Integracija modela za generiranje glasa iz teksta i modela za generiranje slike govornika za realistično stvaranje virtualnih likova

Drobnjak, Antun

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 439

INTEGRACIJA MODELA ZA GENERIRANJE GLASA IZ TEKSTA I MODELA ZA GENERIRANJE SLIKE GOVORNIKA ZA REALISTIČNO STVARANJE VIRTUALNIH LIKOVA

Antun Drobnjak

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 439

INTEGRACIJA MODELA ZA GENERIRANJE GLASA IZ TEKSTA I MODELA ZA GENERIRANJE SLIKE GOVORNIKA ZA REALISTIČNO STVARANJE VIRTUALNIH LIKOVA

Antun Drobnjak

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 439

Pristupnik:	**Antun Drobnjak (0036525726)**
Studij:	Računarstvo
Profil:	Programsko inženjerstvo i informacijski sustavi
Mentor:	prof. dr. sc. Ivica Botički

Zadatak: **Integracija modela za generiranje glasa iz teksta i modela za generiranje slike govornika za realistično stvaranje virtualnih likova**

Opis zadatka:

U okviru ovog diplomskog rada potrebno je istražiti model za generiranje glasa iz teksta (TTS) te model za generiranje slike govornika. Fokus će biti na razvoju efikasnog sustava koji može generirati autentičan govor i povezane vizualne elemente na temelju unaprijed definiranih tekstualnih ulaza. Metodologija rada obuhvatit će proučavanje novih tehnologija u području TTS-a i generativnih modela slika govornika. Cilj je istražiti mogućnosti integracije ova dva procesa kako bi se postigao koherentan dojam virtualnog govornika. Implementacija će se izvršiti pomoću programskog jezika Python, koristeći relevantne knjižnice za obradu teksta, generiranje glasa i stvaranje slika. Rad će istražiti i mogućnosti primjene ovakvih modela u stvarnom vremenu te razmotriti potencijalne izazove i koristi koje proizlaze iz integracije TTS-a i generiranja slika realističnih virtualnih govornika.

Rok za predaju rada: 28. lipnja 2024.

Sadržaj

Uvod 1

1. Teorijska osnova 3

2. Generiranje audio zapisa 6

2.1. Povijesni pregled 6

2.2. Odabir TTS modela 7

2.2.1. Pregled TTS modela 7

2.2.2. Zašto GlowTTS model? 7

2.3. Treniranje modela 8

2.3.1. Podaci za treniranje 8

2.3.2. Proces treniranja 9

2.4. Rezultati treniranja 12

2.4.1. GlowTTS model za hrvatski jezik 12

2.4.2. GlowTTS model za engleski jezik 13

2.5. Zaključci treniranja GlowTTS modela 14

2.6. Korištenje predtreniranog TTS modela 15

2.6.1. Glow-TTS 15

2.6.2. Your-TTS 16

3. Generiranje videozapisa 18

3.1. Povijesni pregled 18

3.2. Moderne metode 19

3.2.1. LipGAN 19

3.2.2. AD-NeRF 19

3.2.3. MakeItTalk 19

3.2.4. Wav2Lip 20

3.3. Komparativna analiza metoda generiranja videa govornika 20

3.4. MakeItTalk 21
4. Arhitektura sustava 23
4.1. Inicijalizacija aplikacije 24
4.2. Interakcija s aplikacijom 26
5. Korištenje aplikacije 31
5.1. Inicijalizacija aplikacije 31
5.2. Interakcija s aplikacijom 32
6. Performanse sustava 34
6.1. Rezultati testiranja 34
Zaključak 38
Literatura 39
Sažetak 42
Summary 43
Skraćenice 44

Uvod

U današnjem digitalnom dobu, tehnologija se razvija nevjerojatnom brzinom, otvarajući nove mogućnosti za unaprjeđenje svakodnevnog života. Razvoj umjetne inteligencije (AI, eng. Artificial Intelligence) donio je mnoge promjene, transformirajući razne industrije i unaprjeđujući svakodnevni život kroz poboljšanju automatizaciju, analizu podataka i personalizirane usluge. Jedan od revolucionarnih napredaka je mogućnost generiranja videozapisa osobe koja govori, sa zvukom, koristeći tekst i fotografiju osobe. Ova tehnologija omogućava kreiranje virtualnih likova koji imaju sposobnost unaprijediti virtualnu komunikaciju i doprinijeti razvoju različitih područja kao što su obrazovanje, medicina, turizam, marketing.

Jedno od obećavajućih područja primjene virtualnih likova je obrazovni sektor. Udaljeno i online učenje, osobito istaknuti tijekom pandemije COVID-19, pokazali su nedostatak zastarjelih školskih modela koji nisu dovoljno fleksibilni za brze promjene u društvu i tehnologiji. Razvoj umjetne inteligencije doprinosi mogućnosti personalizacije obrazovanja koja se smatra ključnim alatom za unaprjeđenje školskog sustava. Virtualni likovi mogu doprinijeti personalizaciji obrazovanja kroz interaktivnu komunikaciju s učenicima, prilagođavajući se njihovim individualnim potrebama, preferencijama i stilovima učenja. Virtualni likovi imaju potencijal značajno unaprijediti medicinsku skrb i podršku osobama s poteškoćama. Mogu služiti kao virtualni asistenti koji pružaju kontinuiranu podršku pacijentima s kroničnim bolestima, podsjećajući ih na uzimanje lijekova i pomažući im u praćenju simptoma. Općenito u virtualnom svijetu, ova tehnologija može poboljšati iskustva korisnika pružajući bogatije, dinamičnije i autentičnije sadržaje.

Motivacija za ovaj rad leži u istraživanju širokog spektra mogućnosti koje nudi generiranje audio zapisa i videozapisa. Fokus je na razumijevanju kako ova tehnologija može biti integrirana u različite sektore te identificiranju njenih prednosti i ograničenja.

Cilj ovog rada je proučiti različite metode i tehnike generiranja zvuka i slike, istražiti trenutne mogućnosti u području tehnologije generiranja sadržaja, analizirati postojeće modele i njihove performanse te identificirati potrebne resurse za uspješnu integraciju.

Rad započinje s teorijskom osnovom koja obuhvaća pregled relevantne literature s naglaskom na postojeće mogućnosti u generiranju audio zapisa i videozapisa. Nakon

teorijskog pregleda, opisuje se proces odabira odgovarajućih modela, uz analizu njihovih karakteristika i mogućnosti. Nadalje, rad se nastavlja s opisom arhitekture sustava i evaluacijom performansi sustava, analizirajući brzinu i kvalitetu generiranog sadržaja.

1. Teorijska osnova

Tehnologije za generiranje videozapisa i audio zapisa napravile su značajan korak u proteklih nekoliko desetljeća, transformirajući velik broj područja kao što su obrazovanje i medicina. Ove tehnologije koriste napredne tehnike strojnog učenje kako bi ostvarile rezultate slične pravom ljudskom glasu i izgledu lica. Povijest krosmodalne sinteze (eng. cross-modal synthesis) u računalnoj grafici seže više od dva desetljeća. U svom radu Voice Pupperty, Brand [1] je generirao potpunu animaciju lica iz audio zapisa, postavljajući tako temelje za kasniji razvoj. Od tada su razvijene razne tehnike za sinkronizaciju audio ulaza s vizualnim izlazima, kao što je generiranje plesa vođeno glazbom [2] i odvajanje izvora zvuka sa sinkroniziranom glazbenom generacijom [3].

Razvojem strojnog učenja (ML, eng. Machine Learning) i dubokog učenja (DL, eng. Deep Learning) dolazi do unaprjeđenja tehnologija za generiranje audio zapisa i videozapisa. Veliku ulogu u tom razvoju imaju rekurentne neuronske mreže (RNN, eng. Recurrent Neural Networks), mreže dugog kratkoročnog pamćenja (LSTM, eng. Long Short-Term Memory) i konvolucijske neuronske mreže (CNN, eng. Convolutional Neural Networks) dok su generativne suparničke mreže (GAN, eng. Generative Adversarial Networks) i varijacijski autokoderi (VAE, eng. Variational Autoencoders) dodatno pomaknuli granice onoga što ove tehnologije mogu ostvariti.

RNN-ovi su dizajnirani za rukovanje sekvencijalnim podacima, što ih čini ključnim za zadatke u kojima je bitan redoslijed unosa, kao što je generiranje zvuka i slike. Za generiranje zvuka, RNN-ovi mogu modelirati vremensku dinamiku zvučnih valova, čineći ih učinkovitima za generiranje koherentnih audio sekvenci iz teksta. U generiranju videa, RNN-ovi mogu pomoći u stvaranju nizova okvira koji imaju vremenski kontinuitet. LSTM-ovi, posebna vrsta RNN-a, mogu obuhvatiti dugoročne ovisnosti rješavanjem problema nestajanja gradijenta. Zbog toga su LSTM-ovi posebno korisni za generiranje zvuka jer mogu održavati kontekst tijekom duljeg trajanja, osiguravajući da su generirani audio segmenti kontekstualno dosljedni. Za generiranje videa, LSTM-ovi mogu pomoći u održavanju vremenske koherencije kroz dulje nizove okvira [4], [5].

CNN-ovi su vrlo učinkoviti za zadatke koji uključuju prostorne podatke, kao što su slike i video okviri, zbog njihove sposobnosti hvatanja prostornih hijerarhija [6]. U generiranju slika i videa, CNN-ovi mogu stvoriti visokokvalitetne okvire učenjem prostornih značajki.

Često se koriste za generiranje pojedinačnih okvira u videu ili poboljšanje detalja slike na način kadar po kadar.

GAN-ovi se sastoje od dvije neuronske mreže (generator i diskriminator) koje rade u tandemu kako bi generirale realne podatke [7]. Bili su revolucionarni u generiranju vrlo realističnih slika i videa obučavanjem generatora da proizvodi podatke koje diskriminator ne može razlikovati od stvarnih podataka. Ovaj suparnički proces rezultira visokokvalitetnim, realističnim rezultatima [5]. GAN-ovi se također mogu prilagoditi za generiranje zvuka, proizvodeći realne audio uzorke uvježbavanjem audio podataka.

VAE su vrsta autoenkodera koji uče probabilističku reprezentaciju podataka, omogućujući kontroliranije i raznovrsnije generiranje [8]. Korisni su za generiranje novih slika i video okvira uzorkovanjem iz naučenog latentnog prostora, omogućujući stvaranje raznolikih i realističnih izlaza. U audio generaciji, VAE se mogu koristiti za sintezu novih, realističnih audio uzoraka [9].

Najnoviji modeli pretvaranja teksta u govor (TTS, eng. Text-to-Speech), uključujući Tacotron 2 [10], Deep Voice 3 [11] i Transformer TTS [12], predstavljaju značajan napredak u sintetiziranju visokokvalitetnih mel-spektrograma izravno iz tekstualnih ulaza. Ovi modeli omogućuju generiranje prirodnog govora s poboljšanom točnošću i izražajnošću. Generativni modeli koji se temelje na protoku kao što je Glow-TTS [13] dodatno su poboljšali mogućnosti TTS-a sintetizirajući govor paralelno, nudeći i visoku kvalitetu i učinkovitost u stvaranju mel-spektrograma.

Paralelno s tim, tehnologije generiranja glasa i slike, osobito u kontekstu zvučno vođene animacije lica, doživjele su značajan razvoj. Zvučno vođena animacija lica uključuje generiranje sinkroniziranih pokreta lica iz audio ulaza. Ovo je područje doživjelo značajan napredak, posebice upotrebom neuronskih mreža. Eskimez i sur. [14], [15] razvili su model za generiranje sinkroniziranih orijentira lica s robusnom otpornošću na buku. Chen i sur. [16] predložili su metodu koja prvo generira orijentire, a zatim proizvodi rasterizirane videozapise, koristeći maske pozornosti kako bi se usredotočili na dinamične dijelove lica, a posebice usne. Greenwood i sur. [17] koristili su dvosmjernu LSTM mrežu za zajedničko učenje izraza lica i položaja glave. Yang i sur. [5] nastoje riješiti problem izostavljanja identiteta i stila govornika te razdvajaju audio sadržaj i informacije o govorniku te bilježe karakteristike koje hvataju dinamiku ovisnu o govorniku za stvarnije animacije lica.

