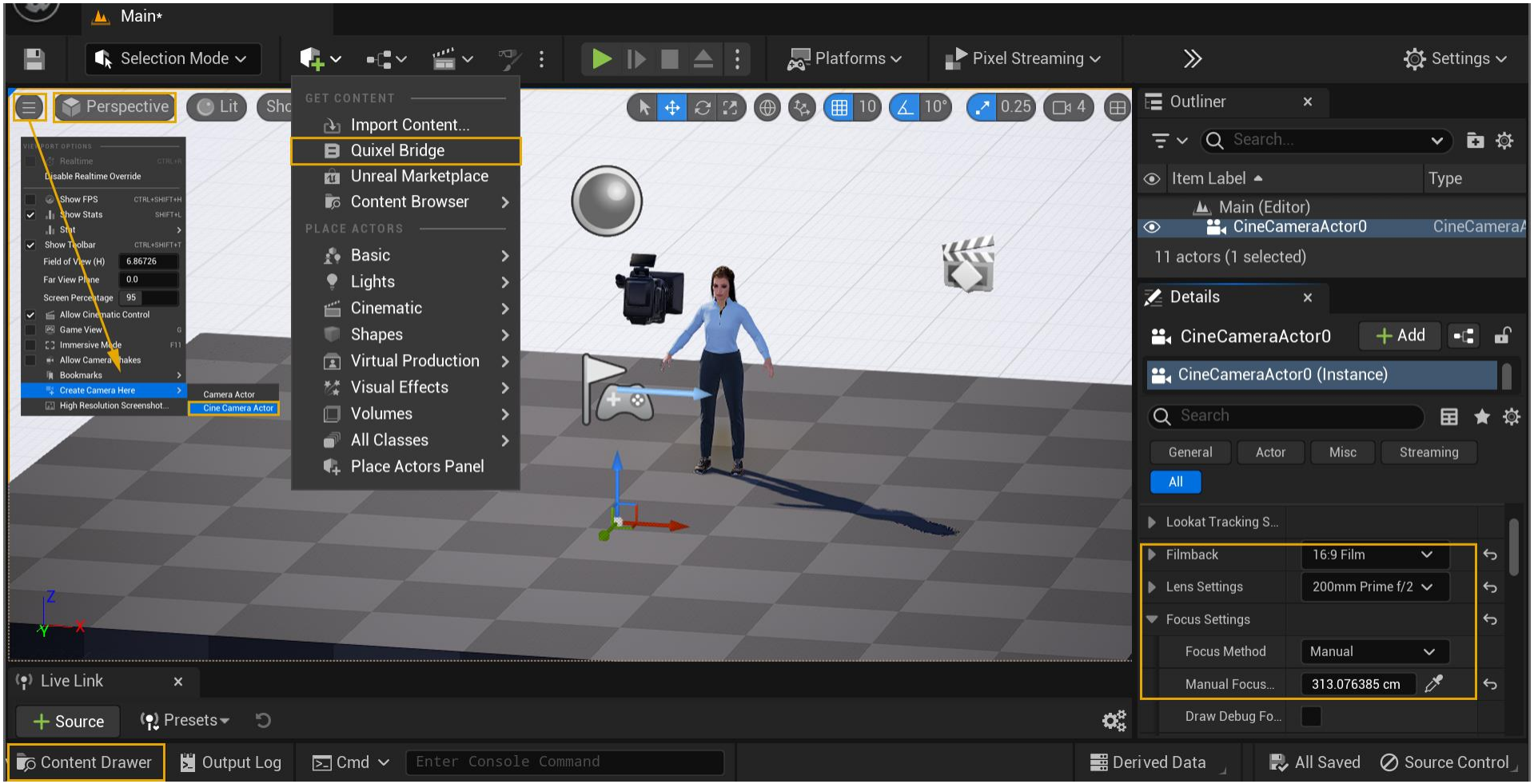
6.2 Unreal Engine

Unreal Engine (krat. UE) je moćan i sveobuhvatan alat za razvoj igara i virtualnih scena. UE može biti korišten za simulacije, vizualizaciju podataka te stvaranje interaktivnih i realističnih virtualnih okolina. UE je odabran zbog kvalitetnih animacija i performansi te zbog već postojeće povezanosti s alatima Omniverse-a. Prije početka rada je potrebno instalirati *Omniverse Livelink* plugin u UE tako što se mapa s navedenim plugin-om, iz direktorija gdje je instaliran A2F, kopira u mapu gdje je instaliran UE (*Engine* > *Plugins*). Navedeni plugin je potrebno dodatno aktivirati unutar UE zajedno s *Live Link, Live Link Control Rig, Live Link Curve Debug UI* i *LiveLink Camera* plugin-ovima.

6.2.1 Priprema radnog prostora

Model lica koji će se koristiti u sklopu UE je izrađen pomoću *MetaHuman Creator* alata. *MetaHuman* (krat. MH) služi za izradu visokokvalitetnih modela ljudi namijenjenih za uporabu unutar UE. Sama izrada lika je iznimno prilagodljiv proces gdje je moguće namještati detalje poput trepavica i boju očiju. Izrađeni lik također dolazi s namještenim kontrolama za animaciju koje se poklapaju s kontrolama unutar A2F. Time se zaobilazi postupak dodavanja kontrola pomoću drugih softvera kao što su *Maya* ili *Blender*.

Pokretanjem novog projekta unutar UE, radna okolina sadrži samo pod. Prema Slika 41 se uvodi MH u radni prostor. Prvo se skine stvoreni lik koristeći *Quixel Bridge*, izrađeni MH je time spremljen u datoteku gdje je spremljen projekt. Otvaranjem *Content Drawer* je moguće dodavati objekte u radni prostor. MH se može jednostavno „povući“ i pozicionirati.



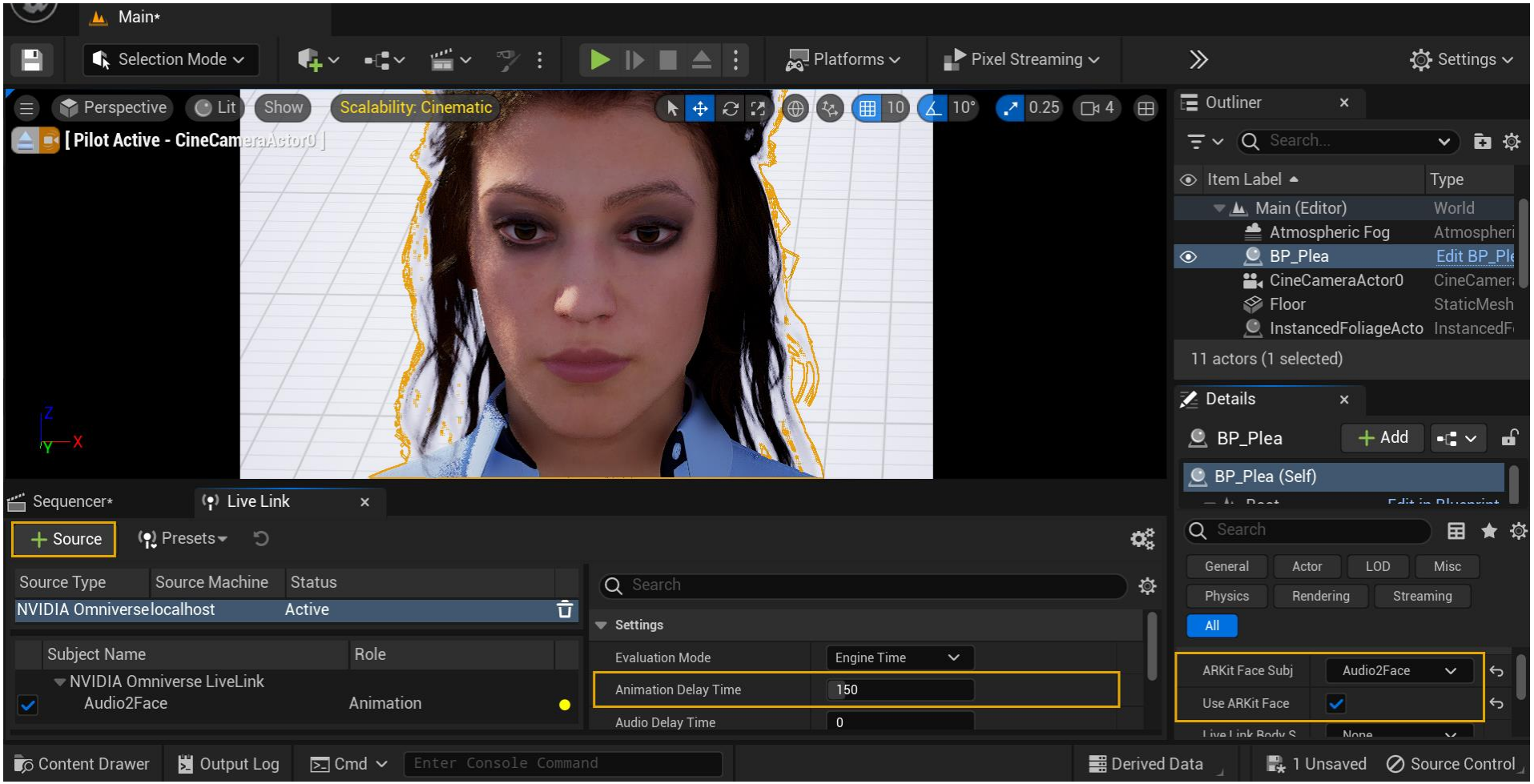
Slika 41 Učitavanje MH u radni prostor i namještanje kamere