Virtualna komunikacija postala je sve popularnija, osobito s porastom rada na daljinu, online obrazovanja i digitalnih društvenih interakcija. Sposobnost stvaranja realističnih virtualnih likova koji se mogu uključiti u realistične razgovore nudi značajne prednosti u ovim područjima. Ti se virtualni likovi mogu koristiti u raznim aplikacijama, uključujući virtualne asistente, službu za korisnike, terapiju i zabavu. Realistični virtualni likovi povećavaju angažman korisnika i pružaju interaktivnije iskustvo, premošćujući jaz između ljudskih i digitalnih interakcija [18]. Tvrtke kao što su Soul Machines [19] i Replika [20] već koriste ove tehnologije za stvaranje realističnih avatara i agenata za razgovor. Kombinacijom naprednih tehnika u stvaranju glasa i slike, moguće je stvoriti stvarne virtualne likove koji mogu sudjelovati u virtualnoj komunikaciji, čineći interakciju prirodnijom i učinkovitijom. Kontinuirani razvoj ovih tehnologija obećava revoluciju u načinu na koji komuniciramo u virtualnim prostorima, utirući put za sofisticiranije digitalne entitete nalik ljudima.

2. Generiranje audio zapisa

2.1. Povijesni pregled

Tehnologija pretvaranja teksta u govor značajno se razvila od svog početka u istraživanju računalne znanosti sredinom 20. stoljeća. Rani napori bili su usmjereni na osnovne metode sinteze govora, kao što je sustav „Pattern playback“ Franklina S. Coopera [21] u kasnim 1940-ima, koji je generirao zvukove slične govoru koristeći akustične obrasce, ali je pokazao ograničen vokabular i robotsku notaciju. Napredak u narednim desetljećima, vođen napretkom u digitalnoj obradi signala (DSP, eng. Digital Signal Processing) i računalne lingvistike, doveo je do razvoja sustava temeljenih na pravilima u 1970-ima, poboljšavajući strukturiranu sintezu teksta u govor, iako uz primjetnu umjetnost.

Kasno 20. stoljeće svjedočilo je usponu statističkih parametarskih metoda sinteze (SPSS, eng. Statistical Parametric Speech Synthesis), povećavajući prirodnost i fleksibilnost sintetičkog govora generiranjem govornih parametara poput visine tona, trajanja i spektralnih značajki iz teksta za razliku od do tada tradicionalnih konkatenativnih metoda koje spajaju unaprijed snimljene segmente govora. Iako napredniji od tadašnjih metoda, SPSS ostaje osjetno lošiji od snimki prirodnog govora u pogledu kvalitete, prirodnosti, sličnosti govornika i razumljivosti u buci [22].

Razvojem tehnika dubokog učenja dolazi do revolucije u TTS tehnologiji. Duboko učenje omogućava modelima kao što su Tacotron [23], Tacotron 2 [24] i Transformer TTS [25] da direktno mapiraju tekstualne ulaze u mel-spektrograme ili valne oblike govora, što omogućava precizno hvatanje nijansi prirodnog govora kao što su intonacija i prozodija. Ovi napredni modeli koriste neuronske mreže za generiranje govornih segmenata, pružajući korisnicima visoko razumljive i izražajne sintetičke glasove.

Evolucija TTS tehnologije od njezinih ranih početaka, u osnovnim metodama sinteze govora, do sadašnje ere, modela temeljenih na dubokom učenju, predstavlja velik napredak u inovacijama u relativno kratkom periodu. Ovim razvojem prirodnosti, prilagodljivosti i jednostavnosti sintetičkog govora polje TTS tehnologije je spremno dodatno poboljšati interakciju između čovjeka i računala kroz različite aplikacije, od jednostavnih web-asistenata do imerzivne virtualne stvarnosti.

2.2. Odabir TTS modela

2.2.1. Pregled TTS modela

Danas su dostupni različiti TTS modeli, svaki sa svojim prednostima i manama. TTS modeli su s godinama značajno evoluirali poboljšavajući kvalitetu i prirodnost sintetiziranog govora. Među najznačajnijim modelima su Tacotron, WaveNet i GlowTTS.

Tacotron je sekvenca-do-sekvence (eng. sequence-to-sequence) model koji pretvara tekst u sekvencu mel-spektrogram okvira, koji se zatim pretvaraju u zvuk pomoću vokodera kao što je Griffin-Lim. Tacotron je poznat po proizvodnji visokokvalitetnog i razumljivog govora. Tacotron 2, poboljšana verzija, koristi WaveNet vokoder za dodatno poboljšanje kvalitete zvuka [10], [23].

WaveNet, razvijen od strane tvrtke DeepMind, je duboki generativni model valnih oblika sirovog zvuka (eng. raw sound). Generira zvuk uzorkovanjem jednog po jednog audio uzorka, ovisno o prethodnim uzorcima i drugim ulazima poput mel-spektrograma. WaveNet je poznat po svojoj sposobnosti da proizvede vrlo prirodan i realističan govor, ali njegovi računalni zahtjevi su značajni, što aplikacije u stvarnom vremenu čini izazovnim bez optimizacija.

GlowTTS je generativni model temeljen na protoku (eng. flow-based) za TTS koji koristi normalizirajuće protoke za mapiranje složenih distribucija audio valnih oblika iz jednostavnijih distribucija. Za razliku od autoregresivnih modela kao što je WaveNet, GlowTTS može generirati cijele valne oblike paralelno, omogućujući bržu sintezu. Ovaj model posebno je cijenjen zbog ravnoteže između visokokvalitetnog izlaza i učinkovite izvedbe u stvarnom vremenu.

2.2.2. Zašto GlowTTS model?

Zbog svoje sposobnosti generiranja visokokvalitetnog govora u stvarnom vremenu, odabran je GlowTTS. U nastavku su neki bitni čimbenici koji su doprinijeli toj odluci:

- **Performanse u stvarnom vremenu:** GlowTTS može paralelno generirati valne oblike govora, što ga čini znatno bržim od auto regresivnih modela poput WaveNeta. Ova sposobnost paralelnog generiranja ključna je za aplikacije koje zahtijevaju sintezu govora u stvarnom ili gotovo stvarnom vremenu.

- **Visokokvalitetni izlazi:** Arhitektura GlowTTS-a koja se temelji na protoku omogućuje produkcije prirodnog i izražajnog govora. Učinkovito hvata nijanse ljudskog govora, uključujući prozodiju i intonaciju, što rezultira ugodnim i realističnim iskustvom slušanja.
- **Lakoća treninga:** Arhitektura GlowTTS-a omogućava stabilnu i jednostavniju obuku u usporedbi sa složenijim modelima što je povoljno pri radu s različitim skupovima podataka.
- **Robusnost:** GlowTTS je poznat po svojoj robusnosti na različitim jezicima i naglascima. Sposobnost modela da dobro generalizira različite jezične značajke osigurava dosljednu izvedbu.
- **Zajednica i podrška:** Usvajanje GlowTTS-a unutar Coqui-AI TTS okvira omogućuje pristup zajednici koja pruža podršku i opsežne resurse. To uključuje unaprijed obučene modele, opsežnu dokumentaciju i aktivne forume koji olakšavaju rješavanje problema te stalno poboljšavanje modela.

Ukratko, GlowTTS je odabran zbog ravnoteže performansi u stvarnom vremenu, visokokvalitetnog izlaza, jednostavnosti obuke i robusnosti na različitim jezicima. Njegov odabir u skladu je s ciljevima razvoja učinkovitog i djelotvornog TTS sustava koji može proizvesti govor koji prirodno zvuči na različitim jezicima.

2.3. Treniranje modela

2.3.1. Podaci za treniranje

Za efektivno treniranje TTS modela potrebno je prikupiti velike količine audio zapisa i transkripata. U ovom diplomskom radu trenirana su tri modela, dva korištenjem skupova audio zapisa na hrvatskom i jedan korištenjem audio zapisa na engleskom jeziku. Svaki je model treniran zasebno kako bi se procijenila njihova individualna izvedba.

Prvi skup podataka sadrži 10.988 audio zapisa saborskih govora više govornika na hrvatskom jeziku s odgovarajućim transkriptom, preuzetih s HuggingFace platforme [26] iz skupa podataka classla/ParlaSpeech-HR [27]. Duljina pojedinog audio zapisa je između 1 i 70 sekundi dok je ukupna duljina audio zapisa 2.090 minuta i 21 sekundu što je približno 35 sati.

Drugi skup podataka sadrži 18.965 audio zapisa saborskih govora više govornika na hrvatskom jeziku s odgovarajućim transkriptom, izvorno iz Parliamentary Spoken Corpus of Croatian ParlaSpeech-HR 2.0 [28]. Duljina pojedinog audio zapisa je između 1 i 151 sekundu dok je ukupna duljina audio zapisa 4.016 minute i 7 sekundi što je približno 67 sati.

Treći skup podataka sadrži 13 000 audio zapisa jednog govornika na engleskom jeziku s odgovarajućim transkriptom, preuzetih iz LJ Speech skupa podataka [29]. Duljina pojedinog audio zapisa je između 1 i 10 sekundi dok je ukupna duljina audio zapisa 1.435 minute i 17 sekundi što je približno 24 sata.

Naziv skupa	Raspon duljina pojedinog zapisa (sekunde)	Ukupna duljina (sekunde)
classla/ParlaSpeech-HR	1 - 70	125.421
Parliamentary Spoken Corpus of Croatian ParlaSpeech-HR	1 - 151	240.967
LJ Speech	1 - 10	86.117

Audio zapisi i transkripti su korišteni u preuzetom oblike bez prethodne obrade ili normalizacije.

2.3.2. Proces treniranja

Treniranje modela GlowTTS podržano je okvirom Coqui-AI TTS [30], odabranim zbog robusnih mogućnosti u rukovanju različitim skupovima podataka koje su bile ključne za optimizaciju izvedbe modela. Coqui-AI je platforma za razvoj i implementaciju govornih tehnologija otvorenog koda, koja omogućuje istraživačima i developerima da grade i prilagođavaju modele za sintezu govora, prepoznavanje govora i druge srodne zadatke. Njegove značajke uključuju visoku prilagodljivost, podršku za različite jezike i skupove podataka, te jednostavnu integraciju s drugim alatima i bibliotekama.

Python skripta za treniranje GlowTTS-a, temeljena na Coqui-AI dokumentaciji [31] i dodatno prilagođena, izvršena je unutar prethodno konfiguriranog kontejnera pomoću Apptainera [32], inovativne platforme za upravljanje kontejnerima, na udaljenom poslužitelju.

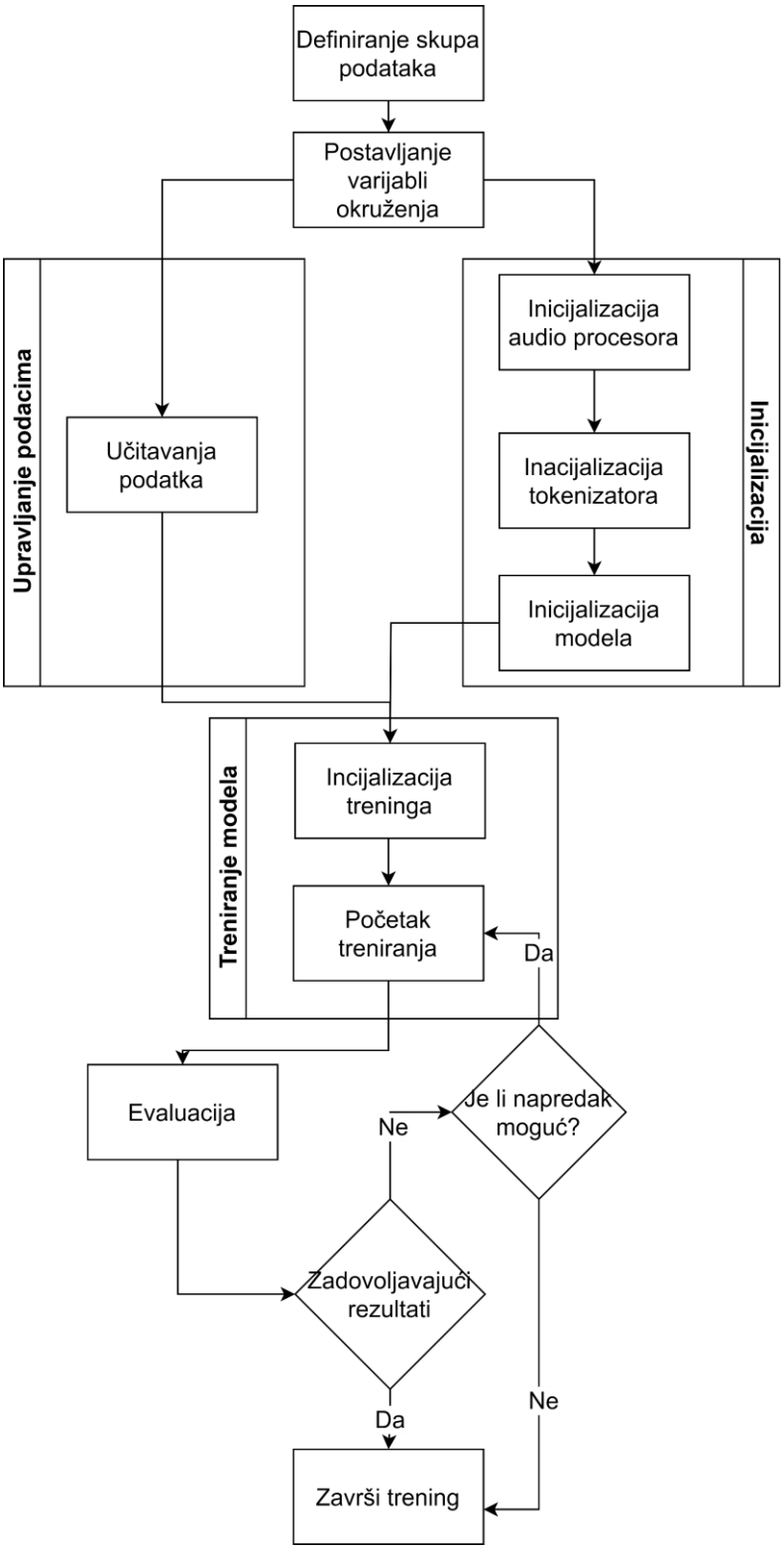
```
import ...
output_path = os.path.dirname('train/')
dataset_config = BaseDatasetConfig(
    formatter="thorsten", meta_file_train="metadata.csv",
path=os.path.join(output_path, "")
```

```
)

config = GlowTTSConfig(
    batch_size=32,
    eval_batch_size=16,
    num_loader_workers=2,
    num_eval_loader_workers=2,
    run_eval=True,
    test_delay_epochs=-1,
    epochs=3000,
    text_cleaner="phoneme_cleaners",
    use_phonemes=True,
    phoneme_language="hr",
    phoneme_cache_path=os.path.join(output_path,
"phoneme_cache"),
    print_step=25,
    print_eval=False,
    mixed_precision=True,
    output_path=output_path,
    datasets=[dataset_config],
    test_sentences=["..."])
# INITIALIZE THE AUDIO PROCESSOR
ap = AudioProcessor.init_from_config(config)
# INITIALIZE THE TOKENIZER
tokenizer, config = TTSTokenizer.init_from_config(config)
# LOAD DATA SAMPLES
train_samples, eval_samples = load_tts_samples(
    dataset_config,
    eval_split=True,
    eval_split_max_size=config.eval_split_max_size,
    eval_split_size=config.eval_split_size,)
# INITIALIZE THE MODEL
model = GlowTTS(config, ap, tokenizer, speaker_manager=None)
# INITIALIZE THE TRAINER
trainer = Trainer(
    TrainerArgs(), config, output_path, model=model,
train_samples=train_samples, eval_samples=eval_samples)
trainer.fit()
```