Približavanjem na lice MH, pogled je malo izobličen te se detalji lica ne vide. Kako bi se to popravilo, potrebno je dodati kameru na mjestu koje približno gleda u portret lika. Postavlja se *Cine Camera Actor* kamera te se *Perspective* namješta na dodanu kameru. Parametri kamere, poput izbora uvećanja, leća i namještanja fokusa, se mogu namještati u *Details* panelu s desne strane.

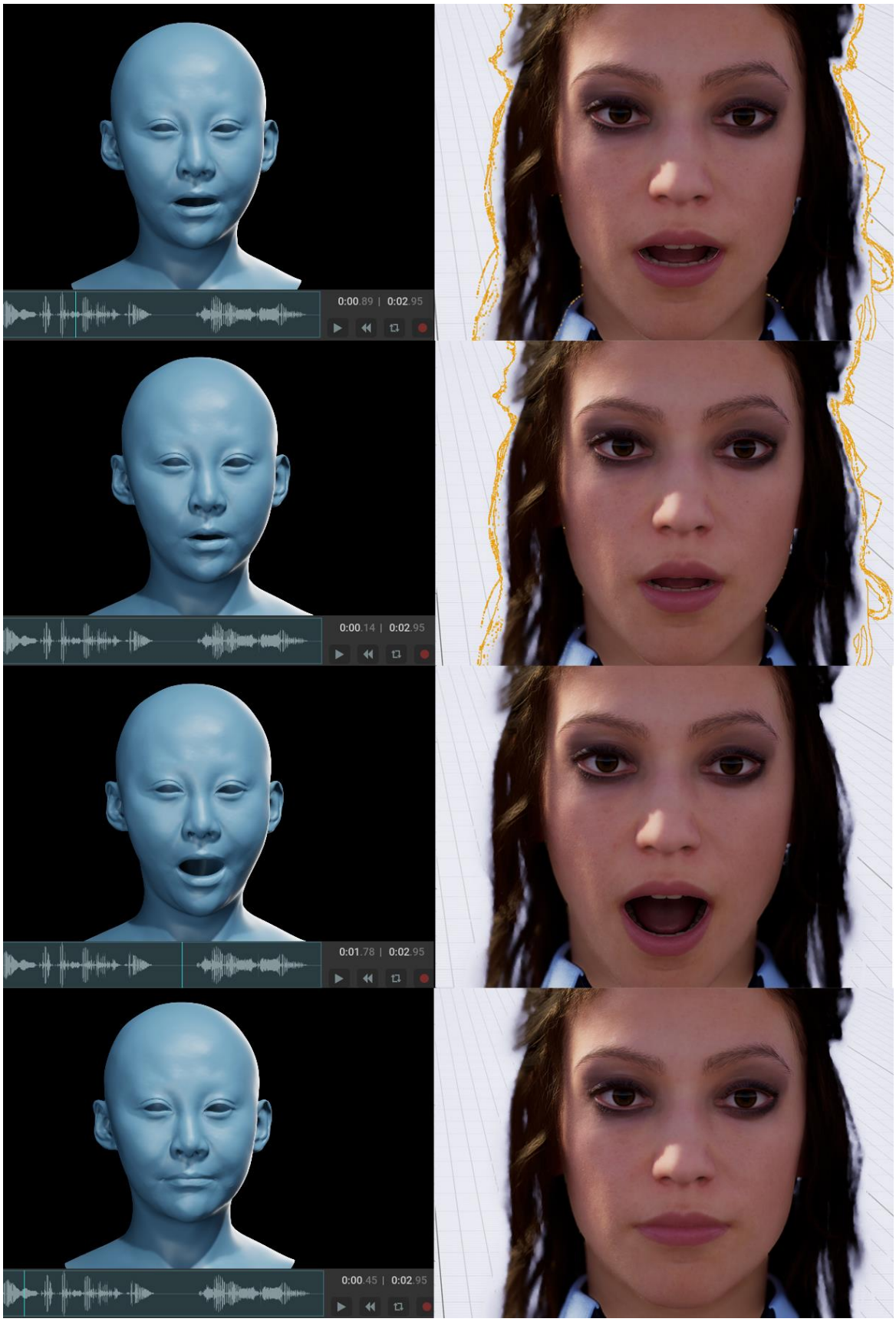
6.2.2 Prijenos animacije iz Audio2Face

Nakon što se namjesti adekvatan pogled, potrebno je povezati UE s A2F preko *Livelink-a*. Prvo se mora omogućiti *Livelink* modul preko *Window > Virutal Production > Live Link.* U njemu se zatim bira izvor prijenosa, odnosno *NVIDIA Omniverse Livelink* za koji se bira odgovarajuća frekvencija uzorkovanja. Većina zvučnih datoteka, uključujući zvuk mikrofona, odgovara frekvenciji od $44{,}1\ kHz$. Odabirom izvora otvara se opcija namještanja *Animation Delay Time* kojim se može upravljati kašnjenje animacije u milisekundama. Kašnjenje je postavljeno na nulu. Naposljetku je potrebno uskladiti animaciju iz A2F s MH modelom na način da se označi model te u panelu *Details* s desne strane odabere *ARKit Face Subject: Audio2Face* i aktivira *Use ARKit Face.