Kôd 2.1 Skripta za treniranje GlowTTS modela

Treniranje se odvijalo na klasteru visokih performansi opremljenim GPU akceleracijom. Ova postavka omogućila je učinkovito rukovanje računalnim zahtjevima svojstvenim obuci TTS modela velikih razmjera, osiguravajući brže vrijeme obrade i poboljšanu produktivnost.



Sl. 2.1 Dijagram tijeka treniranja GlowTTS modela

Proces treniranja GlowTTS modela uključivao je treniranje modela dok se ne dosegne vremensko ograničenje na klasteru. Trening je zatim zaustavljen, model je evaluiran, a trening je nastavljen ili zaustavljen ovisno o rezultatu evaluacije.

2.4. Rezultati treniranja

2.4.1. GlowTTS model za hrvatski jezik

U početku je GlowTTS model treniran na 150 hrvatskih uzoraka na 300 epoha. Očekivano, ovo nije dalo dobre rezultate zbog male veličine uzorka. Treniranje je zatim nastavljeno s 10.988 uzoraka hrvatskih saborskih govora, podijeljenih u nekoliko sesija opisanih u tablici (Tablica 2.1).

Tablica 2.1 Treniranje hrvatskog TTS modela na 10.988 uzoraka

Sesija	Dosegnuta epoha	Rezultati
1. Početno treniranje	1.063	Rezultati su uredni. Čuju se hrvatski jezični obrasci.
2. Nastavljeno treniranje	2.063 (1.000 u sesiji)	Rezultati ostaju gotovo isti; nije primijećeno značajno poboljšanje. Daljnje treniranje zaustavljeno zbog nedostatka značajnog poboljšanja.

Sljedeća faza uključivala je treniranje modela na 18.965 uzoraka, također iz hrvatskih saborskih govora, koje je detaljno opisani u tablici (Tablica 2.2).

Tablica 2.2 Treniranje hrvatskog TTS modela na 18.965 uzoraka

Sesija	Dosegnuta epoha	Rezultati
1. Početno treniranje	300	Rezultati nisu zadovoljavajući.
2. Nastavljeno treniranje	560 (260 u sesiji)	Rezultati ostaju nezadovoljavajući.
3. Nastavljeno treniranje	1.105 (545 u sesiji)	Rezultati ostaju nezadovoljavajući.
4. Nastavljeno treniranje	1.365 (260 u sesiji)	Uočeno blago poboljšanje. Rezultat sličan onome iz 1. koraka u Tablica 2.1.
5. Nastavljeno treniranje	1.930 (565 u sesiji)	Uočeno blago poboljšanje. Daljnje treniranje zaustavljeno zbog nedostatka značajnog poboljšanja.

Sljedeći korak bio je fino ugađanje (engl. fine tuning) već uvježbanog hrvatskog TTS modela [33]. Ovo fino ugađanje izvršeno je na bazi od 10.899 uzoraka. Opis procesa nalazi se u tablici (Tablica 2.3).

Tablica 2.3 Fino ugađanje hrvatskog TTS modela na 10 899 audio zapisa

Sesija	Dosegnuta epoha	Rezultati
1. Početno fino ugađanje	580	Uočeno poboljšanje. Bolji naglasak, ali robotska intonacija ostaje.
2. Nastavak finog ugađanja	1.580 (1.000 u sesiji)	Nema daljnjih poboljšanja. Fino ugađanje zaustavljeno.

Sesije treniranja na hrvatskom skupu podataka, koje su obuhvaćale različite pokušaje s različitim veličinama uzorka i epohama, naišle su na izazove koji prvenstveno proizlaze iz ograničenja skupa podataka. Početni pokušaji s manjim skupom podataka od 150 uzoraka, dosljedno nisu davali zadovoljavajuće rezultate. S većim skupovima podataka od 10.988 i 18.965 uzoraka, iako su uočena poboljšanja, sintetiziranom govoru još uvijek nedostaje puna reprezentativnost prirodnih hrvatskih govornih obrazaca. Ovime je uočeno da povećanjem skupa podataka postižemo bolje rezultate, ali i dalje postoje problemi, poput gubitka dijakritičkih znakova i robotske intonacije, što ukazuje na poteškoće u hvatanju nijansi jedinstvenih za hrvatski izgovor. Ovi izazovi su složeni varijacijama u govoru među različitim govornicima i mogućim razlikama u kvaliteti transkripata, što dodatno utječe na treniranje modela.

2.4.2. GlowTTS model za engleski jezik

Treniranje na engleskom skupu podataka općenito je dalo zadovoljavajuće rezultate. Ovaj se uspjeh može pripisati dostupnosti većih i raznolikijih korpusa engleskog govora kao što je skup podataka LJSpeech, koji je pružio robusnu osnovu za model za učenje i učinkovitije sintetiziranje prirodnog engleskog govora. Kontrast u ishodima obuke između hrvatskog i engleskog jezika naglašava ključnu ulogu veličine skupa podataka, raznolikosti i kvalitete audio zapisa i transkripata.

Sesija	Dosegnuta epoha	Rezultati
1. Početno fino ugađanje	1.000	Rezultati zadovoljavajući. Nastavak treniranja zbog provjere dodatnog unaprijeđena.
2. Nastavak finog ugađanja	2.000 (1.000 u sesiji)	Uočene blago poboljšanje. Zaustavljanje treniranja.

2.5. Zaključci treniranja GlowTTS modela

Na temelju rezultata treniranja GlowTTS modela, može se izvući nekoliko zaključaka u vezi s optimalnim karakteristikama skupova podataka i pojedinačnih uzoraka za učinkovito treniranje:

1. Veličina skupa podataka: Veličina skupa podataka igra ključnu ulogu u izvedbi TTS modela. Veći skupovi podataka, poput onih koji sadrže desetke tisuća uzoraka, poželjniji su jer daju raznovrsniji i sveobuhvatniji prikaz jezičnih varijacija i govornih obrazaca. Ova raznolikost pomaže u hvatanju nijansi i prirodnog tijeka govora u različitim kontekstima.
2. Raznolikost uzoraka: Postizanje optimalne ravnoteže skupa podataka za treniranje TTS modela uključuje snalaženje u prednostima i izazovima raznolikosti uzoraka. Dok različiti skupovi podataka poboljšavaju generalizaciju izlažući modele različitim obrascima govora, naglascima i stilovima, pretjerana raznolikost može zakomplicirati treniranje zbog povećane složenosti modela i potencijalne nedosljedne kvalitete podataka. Ključno je postići ravnotežu, osiguravajući da skup podataka predstavlja širok raspon jezičnih značajki i karakteristika govornika relevantnih za aplikaciju, a istovremeno održavati dosljednu kvalitetu i relevantnost podataka. Ovaj pristup ublažava pristranost, podržava učinkovitu prilagodbu modela različitim naglascima i potiče pouzdanu sintezu govora koji prirodno zvuči.
3. Kvaliteta transkripta: kvaliteta i točnost transkripta uparenih s audio uzorcima su ključni. Dobro transkribirani tekstovi osiguravaju precizno usklađivanje između unosa teksta i izlaza zvuka tijekom treninga, značajno utječući na sposobnost modela da nauči točan izgovor i obrasce intonacije, čime se poboljšava kvaliteta sintetiziranog govora. Konzistentna anotacija transkripta kroz uzorke jednako je kritična jer izravno poboljšava integritet skupa podataka, što dovodi do poboljšane izvedbe modela i prirodnosti u TTS izlazima.
4. Uravnotežena zastupljenost: Skup podataka treba biti uravnotežen u smislu pokrivenosti sadržaja. To znači uključivanje reprezentativne distribucije fonetskih, sintaktičkih i semantičkih značajki tipičnih za ciljni jezik. Uravnotežena reprezentacija sprječava pristranosti prema specifičnim jezičnim značajkama i osigurava snažne mogućnosti generalizacije TTS modela.

Zaključno, idealan skup podataka za obuku TTS modela trebao bi biti velik, raznolik, dobro komentiran i uravnotežen u predstavljanju jezičnih značajki. Ove karakteristike zajednički pridonose poboljšanoj izvedbi modela, omogućujući stvaranje visokokvalitetnog sintetiziranog govora prirodnog zvuka u različitim jezicima i kontekstima.

2.6. Korištenje predtreniranog TTS modela

Unatoč početnim naporima da se obuči TTS model za hrvatski jezik, došlo je do značajnih izazova zbog kvalitete korištenih podataka, što je dovelo do neoptimalne izvedbe. Suprotno tome, obuka engleskih TTS modela pokazala se izvedivom, podržana raznovrsnim skupovima podataka visoke kvalitete i brojnim sofisticiranim modelima koji su prethodno obučeni te javno dostupni. Stoga je korištenje ovih predtreniranih engleskih TTS modela praktičan pristup koji osigurava učinkovita i visokokvalitetna TTS rješenja. Nažalost, trenutni nedostatak dovoljno kvalitetnih javno dostupnih podataka na hrvatskom jeziku otežava razvoj robusnog hrvatskog TTS modela u ovom trenutku. Budući napori trebali bi se usredotočiti na prikupljanje i dijeljenje visokokvalitetnih hrvatskih skupova podataka kako bi se riješio ovaj problem.

Predtrenirani model (eng. pre-trained model) je model strojnog učenja koji je već treniran na velikom skupu podataka i može se koristiti ili dodatno fino podešavati (eng. fine-tune) za određene zadatke. Obuka TTS modela od početka zahtijeva značajne računalne resurse i vrijeme. Korištenje predtreniranog modela omogućava zaobilaženje ovog zahtjevnog procesa. Često, javno dostupni modeli imaju izlaze visoke kvalitete jer je model već naučio nijanse prirodnog govora iz podataka visoke kvalitete. Također, predtrenirani modeli mogu se brzo integrirati u aplikacije, osiguravajući brže razvojne cikluse.