* Time se *blendshape* promjene iz A2F usklađuju s *blednshape* promjenama na željenom liku unutar UE. Namještene postavke zajedno s namještenom kamerom se mogu vidjeti na Slika 42.

Pokretanjem zvuka i promjenom ostalih parametara (kao što su emocije) unutar A2F, animacija će se analogno preslikavati na model u UE. Usporedni rad UE i A2F su vidljivi na Slika 43.



Slika 42 Povezivanje UE s A2F pomoću livelink plugin-a

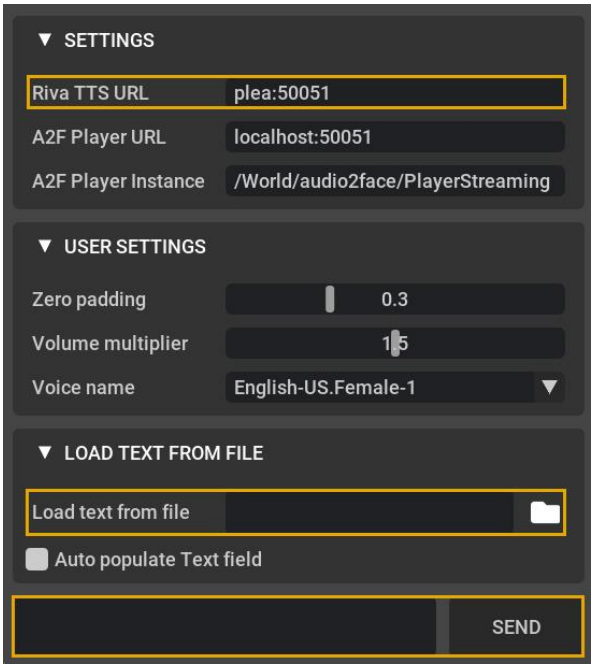


Slika 43 Prikaz sinkroniziranih animacija za različite dijelove zvuka

6.3 Ekstenzija TTS Riva

U sklopu A2F aplikacije, kreatori su omogućili raznovrsniji i podrobniji rad dodavanjem ekstenzija. Postoji niz različitih ekstenzija koje omogućavaju ili lakši i vizualno pristupačniji rad ili alate s kojima A2F dobiva novu dimenziju. U ovom radu se istražila ekstenzija *TTS (Text To Speech) Riva* kojom je omogućeno korisniku da, umjesto zvučnih datoteka, upisuje tekst koji se transformira u glas. Transformirani glas zatim služi za animaciju lica.