2.6.1. Glow-TTS

Za generiranje audio zapisa na engleskom jeziku korišten je „tts_models/en/ljspeech/glow-tts“ predtrenirani model koji je temeljen na Glow-TTS arhitekturi. Ovaj model treniran je na LJSpeech skupu podataka [29] koji sadrži tisuće audio zapisa jednog govornika koji čita različite tekstove u javnoj domeni te ga je moguće koristiti bez dodatnih treniranja. Upotreba uz pomoć Pythona je vrlo jednostavna. Potrebno je imati instaliranu TTS knjižnicu koju je moguće instalirati pomoću naredbe „`pip install tts`“. Pomoću skripte Kôd 2.2. generira se audio zapis koji se sastoji od dvije rečenice definirane unutar varijable *texts*.

```
import os
def generate_speech(text, index):
    command = f'tts --text "{text}" --model_name
"tts_models/en/ljspeech/glow-tts" --out_path
"./output_{index}.wav"'
    os.system(command)

texts = ["Hello, world!", "This is a test of the Glow-TTS
model."]
for idx, text in enumerate(texts):
    generate_speech(text, idx)
```

Kôd 2.2 Primjer korištenja TTS modela

Glow-TTS radi učinkovito, generirajući govor za kratke do srednje duge rečenice u sekundi, što ga čini prikladnim za interaktivne aplikacije. Često postiže ocjene preko 90% u procjenama prirodnosti i razumljivosti. Integracija u aplikacije je jednostavna, podržava popularne programske jezike i okvire, poboljšavajući korisničko iskustvo u implementaciji TTS-a. Glow-TTS se ističe svojom vrhunskom kvalitetom govora, učinkovitosti, prilagodljivosti i besprijekornoj integraciji, obećavajući kontinuirani napredak u tehnologiji TTS-a.

2.6.2. Your-TTS

Korištenjem your-tts modela moguće je dodatno personalizirati zvuk dodavanjem ulaznog uzorka koji omogućava kloniranje glasa (eng. voice cloning) čime se pospješuje sličnost ulaznog i izlaznog zvuka. Model „tts_models/multilingual/multi-dataset/your_tts“ temeljen je na naprednoj arhitekturi koja je prilagođena višejezičnim zadacima i trenirana na raznovrsnim skupovima podataka. Sličan je Glow-TTS modelu u smislu učinkovitosti i kvalitete generiranog govora, ali se ističe svojom sposobnošću da podrži više jezika i različite stilove govora.

Your-TTS koristi napredne tehnike obrade zvuka i strojnog učenja kako bi generirao govor koji je izuzetno sličan glasu uzorka koji se koristi kao ulaz. Funkcionalnost kloniranja glasa omogućava modelu da uči karakteristike specifičnog govornika iz kratkog audio uzorka, čime se postiže visok stupanj prilagodbe i autentičnosti. Ova mogućnost je posebno korisna u aplikacijama gdje je važno očuvati specifične karakteristike glasa, poput personaliziranih asistenata, audio knjiga i sličnih primjena.

Integracija your-tts modela u aplikaciju također je jednostavna, a njegova sposobnost da radi s više jezika i prilagođava se različitim govornicima čini ga izuzetno fleksibilnim alatom za raznovrsne primjene u području sintetiziranog govora.

3. Generiranje videozapisa

Sposobnost generiranja realističnih videozapisa pojedinaca koji govore iz audio zapisa i statičnih slika predstavlja značajan napredak u području multimedije i umjetne inteligencije. Ova tehnologija, koja nalazi primjenu u virtualnim asistentima, filmskim sinkronizacijama, personaliziranim avatarima itd., koristi sofisticirane modele za postizanje fotorealističnih i sinkroniziranih pokreta usana.

3.1. Povijesni pregled

Animiranje lica iz audio zapisa i statičnih slika ima dugu povijest, ukorijenjenu u ranoj animaciji i računalnoj grafici. U početku su animatori ručno sinkronizirali pokrete usana sa zvučnim zapisima, što je bio naporan proces prikazan klasičnom tehnikom rotoskopiranja [34], gdje su animatori pratili preko snimke slike u pokretu, kadar po kadar, kako bi proizveli realističnu radnju.

S pojavom računalne grafike 1970-ih i 1980-ih, istraživači su počeli istraživati automatizirane metode. Jedan od prvih značajnih napredaka bilo je korištenje animacije s ključnim kadrovima, gdje su definirane kritične točke kretanja, a srednji okviri generirani automatski. Ova metoda smanjila je fizički rad, ali je i dalje zahtijevala značajan doprinos animatora.

U 1990-ima uvedene su sofisticiranije tehnike pomoću softvera za 3D modeliranje i animaciju. Na primjer, Candide-3 je parametrizirana maska za lice koja se sastoji od 113 vrhova i 168 ploha, a omogućila je animaciju izraza lica u stvarnom vremenu manipuliranjem ograničenim brojem parametara [35]. Ovo je razdoblje također označilo početak korištenja tehnologije snimanja pokreta za snimanje pokreta lica i njihovu primjenu na 3D modele, čime je značajno poboljšan realizam.

Početkom 2000-ih uvedeno je strojno učenje u područje animacije lica. Istraživači su počeli koristiti skrivene Markovljeve modele (HMM, eng. Hidden Markov Model) i potporne vektorske strojeve (SVM, eng. Support Vector Machine) za modeliranje i predviđanje pokreta usana iz audio signala. Ovi su pristupi postavili temelje za naprednije metode temeljene na neuronskim mrežama koje danas dominiraju ovim područjem.

3.2. Moderne metode

3.2.1. LipGAN

Lip Generative Adversarial Network (LipGAN) dizajniran je za sintetiziranje pokreta usana u videozapisima. Koristi GAN-ove za kreiranje sinkroniziranih pokreta usana tako da uvjetuje proces generiranja na temelju ulaznog zvuka. Međutim, njegove mogućnosti su donekle ograničene. LipGAN radi tako da prvo izdvaja audio značajke iz ulaznog govora. Te se značajke unose u mrežu generatora koja predviđa odgovarajuće pokrete usana. Mreža diskriminatora procjenjuje autentičnost generiranih pokreta usana, osiguravajući njihovu usklađenost s pruženim zvukom. Ovaj suparnički proces obuke s vremenom poboljšava mogućnosti generatora.

LipGAN je vješt u stvaranju točne sinkronizacije usana, ali ima znatna ograničenja. Prvenstveno se fokusira na pokrete usana i ne uzima u obzir druge izraze lica ili pokrete glave, što rezultira manje prirodnim videozapisom. Ovo ga ograničenje čini manje prikladnim za aplikacije koje zahtijevaju punu dinamiku lica.

3.2.2. AD-NeRF

Audio-Driven Neural Radiance Fields (AD-NeRF) predstavlja vrhunski pristup sintetiziranju videozapisa glave koja govori korištenjem neuralnih polja zračenja. Ovaj model integrira audio značajke u 3D prikaz scene, proizvodeći vrlo realistične i detaljne izlaze. AD-NeRF kombinira neuronska polja radijacije s dinamikom pokretanom zvukom kako bi stvorio volumetrijski prikaz lica osobe. Model koristi ulazni zvuk za pokretanje pokreta i izraza 3D lica, stvarajući fotorealistične videozapise s izuzetnim detaljima u teksturi i pokretima.

AD-NeRF se ističe svojom sposobnošću generiranja iznimno realističnih videozapisa sa zamršenim detaljima. 3D pristup modela omogućuje prirodnije i izražajnije pokrete lica. Međutim, računalna složenost AD-NeRF-a znatno je veća, što dovodi do duljeg vremena obrade i veće potrošnje resursa.

3.2.3. MakeItTalk

MakeItTalk nudi drugačiji pristup fokusirajući se na brzu sintezu videozapisa glave koja govori. Dizajniran je tako da bude jednostavan za korištenje i učinkovit, pružajući rezultate

dobre kvalitete u znatno kratkom vremenu. MakeItTalk koristi duboku neuronsku mrežu koja kombinira zvučnu animaciju lica s tehnikama generiranja slike. Model obrađuje ulazni zvuk kako bi izdvojio fonetske značajke, a zatim ih usklađuje sa statičnom slikom osobe. Neuronska mreža je uvježbana da predvidi pokrete lica, uključujući sinkronizaciju usana, pokrete glave i izraze lica, koji odgovaraju danom zvuku.

MakeItTalk se ističe u pogledu vremenske učinkovitosti. Brzo generira realistične videozapise glave koja govori, što ga čini idealnim izborom za aplikacije koje zahtijevaju brzo vrijeme obrade. Model učinkovito integrira sinkronizaciju usana s pokretima glave i izrazima lica, što rezultira prirodnijim i impresivnijim videozapisom. Iako je kvaliteta nešto manje detaljna u usporedbi s modelima poput AD-NeRF-a, kompromis između brzine i vizualne vjernosti čini ga snažnim konkurentom za aplikacije u stvarnom vremenu.

3.2.4. Wav2Lip

Wav2Lip je napredni model dizajniran za rješavanje problema sinkronizacije usana. Wav2Lip koristi arhitekturu neuronske mreže koja se posebno fokusira na usklađivanje pokreta usana osobe u videu s određenim audio ulazom. Model je uvježban na velikim skupovima podataka koji sadrže različite govorne stilove i izraze, što mu omogućuje dobru generalizaciju na različite glasove i lica. Arhitektura uključuje mreže kodera i dekodera za učinkovitu obradu audio i video ulaza.

Wav2Lip postiže impresivne rezultate u pogledu točnosti sinkronizacije. Osigurava da pokreti usana točno odgovaraju zvuku, što je ključno za aplikacije koje zahtijevaju visoku realističnost. Međutim, poput LipGAN-a, prvenstveno se fokusira na sinkronizaciju usana i ne bavi se značajno pokretima glave ili izrazima lica, što može ograničiti prirodnost generiranih videozapisa.

3.3. Komparativna analiza metoda generiranja videa govornika

Model	Kvaliteta	Učinkovitost	Lakoća upotrebe
LipGAN	Fokusira se na sinkronizaciju usana; nedostaje razmatranje pokreta glave.	Zahtijeva značajne računalne resurse; ograničenog opsega.	Jednostavna integracija; prilagođen korisniku.

AD- NeRF	Proizvodi videozapise najviše kvalitete s finim detaljima i točnom sinkronizacijom usana; uključuje pokrete glavom.	Najmanje učinkovit; računalno intenzivan.	Složenija integracija zbog velikih zahtjeva za resursima.
MakeItTalk	Dobar omjer kvalitete i učinkovitosti; integrira pokrete usana i glave.	Vremenski najučinkovitiji; pogodan za aplikacije u stvarnom vremenu.	Lako integracija; prilagođen korisniku .
Wav2Lip	Točna sinkronizacija usana; ne bavi se značajno pokretima glave.	Učinkovit, ali nešto manje od MakeItTalk-a.	Lako integracija; pogodan za razne aplikacije .

Nakon evaluacije, MakeItTalk se pokazao kao optimalan izbor zbog svoje ravnoteže između vremenske učinkovitosti i kvalitete rezultata, što ga čini osobito pogodnim za aplikacije koje zahtijevaju brzo i realistično generiranje videa. U usporedbi, modeli poput Wav2Lip nude visoku preciznost u sinkronizaciji pokreta usana s audio zapisom, ali ograničavaju se na animaciju samo usana, bez kompletnog animiranja lica. S druge strane, AD-NeRF pruža vrhunsku kvalitetu i detaljnu animaciju cijelog lica, ali zahtijeva veće računalne resurse i nije uvijek praktičan za aplikacije u stvarnom vremenu ili s ograničenim resursima. Učinkovitost i jednostavnost MakeItTalk-a osiguravaju pouzdano rješenje za generiranje videa lica koje govori iz audio datoteka i statičnih slika.

3.4. MakeItTalk

MakeItTalk, inovativna AI arhitektura, revolucionira stvaranje sinkroniziranih animacija glave koja govori iz jedne slike lica i audio isječka. MakeItTalk pojednostavljuje proces stvaranja animacija glave koje govore. Potrebna je dobro osvijetljena slika lica te audio datoteka u WAV ili MP3 formatu za sinkronizaciju. Generiranje animacije može trajati od nekoliko sekundi do nekoliko minuta, ovisno o složenosti. Konačni rezultat je MP4 datoteka animacije glave koja govori.

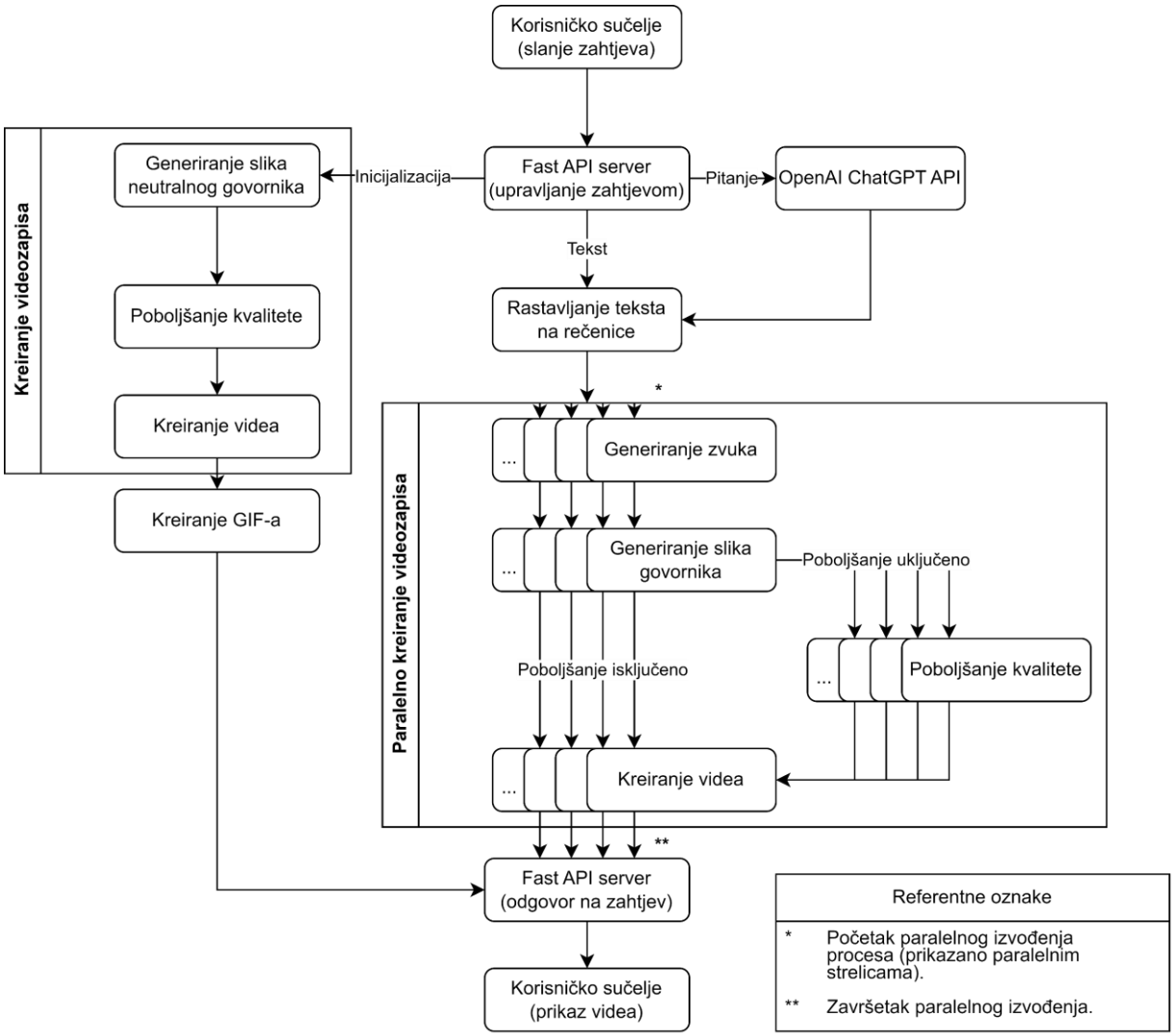
MakeItTalk koristi standardni 3D detektor orijentira lica za pretprocesiranje ulaza i izdvajanje orijentira lica. Iako bi osnovni model mogao animirati govor izravno s ovih orijentira, postizanje dinamike visoke vjernosti zahtijeva sofisticiraniji pristup. Predviđanje orijentira iz rastavljene reprezentacije govornog sadržaja i identiteta govornika značajno

poboljšava kvalitetu animacije. Kako bi to postigao, MakeItTalk koristi neuronsku mrežu za pretvorbu glasa kako bi odvojio govorni sadržaj od identiteta govornika. Rezultirajući sadržaj nevezan za govornika bilježi opće pokrete usana i obližnje izraze lica. U međuvremenu, dinamiku svjesnu govornika određuje određeni govornik, utječući na detalje kao što su oblik usana, pokreti očiju i pokreti glavom. Uvjetovan razdvojenim sadržajem i identitetom govornika, MakeItTalkov duboki model predviđa slijed orijentira sinkroniziranih sa zvukom, čineći osnovu za generiranje rasteriziranih slika. Za pretvaranje u slike, MakeItTalk koristi dva različita algoritma. Za nefotorealistične slike ili vektorske umjetnosti koristi Face Warp metodu koja se temelji na Delaunayevoj triangulaciji. Nasuprot tome, prirodna sinteza slike koristi mrežu prevođenja sa slike na sliku (Image2Image) nadahnutu pix2pixom. Ove tehnike osiguravaju da generirane slike vjerno predstavljaju animirane izraze lica. Kombinirajući ove sintetizirane slike s ulaznim zvukom, MakeItTalk proizvodi konačne, uvjerljive animacije glave koja govori.

4. Arhitektura sustava

Sustav za generiranje videozapisa sa zvukom temelji se na poslužiteljskoj arhitekturi koja je dizajnirana za učinkovito upravljanje različitim korisničkim zahtjevima. Ovaj poslužitelj, razvijen u Pythonu pomoću programskog okvira FastAPI [36], djeluje kao središnje čvorište za obradu zahtjeva te orkestraciju složenih zadataka poput generiranja videa i AI interakcije. FastAPI je moderan web-okvir visokih performansi za izradu aplikativnih programskih sučelja s Pythonom. Dizajniran je za brzi razvoj i uključuje značajke kao što je automatsko generiranje OpenAPI i JSON Schema dokumentacije. FastAPI iskorištava mogućnosti asinkronog programiranja za učinkovito rukovanje velikim brojem zahtjeva, što ga čini posebno pogodnim za aplikacije koje zahtijevaju visoku propusnost i nisku latenciju.

Poslužiteljska infrastruktura optimizirana je za generiranje videa visokih performansi, radi na hardveru opremljenim snažnim GPU-ovima. Ovi GPU-ovi su ključni za ubrzavanje intenzivnih izračuna uključenih u video renderiranje i poboljšanje slike. Kako bi se osigurao dosljedan i izoliran rad poslužitelja u različitim okruženjima, Apptainer [37] se koristi za kontejnerizaciju. Ovaj pristup ublažava ovisnosti i nedosljednosti konfiguracije koje bi inače mogle utjecati na performanse i pouzdanost.



Sl. 4.1 Dijagram toka zahtjeva

4.1. Inicijalizacija aplikacije

Srž ove funkcionalnosti leži u krajnjoj točki „/init“, koja upravlja učitavanjem slike i naknadnom obradom.

```
@app.post("/init")
async def init(file: UploadFile = File(...)):
...
    # Spremanje učitane slike na server
    image_path = os.path.join("video_generator/videos",
'input.jpg')
    with open(image_path, "wb") as buffer:
        buffer.write(await file.read())
...
```

```
    # Kreiranje neutralnog videa
    index, enhanced = 0, True
    shutil.copyfile("files/silence.wav",
f"video_generator/videos/output_{index}.wav")
    os.chdir("video_generator")
    os.system(f'python main_end2end.py --jpg input.jpg --
audio_file output_{index}.wav --index {index} --enhanced
{enhanced}')
    os.rename(f'output/output_{index}.mp4',
f'output/silence.mp4')
 ...
    # Generiranje poster.gif
    mp4_to_gif("output/silence.mp4", "output/poster.gif")

    return JSONResponse(content={"message": "Initialized"})
```

Kôd 4.1 Krajnja točka inicijalizacije aplikacije

Krajnja točka „/init“ prihvaća POST zahtjev sa slikovnom datotekom, koja se zatim obrađuje za stvaranje neutralnog videa i GIF-a. Kada se slika učita, sprema se u direktorij video_generator/videos kao input.jpg. Ova slika služi kao osnova za generiranje neutralnog videa i GIF-a, ali i za generiranje videa govornika u budućoj interakciji. Za stvaranje tihog videa, tiha audio datoteka (silence.wav) se kopira u direktorij video_generator/videos i preimenuje u output_0.wav. Ovaj tihi zvučni zapis uparen je s učitanom slikom kako bi se stvorio neutralni videozapis. Skripta zatim pokreće skriptu main_end2end.py s potrebnim parametrima za generiranje videa. Ovaj korak uključuje pretvaranje slike u video format uz njezinu sinkronizaciju s tihim audio zapisom. Izvršena naredba je Kôd 4.2.

```
python main_end2end.py --jpg input.jpg --audio_file
output_0.wav --index 0 --enhanced True
```

Kôd 4.2 Naredba za generiranje videozapisa iz slike

Nakon što se generirao videozapis, rezultirajuća video datoteka preimenuje se u silence.mp4 radi dosljednosti i lakog pristupa u budućim operacijama. Zatim se poziva funkcija mp4_to_gif za pretvaranje videa silence.mp4 u GIF datoteku pod nazivom poster.gif. Ovaj GIF se prikazuje kada osoba ne govori, povećavajući realistično interakcije korisnika i sustava. Krajnja točka vraća JSON odgovor koji pokazuje da je proces inicijalizacije završen, s porukom "Inicijalizirano", a aplikacija prelazi na sučelje za komunikaciju.

4.2. Interakcija s aplikacijom

Krajnja točka „/generate“ prihvaća POST zahtjev koji se sastoji od upita i razine željene kvalitete videozapisa. Ako je zahtjev upit, koristi se odgovor iz OpenAI ChatGPT API-ja kao ulaz za TTS model. Ako nije upit, koristi se korisnikov unos kao ulaz za TTS model. Nakon zaprimanja tekstnog ulaza, sustav koristi GlowTTS, točnije „tts_models/en/ljspeech/glow-tts“, model kako bi tekst pretvorio u živopisni govor. Ovaj korak ključan je jer stvara zvučnu podlogu za videozapis, osiguravajući prirodno i koherentno predstavljanje glasa. Dobiveni audio zapis i predefinirana slika lica koriste se kao ulaz za generiranje videozapisa. Za generiranje videozapisa koristi se MakeItTalk.

```
def make_video(index, answer, enhanced=False):
    ...
    os.system(f'tts --text "{answer}" --model_name
"tts_models/en/ljspeech/glow-tts" --out_path
"./video_generator/videos/output_{index}.wav"')
    # copy silence to video_generator/videos
    shutil.copyfile("files/silence.wav",
f"video_generator/videos/silence.wav")
    # Load the sounds
    sound1 =
AudioSegment.from_file(f"./video_generator/videos/output_{ind
ex}.wav")
    sound2 =
AudioSegment.from_file("./video_generator/videos/silence.wav"
)
...
combined_sound.export(f"./video_generator/videos/output_{inde
x}.wav", format="wav")
    os.chdir("video_generator")
    os.system(f'python main_end2end.py --jpg input.jpg --
audio_file output_{index}.wav --index {index} --enhanced
{enhanced}')
    with contextlib.suppress(FileNotFoundError):

os.remove(f'/videos/input_pred_fls_output_{index}_audio_embed
.mp4')
        os.remove(f'out_{index}.mp4')
```

Kôd 4.3 Generiranje videozapisa govornika

Kako bi se povećala učinkovitost i smanjilo vrijeme obrade poslužitelj dijeli tekst u pojedinačne rečenice, dopuštajući da se svaka rečenica samostalno obradi i prikaže. Ova strategija poboljšava granularnost zadatka i olakšava precizniju sinkronizaciju audio i video zapisa. Nadalje, paralelne tehnike obrade koriste se za ubrzavanje video generiranja. Paralelnom distribucijom zadataka generiranja slika videozapisa, poslužitelj može rukovati s nekoliko okvira istovremeno. Ova paralelizacija značajno smanjuje vrijeme potrebno za generiranje kompletnog videa, osiguravajući brzu isporuku visokokvalitetnog sadržaja korisnicima.

```
batch_size = 2  # Number of processes to run in parallel
def start_processes(answers, batch_size):
    processes = []
    for index, answer in enumerate(answers):
        process = multiprocessing.Process(target=make_video,
args=(index, answer, tts_text.video_quality == 'high'))
        processes.append(process)

    for i in range(0, len(processes), batch_size):
        batch = processes[i:i + batch_size]
        for process in batch:
            process.start()
        for process in batch:
            process.join()
```

Kôd 4.4 Paralelno pokretanje procesa za stvaranje videozapisa

```
batch_size, total_frames = 8, len(fls)
for start in range(0, total_frames, batch_size):
    end = min(start + batch_size, total_frames)
    batch = fls[start:end]
    with concurrent.futures.ThreadPoolExecutor() as executor:
        futures = [executor.submit(enhance_frame, i, frame,
enhanced) for i, frame in enumerate(batch, start=start)]

    # Wait for all futures in the current batch to complete
    concurrent.futures.wait(futures)
```