Ekstenzija se aktivira u glavnoj upravljačkoj traci pomoću *Window* > *Exstensions* gdje se pretražuju sve ekstenzije. Pronalaskom *TTS Riva,* ekstenziju je potrebno aktivirati te se otvara modul (Slika 44) za njeno upravljanje. Korisnik zatim upisuje server i ekstenzija je spremna za rad. Omogućeno je dodavanje cijelih datoteka s tekstom ili jednostavno upisivanje teksta. Ekstenzija je odličan alat za situacije gdje korisnik: nema mikrofon, nalazi se u glasnom okružju, želi koristiti prevedeni tekst i slične.



Slika 44 Modul za upravljanje TTS Riva ekstenzijom

7. Zaključak

U ovom radu istražena je problematika i osnovni principi izrade modela koji su sposobni generirati animaciju lica vođenu zvukom.

Pravilna ekstrakcija bitnih atributa omogućuje brži, jednostavniji i precizniji rad metoda dubokog učenja. Naime, zvučni signal je kompleksan podatak koji se može opisati s velikim brojem atributa stoga se stavlja poseban naglasak na izbor odgovarajuće reprezentacije. S druge strane, promjene izraza lica su također uvjetovane velikim brojem parametara te autentična reprezentacija ljudskog lica zahtijeva ujedno i pravilno mapiranje točaka i raspoznavanje emocija.

Pravilna obrada podataka je temelj daljnjeg razvoja modela. U radu su obrađene najčešće korištene neuronske mreže u kontekstu zadatka, no uspješan model najčešće zahtijeva kompleksnije arhitekture koje se postižu kombiniranjem različitih mreža i dodavanjem više modaliteta. Kvaliteta krajnjeg modela dakle ovisi kvaliteti podataka na kojima se trenira, o arhitekturi ukupnog modela te o pravilnom izboru reprezentacije podataka. Iz pregleda aktualnih istraživanja se moglo uočiti kako, unatoč istom cilju, modeli znatno variraju. Stoga se može zaključiti kako u tom području postoji puno prostora za daljnji razvoj i dublja istraživanja.

Jedan uspješan primjer takvog modela predstavlja Audio2Face aplikacija koja ima mogućnost generiranja lica iz zvuka u stvarnom vremenu. Osim toga, u praktičnom dijelu rada se jednostavnim postupkom ostvario prijenos animacije uživo u Unreal Engine s time da je kašnjenje animacije bilo minimalno. Aplikacija također nudi niz dodatnih alata i podešavanja kao što je ručno namještanje i automatsko prepoznavanje emocija. Iako je aplikacija u razvoju, intuitivno sučelje i jednostavnost korištenja ju čine izuzetnim postignućem u području generiranja lica vođeno zvukom.

8. Literatura

[1] W. Ertel, Introduction to Artificial Intelligence, eBook: Springer Cham, 2018.

[2] K. P. Murphy, Machine Learning: A Probabilistic Perspective, Cambridge: The MIT Press, 2012.

[3] I. Goodfellow, Y. Bengio i A. Courville, Deep Learning, Springer, 2016.

[4] A. Pouliakis, E. Karakitsou, N. Margari i P. Bountris, »Artificial Neural Networks as Decision Support Tools in Cytopathology: Past Present and Future,« *Biomedical Engineering and Computational Biology,* svez. 7, pp. 1-16, 2016.

[5] D. Yu i L. Deng, Automatic Speech Recognition: A Deep Learning Approach, London: Springer, 2015.

[6] L. R. Rabiner i R. W. Schafer, Theory and Applications of Digital Speech Processing, Upper Saddle River: Pearson Higher Education, Inc., 2011.

[7] D. Creasey, Audio Processes: Musical Analysis, Modification, Synthesis, and Control, New York: Routledge, 2017.

[8] T. Bäckström, O. Räsänen, A. Zewoudie, P. Pérez Zarazaga, L. Koivusalo, S. Das, E. Gómez Mellado, M. Bouafif Mansali i D. Ramos, Introduction to Speech Processing, Aalto University, 2022.

[9] J.-P. Thiran, F. Marques i H. Bourlard, Multimodal Signal Processing: Theory and Applications for Human-Computer Interaction, Oxford: Elsevier, 2010.

[10] I. Tokuda, »The Source–Filter Theory of Speech,« u *Oxford Research Encyclopedia of Linguistics*, Oxford University Press, 2021.