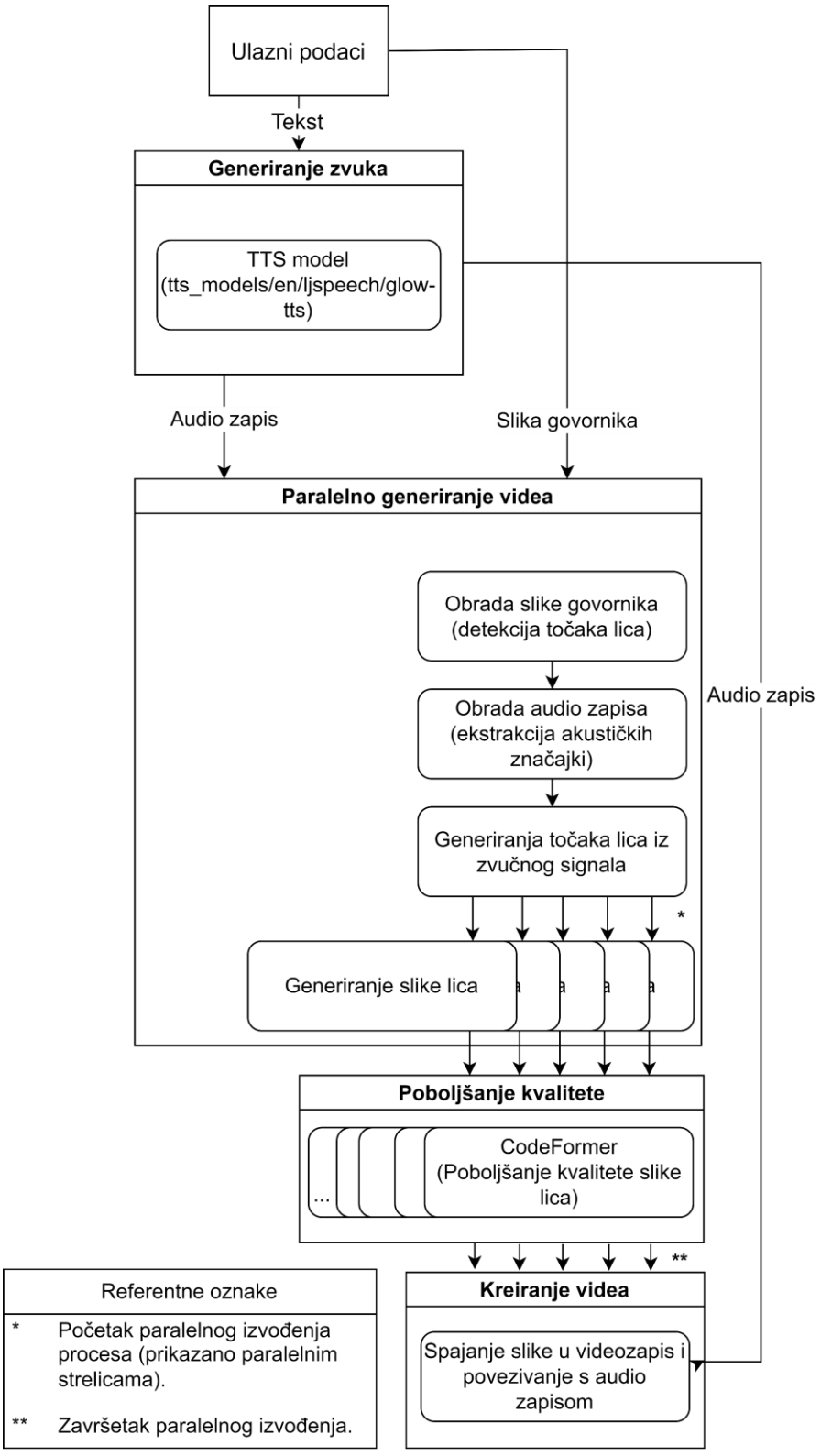
Kôd 4.5 Paralelno procesiranje slika koristeći ThreadPoolExecutor

Određivanje kvalitete kritični je aspekt sustava. Ovisno o korisničkim specifikacijama, videozapisi se mogu generirati u visokoj ili niskoj kvaliteti. Visokokvalitetna video

produkcija uključuje dodatni sloj obrade, gdje poslužitelj primjenjuje napredne tehnike poboljšanja slike koristeći CodeFormer model [38]. CodeFormer je robustan projekt restauracije lica koji koristi Codebook Lookup Transformer za poboljšanje kvalitete slika lica. Dizajniran je za različite primjene, uključujući poboljšanje cijele slike i videa, bojanje lica i crtanje. Na ovaj način, svaka slika iz generiranog videa prolazi kroz precizno poboljšanje koristeći CodeFormer, čime se osigurava da je konačni rezultat jasan, koherentan i vizualno upečatljiv.

```
if enhanced:
  os.system(f'python
../video_enhancement/inference_codeformer.py -w 0.7 --
input_path {frame_path} --output_path images_{index}')
```

Kôd 4.6 Poboljšanje kvalitete slike



Sl. 4.2 Detaljniji prikaz procesa kreiranja videozapisa

Ovaj proces rezultira stvaranjem videozapisa govornika koji simulira prirodne pokrete usana i ekspresije lica. Korištenjem naprednih tehnika obrade slika i dubokog učenja, postiže se visoka razina detalja i fluidnosti animacije. Generirani videozapisi mogu na autentičan način

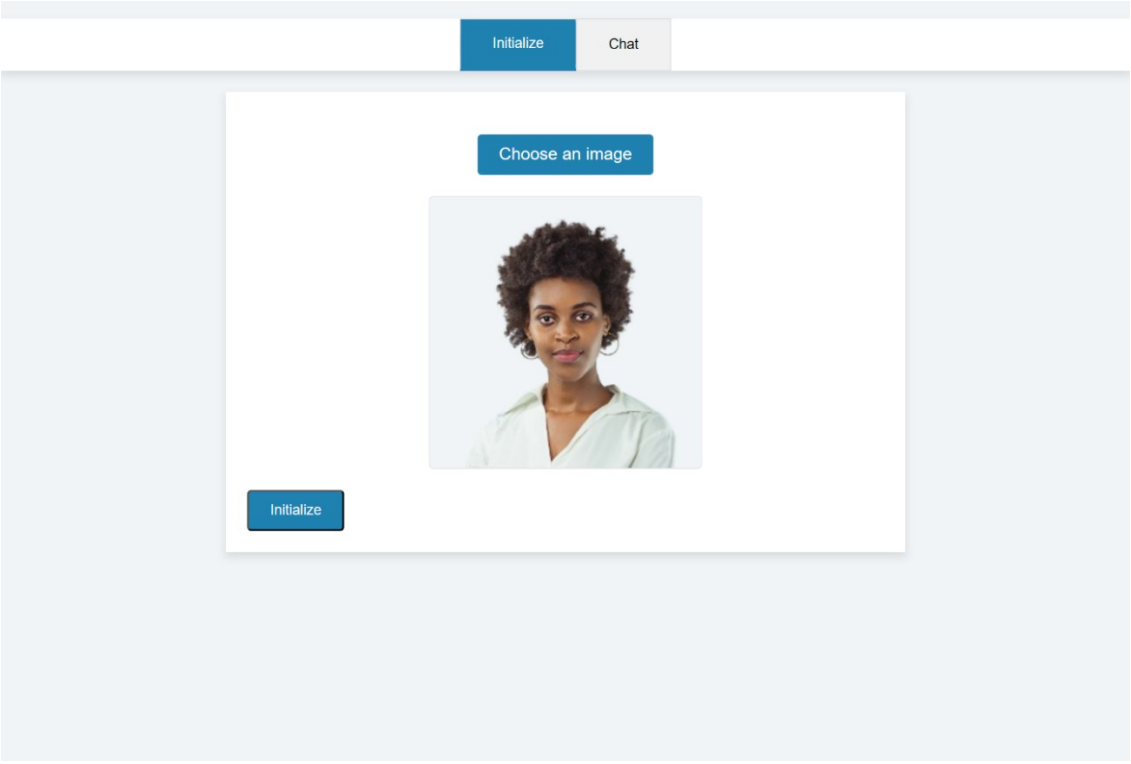
reproducirati izraze lica i intonacije glasa, što ih čini korisnima u aplikacijama kao što su digitalni asistenti, edukacijski sadržaji i stvaranje personaliziranih video poruka.

5. Korištenje aplikacije

5.1. Inicijalizacija aplikacije

Korisničko sučelje aplikacije korisnicima omogućuje proces inicijalizacije aplikacije koji se sastoji od učitavanja slike i pokretanje procesa za generiranje neutralnog videa i GIF-a. Ova je funkcionalnost sastavni dio stvaranja vizualno privlačnog i jedinstvenog korisničkog iskustva u sučelju za chat aplikacije.

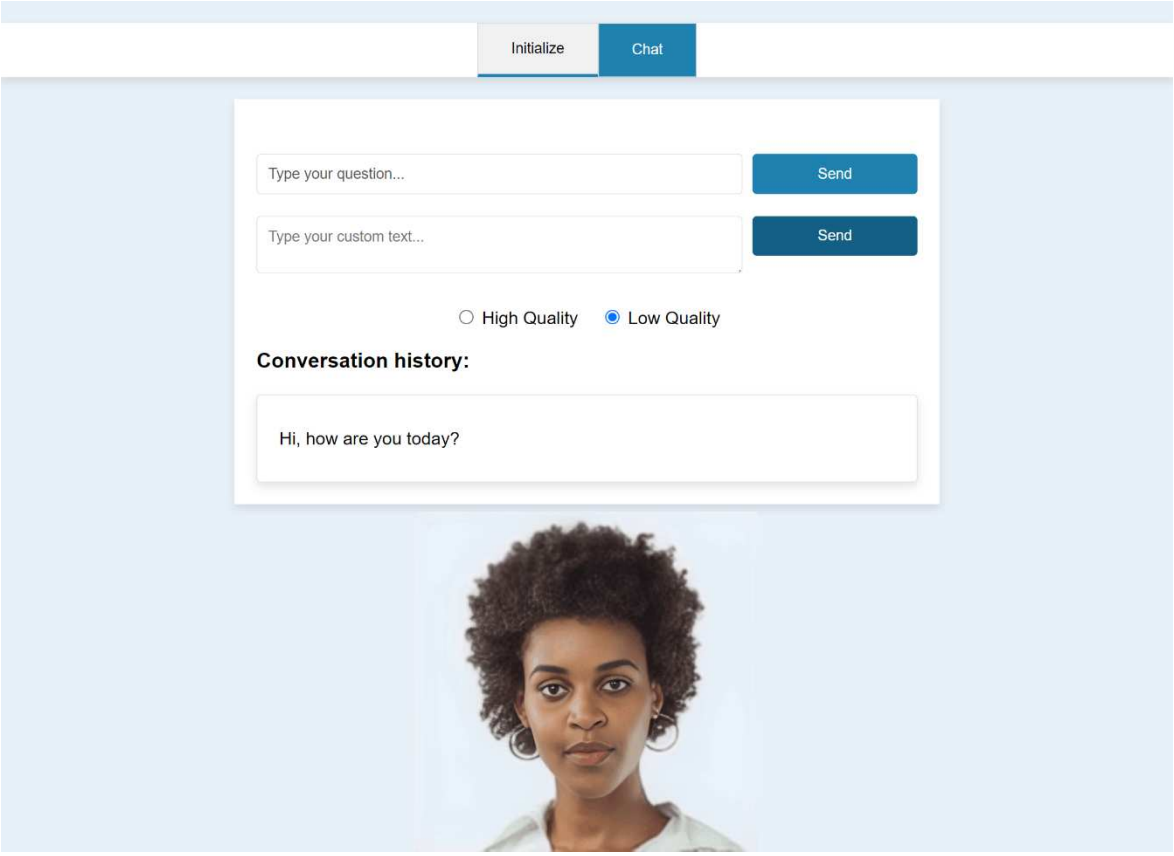
Prilikom posjete web stranici klikom na gumb „Choose an image“ učitava se slika osobe u formatu 256x256 piksela. Nakon što se slika učita i klikne gumb „Initialize“, sučelje šalje POST zahtjev pozadinskoj krajnjoj točki „/init“. Aplikacija sadrži krajnju točku dizajniranu za rukovanje učitavanjem slike i pretvaranjem slike u neutralni video govornika te GIF koji se generira iz navedenog videa. Ova je funkcionalnost ključna za stvaranje vizualno privlačnog i personaliziranog korisničkog iskustva u sučelju za chat aplikacije. Prikazuje se ikona učitavanja koja označava da je obrada u tijeku. Kada je proces dovršen, aplikacija prelazi na sučelje za chat gdje se koriste generirani neutralni video i GIF, poboljšavajući iskustvo razgovora.



Sl. 5.1 Snimka zaslona aplikacije – proces inicijalizacije

5.2. Interakcija s aplikacijom

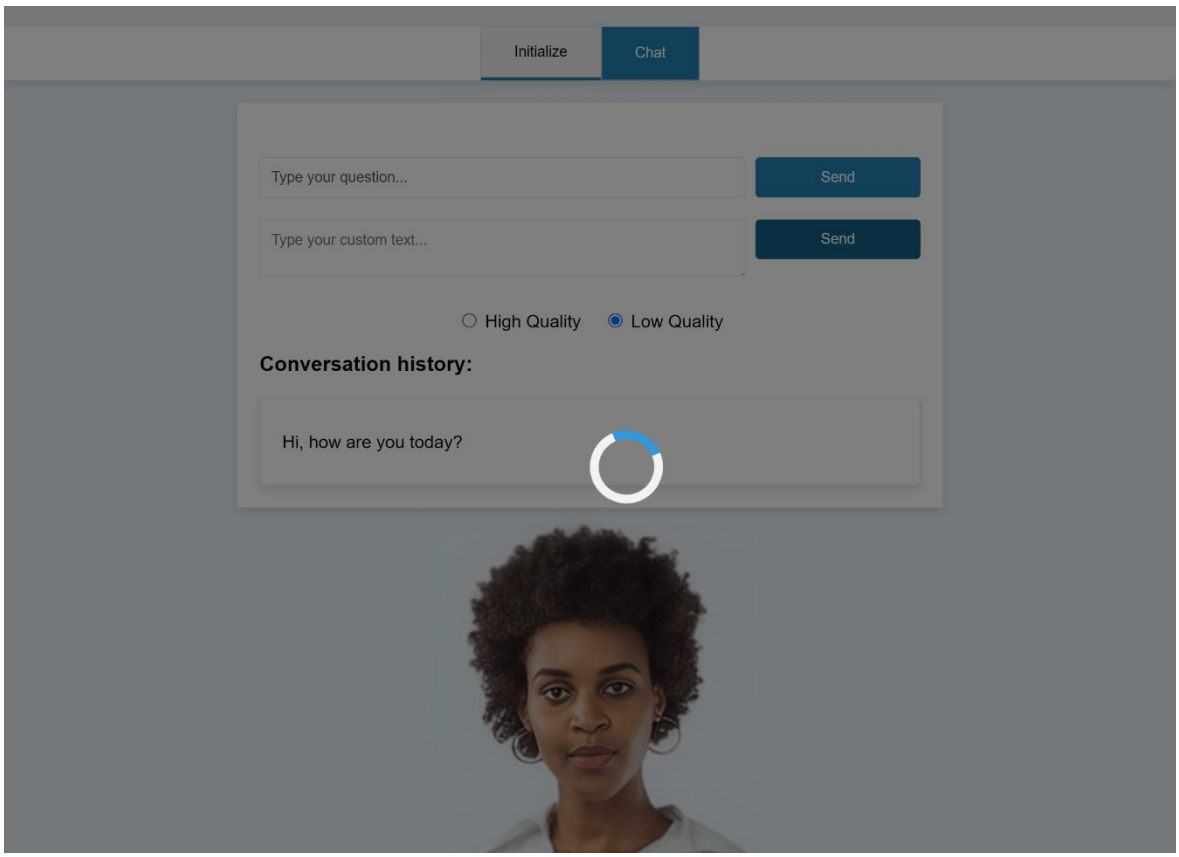
Korisničko sučelje omogućuje korisnicima slanje teksta ili pitanja. Prilikom posjeta web-stranici, korisnici mogu unijeti tekst u jedno od dostupnih polja za unos teksta. Nakon toga imaju opciju odabrati žele li generirati videozapis visoke ili niske kvalitete (izbor će utjecati na vrijeme potrebno za generiranje videa). Nakon toga potrebno je kliknuti na „Send“ gumb pokraj ispunjenog polja. Ova akcija radi slanje POST zahtjeva na krajnju točku „/generate“. Ako je zahtjev upit, koristi se odgovor iz OpenAI ChatGPT API-ja kao ulaz za TTS model. Ako nije upit, koristi se korisnikov unos kao ulaz za TTS model.



Sl. 5.2 Snimka zaslona aplikacije – sučelje za komunikaciju

Nakon zaprimljenog zahtjeva, aplikacija prikazuje ikonu učitavanja koja označava da je obrada u tijeku. Po završetku obrade, generirani videozapis se prikazuje, a u dijelu

"Conversation history" pojavljuje se poruka. Klikom na tu poruku prikazuje se potrebno vrijeme za izvršenje obrade.



Sl. 4.3 Snimka zaslona aplikacije – generiranje videa

6. Performanse sustava

Procjena performansi ključna je za procjenu učinkovitosti sustava. Ovo poglavlje istražuje mogućnosti i ograničenja računalne konfiguracije opremljene AMD EPYC 7763 procesorima i NVIDIA A100 grafičkim karticama, fokusirajući se na njihove uloge i doprinose ukupnoj performansi sustava pri različitim radnim opterećenjima.

Sustav koristi napredne hardverske komponente:

- AMD EPYC 7763: Sa 64 jezgre i 128 niti, optimiziran za visoku računalnu propusnost i energetsku učinkovitost.
- NVIDIA A100 40GB GPU: Iskorištava Ampere arhitekturu s 6,912 CUDA jezgri i 432 Tensor jezgri, dizajniran za paralelno procesiranje i kompleksne matematičke operacije.

Ključni mjerni pokazatelji performansi uključuju:

- Vrijeme obrade: Mjereno u sekundama za generiranje video sadržaja pri različitim veličinama grupa i razinama kvalitete.
- Iskorištenje memorije: Korištenje memorije u scenarijima visokog opterećenja za identifikaciju potencijalnih uskih grla.
- Razine kvalitete: Procjena utjecaja primjene naprednih tehnika poput CodeFormer-a na poboljšanje kvalitete videozapisa.

Testni scenariji:

- Varijabilnost veličine grupe: Usporedba performansi između veličina grupe 4, 8, 16 i 32 radi razumijevanja skalabilnosti i alokacije resursa.
- Usporedba CPU-a i GPU-a: Uvođenje testova isključivo na CPU-u kako bi se istaknula važnost GPU-a u ubrzanju vremena obrade i upravljanju zadacima s visokim zahtjevima za memoriju.

6.1. Rezultati testiranja

Veličina grupe obrade slika, označava broj slika koje se istovremeno obrađuju tijekom generiranja i poboljšanja videozapisa. Veća veličina grupe može povećati efikasnost obrade, ali zahtijeva više resursa. Uređaj prikazuje na kojem je uređaju obavljeno testiranje, a može

biti kombinacija GPU + CPU (centralna procesorska jedinica) ili samo na CPU. GPU je često puno brži u obradi paralelnih zadataka, kao što je generiranje i poboljšanje slika. Razina kvalitete pokazuje na kojoj razini kvalitete su slike generirane. To može uključivati postavke niska ili visoka kvaliteta, što utječe na detalje i jasnoću generiranih slika, ali također može utjecati i na vrijeme obrade. Vrijeme obrade u sekundama pokazuje koliko je vremena bilo potrebno za obradu zadane veličine grupe slika na određenom uređaju i razini kvalitete. Testiranje je provedeno kako bi se dokazala potreba za GPU-om, uspoređujući performanse generiranja videozapisa na različitim uređajima, tako da se može vidjeti koliko GPU doprinosi bržoj obradi u odnosu na CPU.

- Ulazni tekst: „Hi, how are you today?“

Tablica 6.1 Rezultati testiranja performansi s jednom rečenicom

Veličina grupe obrade slika	Uređaj	Razina kvalitete	Vrijeme obrade (sekunde)
32	CPU + GPU	Niska	48
32	CPU + GPU	Visoka	*Out of memory*
32	CPU	Niska	83
16	CPU + GPU	Niska	48
16	CPU + GPU	Visoka	589
16	CPU	Niska	75
8	CPU + GPU	Niska	48
8	CPU + GPU	Visoka	627
8	CPU	Niska	65
4	CPU + GPU	Niska	48
4	CPU + GPU	Visoka	766
4	CPU	Niska	67
2	CPU + GPU	Niska	52
2	CPU + GPU	Visoka	1.140
2	CPU	Niska	79

Pri evaluaciji performansi sustava za generiranje videa korištenjem različitih veličina grupa (eng. batch sizes) i postavki kvalitete, ističe se nekoliko ključnih opažanja. Pri veličini grupe od 32, konfiguracije s CPU-om + GPU-om učinkovito obrađuju niskokvalitetnu obradu u samo 48 sekundi dok pokušaj visokokvalitetnog izlaza rezultira greškama nedostatka memorije, ističući ograničenja GPU memorije pri intenzivnim zadacima. U usporedbi s time, obrada samo s CPU-om pri istoj veličini grupe traje 83 sekunde za niskokvalitetne rezultate, što pokazuje značajnu brzinsku prednost ubrzanog računanja pomoću GPU-a.

Kako se veličine grupa smanjuju na 16 i 8, postavke s CPU-om + GPU-om zadržavaju brze obrade od 48 sekundi za zadatke niskokvalitetne obrade, ali nailaze na primjetan porast na 589 i 627 sekundi, redom, za visokokvalitetne izlaze, što pokazuje računalni teret

poboljšanja kvalitete videa. S druge strane, oslanjanje samo na CPU produžuje vrijeme obrade na 75 sekundi pri veličini grupe 16, te na 65 sekundi pri veličini grupe 8, što potvrđuje ključnu ulogu GPU-a u ubrzanju multimedijalnih zadataka.

Pri manjim veličinama grupa od 4 i 2, konfiguracije s CPU-om + GPU-om i dalje učinkovito obavljaju obradu niskokvalitetnih rezultata, isporučujući ih u 48 i 52 sekunde, ali se suočavaju s izazovima pri visokokvalitetnim izlazima, s trajanjem od 766 i 1.140 sekundi, što ukazuje na značajne računalne zahtjeve. Istovremeno, operacije samo s CPU-om u sličnim scenarijima kreću se od 67 do 79 sekundi za zadatke niskokvalitetne obrade, naglašavajući dobitak u učinkovitosti prilikom korištenja GPU resursa.

- Ulazni tekst: „The serene lake reflected the colors of the sunset like a painter's masterpiece. In the bustling city, amidst the honking cars and hurried pedestrians, she found a moment of quiet in a quaint cafe.“

Tablica 6.2 Rezultati testiranja performansi s dvije rečenice

Veličina grupe obrade slika	Uređaj	Razina kvalitete	Vrijeme obrade (sekunde)
16	CPU + GPU	Niska	48
16	CPU + GPU	Visoka	*Out of memoray*
16	CPU	Niska	140
8	CPU + GPU	Niska	51
8	CPU + GPU	Visoka	1.961
8	CPU	Niska	103
4	CPU + GPU	Niska	53
4	CPU + GPU	Visoka	1.075
4	CPU	Niska	123

U analizi rezultata performansi za generiranje videa iz ulaznog teksta može se izvući nekoliko ključnih uvida u vezi učinkovitosti sustava i utjecaja hardverskih konfiguracija. Za veličinu grupe obrade od 16 slika, konfiguracija CPU + GPU učinkovito obrađuje izlaze niske kvalitete za 48 sekundi, ali nailazi na grešku zbog nedostatka memorije pri obradi visoke kvalitete, što ukazuje na ograničenja GPU memorije. Nasuprot tome, obrada samo na CPU-u za izlaze niske kvalitete traje znatno dulje, 140 sekundi, što naglašava važnost GPU akceleracije. Kod veličine grupe od 8, konfiguracija CPU + GPU održava učinkovitu obradu niske kvalitete za 51 sekundu, dok vrijeme obrade visoke kvalitete raste na 1.961 sekundu zbog intenziteta računanja potrebnog za poboljšanje videa. Konfiguracija samo na CPU-u pokazuje bržu obradu nego kod veličine serije od 16, ali i dalje zaostaje za kombiniranom

konfiguracijom CPU + GPU, s 103 sekunde za izlaze niske kvalitete. Za veličinu serije od 4, sustav CPU + GPU nastavlja učinkovito obrađivati izlaze niske kvalitete s minimalnim povećanjem vremena, ali obrada visoke kvalitete ostaje vremenski zahtjevna s 1.075 sekundi. Obrada samo na CPU-u za ovu veličinu serije traje 123 sekunde, što dodatno naglašava prednosti performansi korištenja GPU-a.