[11] X. Huang, A. Acero i H.-W. Hon, Spoken Language Processing: A Guide to Theory, Algorithm, and System Development, New Jersey: Prentice Hall, 2001.

[12] N. Sharma, »What, how, and why of MFCCs,« COSWARA, 20 Kolovoz 2020. [Mrežno]. Available: https://iiscleap.github.io/coswara-blog/coswara/tutorial/2020/08/20/mfcc.html. [Pokušaj pristupa Listopad 2023].

[13] J. Lyons, »Mel Frequency Cepstral Coefficient (MFCC) tutorial,« Practical Cryptography, 2012. [Mrežno]. Available: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/. [Pokušaj pristupa 30 Listopad 2023].

[14] C. E. Vigliante, »Anatomy and functions of the muscles of facial expression,« *Oral and Maxillofacial Surgery Clinics of North America,* svez. 17, br. 1, pp. 1-15, 2005.

[15] Y. Liu, J. Zhou, X. Li, X. Zhang i G. Zhao, »Graph-based Facial Affect Analysis: A Review of Methods, Applications and Challenges,« *JOURNAL OF LATEX CLASS FILES,* svez. 14, br. 8, pp. 1-23, 2015.

[16] C. Sagonas, . E. Antonakos , G. Tzimiropoulos, S. Zafeiriou i M. Pantic, »300 Faces In-The-Wild Challenge: database and results,« *Image and Vision Computing,* svez. 47, pp. 3-18, 2016.

[17] S. Asadiabadi, R. Sadiq i E. Erzin, »Multimodal Speech Driven Facial Shape Animation Using Deep Neural Networks,« u *Asia-Pacific Signal and*

Information Processing Association Annual Summit and Conference, Honolulu, 2018.

[18] G. Chen, »A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation,« 2016.

[19] C. C. Aggarwal, Neural Networks and Deep Learning, Springer, 2018.

[20] C. Olah, »Understanding LSTM Networks,« 27 Kolovoz 2015. [Mrežno]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Pokušaj pristupa 15 Studeni 2023].

[21] V. Dumoulin i F. Visin, »A guide to convolution arithmetic for deep learning,« 23 Ožujak 2016. [Mrežno]. Available: https://arxiv.org/abs/1603.07285. [Pokušaj pristupa 11 Studeni 2023].

[22] M. Kahng , N. Thorat, D. H. Chau, . F. B. Viegas i M. Wattenberg, »GAN Lab: Understanding Complex Deep Generative Models using,« *IEEE Transactions on Visualization and Computer Graphics,* svez. 25, br. 1, Siječanj 2018.

[23] T. Baltrusaitis, C. Ahuja i L.-P. Morency, »Multimodal Machine Learning: A Survey and Taxonomy,« *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* svez. 41, br. 2, pp. 423-443, 2019.

[24] B. Baheti, S. Gajre i S. Talbar, »Semantic Scene Understanding in Unstructured Environment with Deep Convolutional Neural Network,« u *TENCON 2019*, 2019.

[25] Y. Zhou, X. Han, E. Shechtman, J. Echevarria, E. Kalogerakis i D. Li, »MakeltTalk: speaker-aware talking-head animation,« *ACM Transactions on Graphics,* svez. 39, br. 6, pp. 1-15, 2020.

[26] J. Thies, M. Elgharib, A. Tewari, C. Theobalt i M. Nießner, »Neural Voice Puppetry: Audio-driven Facial Reenactment,« 2020.

[27] T. Karras, T. Aila, S. Laine, A. Herva i J. Lehtinen, »Audio-Driven Facial Animation by Joint End-to-End Learning of Pose and Emotion,« *Association for Computing Machinery,* svez. 36, br. 4, 2017.

[28] »Omniverse Audio2Face,« Nvidia, [Mrežno]. Available: https://www.nvidia.com/en-us/omniverse/apps/audio2face/. [Pokušaj pristupa 15 Studeni 2023].

[29] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. Pighin i Z. Deng, »Practice and Theory of Blendshape Facial Models,« u *The Eurographics Association*, 2014.

[30] B. Logan, »Mel Frequency Cepstral Coefficients for Music Modeling,« u *International Society for Music Information Retrieval Conference*, 2000.

[31] R. Yi, Z. Ye, J. Zhang, H. Bao i Y.-J. Liu, »Audio-driven Talking Face Video Generation with Learning-based Personalized Head Pose,« *arxiv,* 2020.