Sveukupno, ovi rezultati pokazuju da konfiguracije CPU + GPU značajno poboljšavaju učinkovitost obrade za izlaze niske kvalitete za sve veličine serija, dok obrada visoke kvalitete postavlja značajne računalne izazove, posebno u smislu kapaciteta GPU memorije i vremena obrade. Oštar kontrast u vremenima obrade između konfiguracija samo na CPU-u i CPU + GPU za izlaze niske kvalitete naglašava ključnu ulogu GPU-a u ubrzavanju zadataka generiranja videa. Ova saznanja naglašavaju važnost optimizacije upravljanja memorijom i računalnim resursima kako bi se učinkovito balansirala brzina obrade i kvaliteta izlaza, zadovoljavajući zahtjeve složenih multimedijskih aplikacija.

Zaključak

Proces stvaranja audio zapisa i videozapisa iz teksta i slika uključuje nekoliko naprednih tehnologija. MakeItTalk, istaknuti alat u ovoj domeni, generira realistične videozapise govornih glava iz jedne slike. Ovaj proces uključuje modele dubokog učenja koji animiraju izraze lica i sinkroniziraju pokrete usana s govorom. Za generiranje zvuka koriste se TTS modeli poput GlowTTS-a koji pretvaraju pisani tekst u prirodno zvučeći govor. Ovi modeli koriste arhitekture neuronskih mreža kako bi proizveli visokokvalitetni zvuk koji odgovara intonaciji i ritmu ljudskog govora. Postoje brojne alternative kako za MakeItTalk tako i za GlowTTS, što odražava dinamično i brzo napredujuće područje.

Analiza performansi generiranja audio zapisa i videozapisa iz tekstualnih i slikovnih unosa naglašava značajne prednosti korištenja naprednih hardverskih konfiguracija. Sustavi koji kombiniraju CPU i GPU resurse dosljedno nadmašuju konfiguracije koje koriste samo CPU, posebno za računalno intenzivne zadatke. Ovo pokazuje važnost GPU akceleracije u optimizaciji učinkovitosti obrade i kvalitete izlaza. Testovi performansi su pokazali da su konfiguracije CPU + GPU vrlo učinkovite za obradu videa niske kvalitete, dok izlazi visoke kvalitete predstavljaju veće izazove zbog povećanih zahtjeva za memorijom i računalnim resursima. Povijesno gledano, evolucija hardverske tehnologije potaknula je značajna poboljšanja u računalnim sposobnostima. Višejezgreni CPU-ovi i napredni GPU-ovi revolucionirali su paralelnu obradu i upravljanje velikim količinama podataka, što je ključno za moderne zadatke.

Prednosti ovakvih naprednih sustava su brojne. Brže vrijeme obrade, kvalitetniji izlazi i poboljšana učinkovitost u stvaranju multimedijalnog sadržaja imaju praktične primjene u zabavi, obrazovanju, marketingu i virtualnoj stvarnosti. Sposobnost generiranja videa iz teksta i slika otvara nove mogućnosti za kreativno izražavanje i automatizaciju, povećavajući produktivnost i inovativnost.

Zaključno, analiza performansi naglašava kritičnu važnost GPU akceleracije i učinkovitog upravljanja resursima u sustavima za generiranje videa. Trenutna tehnologija omogućuje izgradnju visokoučinkovitih sustava koji brzo i učinkovito isporučuju kvalitetne rezultate. Kako se hardver nastavlja razvijati, mogu se očekivati daljnja poboljšanja u računalnim sposobnostima, omogućujući još sofisticiranije stvaranje multimedijalnog sadržaja.

Literatura

[1] M. Brand, "Voice puppetry," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999, pp. 21–28.

[2] T. Shiratori, A. Nakazawa, and K. Ikeuchi, "Dancing-to-music character animation," in *Computer Graphics Forum*, 2006, pp. 449–458.

[3] C. Gan, D. Huang, P. Chen, J. B. Tenenbaum, and A. Torralba, "Foley music: Learning to generate music from videos," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, 2020, pp. 758–775.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[5] Yang Zhou, Xintong Han, Eli Shechtman, Jose Echevarria, Evangelos Kalogerakis, and Dingzeyu Li, "MakeItTalk - SA'20." Accessed: Feb. 01, 2024. [Online]. Available: https://people.umass.edu/~yangzhou/MakeItTalk/

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[7] I. Goodfellow *et al.*, "Generative adversarial nets," *Adv Neural Inf Process Syst*, vol. 27, 2014.

[8] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes." 2022.

[9] A. Caillon and P. Esling, "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis." 2021.

[10] "Tacotron 2 | PyTorch." Accessed: Jun. 13, 2024. [Online]. Available: https://pytorch.org/hub/nvidia_deeplearningexamples_tacotron2/

[11] W. Ping *et al.*, "Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning." 2018.

[12] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Neural Speech Synthesis with Transformer Network." 2019.

[13] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-tts: A generative flow for text-to-speech via monotonic alignment search," *Adv Neural Inf Process Syst*, vol. 33, pp. 8067–8077, 2020.

[14] S. E. Eskimez, R. K. Maddox, C. Xu, and Z. Duan, "Noise-resilient training method for face landmark generation from speech," *IEEE/ACM Trans Audio Speech Lang Process*, vol. 28, pp. 27–38, 2019.

[15] S. E. Eskimez, R. K. Maddox, C. Xu, and Z. Duan, "Generating talking face landmarks from speech," in *Latent Variable Analysis and Signal Separation: 14th International Conference, LVA/ICA 2018, Guildford, UK, July 2–5, 2018, Proceedings 14*, 2018, pp. 372–381.

[16] L. Chen, R. K. Maddox, Z. Duan, and C. Xu, "Hierarchical cross-modal talking face generation with dynamic pixel-wise loss," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7832–7841.

[17] D. Greenwood, I. Matthews, and S. Laycock, “Joint learning of facial expression and head pose from speech,” 2018.

[18] P. Pataranutaporn, J. Leong, V. Danry, A. P. Lawson, P. Maes, and M. Sra, “AI-Generated Virtual Instructors Based on Liked or Admired People Can Improve Motivation and Foster Positive Emotions for Learning,” in *2022 IEEE Frontiers in Education Conference (FIE)*, 2022, pp. 1–9. doi: 10.1109/FIE56618.2022.9962478.

[19] “Soul Machines Gallery | AI Assistants.” Accessed: Jun. 13, 2024. [Online]. Available: https://www.soulmachines.com/

[20] “Replika.” Accessed: Jun. 13, 2024. [Online]. Available: https://replika.com/

[21] “Pattern playback.” Accessed: Jun. 13, 2024. [Online]. Available: https://infosekolah.net/en/Pattern_playback

[22] G. E. Henter, S. King, T. Merritt, and G. Degottex, “Analysing Shortcomings of Statistical Parametric Speech Synthesis,” Jul. 2018, Accessed: Jun. 13, 2024. [Online]. Available: https://arxiv.org/abs/1807.10941v1

[23] Y. Wang *et al.*, “Tacotron: Towards end-To-end speech synthesis,” *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2017-August, pp. 4006–4010, 2017, doi: 10.21437/Interspeech.2017-1452.

[24] J. Shen, R. Pang, R. Weiss, M. S.-… on acoustics, undefined speech, and undefined 2018, “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions,” *ieeexplore.ieee.orgJ Shen, R Pang, RJ Weiss, M Schuster, N Jaitly, Z Yang, Z Chen, Y Zhang, Y Wang2018 IEEE international conference on acoustics, speech and signal …, 2018•ieeexplore.ieee.org*, Accessed: Jun. 13, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8461368/?casa_token=HgTASVZpgiQAAAAA:jSmO3RL-B3djn1560IGrub1imhjQTrdRyhwZMicQyRdmkfU0qvE3l8b0pY_OnEBPiyYoVhO8ZdAVUw

[25] Y. Ren *et al.*, “Fastspeech: Fast, robust and controllable text to speech,” *proceedings.neurips.ccY Ren, Y Ruan, X Tan, T Qin, S Zhao, Z Zhao, TY LiuAdvances in neural information processing systems, 2019•proceedings.neurips.cc*, Accessed: Jun. 13, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/hash/f63f65b503e22cb970527f23c9ad7db1-Abstract.html

[26] “Hugging Face – The AI community building the future.” Accessed: Feb. 01, 2024. [Online]. Available: https://huggingface.co/

[27] N. Ljubešić, D. Koržinek, P. Rupnik, and I.-P. Jazbec, “ParlaSpeech-HR - a Freely Available ASR Dataset for Croatian Bootstrapped from the ParlaMint Corpus,” in *Proceedings of the Workshop ParlaCLARIN III within the 13th Language Resources and Evaluation Conference*, D. Fišer, M. Eskevich, J. Lenardič, and F. de Jong, Eds., Marseille, France: European Language Resources Association, Jun. 2022, pp. 111–116. [Online]. Available: https://aclanthology.org/2022.parlaclarin-1.16

[28] N. Ljubešić, D. Koržinek, and P. Rupnik, “Parliamentary spoken corpus of Croatian ParlaSpeech-HR 2.0.” 2024. [Online]. Available: http://hdl.handle.net/11356/1914

[29] K. Ito and L. Johnson, “The LJ Speech Dataset.” 2017.

[30] Coqui GmbH, “Glow TTS - TTS 0.22.0 documentation.” Accessed: Feb. 01, 2024. [Online]. Available: https://docs.coqui.ai/en/dev/models/glow_tts.html

[31] “Training a Model - TTS 0.22.0 documentation.” Accessed: Feb. 01, 2024. [Online]. Available: https://docs.coqui.ai/en/dev/training_a_model.html

[32] “Apptainer - Portable, Reproducible Containers.” Accessed: Feb. 01, 2024. [Online]. Available: https://apptainer.org/

[33] “neongeckocom/tts-vits-cv-hr · Hugging Face.” Accessed: Jun. 17, 2024. [Online]. Available: https://huggingface.co/neongeckocom/tts-vits-cv-hr

[34] “Rotoscoping - Wikipedia.” Accessed: Jun. 17, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Rotoscoping

[35] J. Ahlberg, “CANDIDE-3—An updated parameterized face,” Feb. 2001.

[36] “FastAPI.” Accessed: Mar. 01, 2024. [Online]. Available: https://fastapi.tiangolo.com/

[37] “Apptainer - Portable, Reproducible Containers.” Accessed: Jan. 03, 2024. [Online]. Available: https://apptainer.org/

[38] S. Zhou, K. C. K. Chan, C. Li, and C. C. Loy, “Towards Robust Blind Face Restoration with Codebook Lookup TransFormer,” in *NeurIPS*, 2022.

Sažetak

Generiranje audio zapisa i videozapisa: korištenje GPU ubrzanja i naprednih alata

Performanse sustava za generiranje videa koristeći napredne hardverske konfiguracije, naglašavaju važnost GPU akceleracije. Testovi s različitim veličinama grupa i postavkama kvalitete otkrivaju izazove i učinkovitost u obradi. Studija prikazuje alate poput MakeItTalk-a za kreiranje videa i GlowTTS-a za sintezu zvuka, ističući njihovu ulogu u proizvodnji visokokvalitetnog multimedijalnog sadržaja. Korištenjem moćnih CPU-ova i GPU-ova, pokazano je kako trenutna tehnologija omogućuje učinkovito i sofisticirano generiranje videa i zvuka. Rezultati naglašavaju potrebu za optimiziranim upravljanjem resursima kako bi se postigla uravnotežena brzina obrade i kvaliteta izlaza.

Ključne riječi: generiranje zvuka, generiranje videa, napredne hardverske konfiguracije

Summary

Audio and video generation: using GPU acceleration and advanced tools

The performance of a video generation system using advanced hardware configurations highlights the importance of GPU acceleration. Tests with varying batch sizes and quality settings reveal the challenges and efficiencies in processing. Tools like MakeItTalk for video creation and GlowTTS for audio synthesis are showcased, emphasizing their roles in producing high-quality multimedia content. By leveraging powerful CPUs and GPUs, the study demonstrates how current technology enables efficient and sophisticated video and audio generation. The findings underscore the need for optimized resource management to achieve balanced processing speed and output quality.

Keywords: sound generation, video generation, advanced hardware configurations

Skraćenice

AI	Artificial Intelligence	umjetna inteligencija
ML	Machine Learning	strojno učenja
DL	Deep Learning	duboko učenje
RNN	Recurrent Neural Networks	rekurentne neuronske mreže
LSTM	Long Short-Term Memory	mreže dugog kratkoročnog pamćenja
CNN	Convolutional Neural Networks	konvolucijske neuronske mreže
GAN	Generative Adversarial Networks	generativne suparničke mreže
VAE	Variational Autoencoders	varijacijski autokoderi
TTS	Text-to-Speech	tekst u govor
SPSS	Statistical Parametric Speech Synthesis	statistička parametarska metoda sinteze
HMM	Hidden Markov Model	skriven Markovljev model
SVM	Support Vector Machine	potporni vektorski stoj