

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

RUNAS SINTĒZE LATVIEŠU VALODĀ

BAKALaura DARBS

Autors: **Krišs Saulītis**

Studentu apliecības Nr.: ks18108

Darba vadītājs: Phd. Comp. Sc. Ēvalds Urtāns

RĪGA, 2024

ANOTĀCIJA

Pētījuma mērķis bija izstrādāt augstākas kvalitātes un precizitātes latviešu valodas runas sintēzes modeli, izpētot datu kopas priekšapstrādes un apmācības metodes. Lai to sasniegtu tika veikta eksperimentālā salīdzināšana un izstrādāts jauns runas sintēzes modelis latviešu valodā, izmantojot inovatīvas datu priekšapstrādes un apmācības metodes. Pārbaudot esošās runas sintēzes datu kopas un to priekšapstrādes procedūras, tika sagatavotas un apmācītas datu kopas, kā arī veikts salīdzinājums ar tirgū pieejamiem runas sintēzes rīkiem. Jaunizstrādātais modelis sasniedza augstākas kvalitātes rādītāju nekā citi runas sintēzes rīki - NISQA 5.02 un CER 0.17%. Pētījums norāda, ka kvalitatīvai runas sintēzei ir būtiski veikt rūpīgu datu kopu priekšapstrādi, izmantojot ASR modeļus, uzlabojot balss ierakstu kvalitāti un runātāju balss pārneši.

Darba kopējais apjoms ir 42 lappuses.

Atslēgvārdi: runas sintēze, latviešu valoda, dziļā mašīnmācīšanās, balss pārveidošana, runas uzlabošana

ABSTRACT

The goal of this study was to develop speech synthesis model for the Latvian language that achieves a higher quality and precision. To do this, the study conducted an experimental comparison and the development of a new speech synthesis model for the Latvian language, employing innovative data preprocessing and training methods. Existing speech synthesis datasets and their preprocessing procedures were examined, leading to the preparation and training of a dataset, which was then compared with commercially available speech synthesis tools. The newly developed model achieved higher quality metrics, than other speech synthesis tools, recording a NISQA score of 5.02 and a CER of 0.17%. The research indicates that for high-quality speech synthesis, meticulous preprocessing of datasets is essential, utilizing ASR models to enhance the quality of voice recordings and the transfer of speaker voices.

The total amount of this work is 42 pages.

Keywords: speech synthesis, latvian language, deep-learning, voice conversion, speech enhancement

Saturs

Ievads	6
1. Saistītie pētījumi	7
1.1. Digitālā signālu apstrāde	7
1.2. Savienošanas metode	8
1.3. Mašīnmācīšanās metode	9
1.4. Rādītāji	16
1.5. VITS	17
2. Sistemātiskā literatūras analīze	21
2.1. Runas sintēze angļu valodā	21
2.2. Runas sintēze latviešu valodā	21
2.3. Audio attrokšņošana un uzlabošana	23
2.4. Runas stila pārnese	24
3. Metodoloģija	26
3.1. Datu kopas un to priekšapstrāde	26
3.2. Rādītāji	28
3.2.1. WER un CER	28
3.2.2. NISQA	28
3.2.3. Kosinusa attālums	29
3.3. Apmācības protokols	30
4. Rezultāti	32
5. Secinājumi	35
6. Tālākie pētījumi	37
Bibliogrāfija	38

Apzīmējumu saraksts

AI (Artificial Intelligence) - Mākslīgais intelekts

CE (Cross-Entropy) - Krustentropija

CER (Character Error Rate) - Simbolu kļūdas biežums

MAE (Mean Absolute Error) - Vidējā absalūtā kļūda

ML (Machine Learning) - Mašīnmācīšanās

MSE (Mean Squared Error) - Vidējā kvadrātiskā kļūda

NISQA (Non-intrusive Objective Speech Quality Assessment) - Neinvazīvs objektīvs runas kvalitātes novērtējums

STD (Standard Deviation) - Standartnovirze

TTS (Text to Speech) - No teksta uz runu

WER (Word Error Rate) - Vārdu kļūdas biežums

Ievads

Runāšana ir viena no svarīgākajām cilvēku valodas prasmēm. Datoriem šo prasmi nodrošina teksta runas sintēze, kas veic teksta pārveidošanu dabiskā cilvēka balsī. Šī ir būtiska tehnoloģija plašam lietojumu lokam – sākot ar ekrānu lasītājiem un audio grāmatām, līdz pat virtuālo asistentu izstrādei. Šī tehnoloģija ir īpaši nozīmīga latviešu valodā, kurā ir mazāk pieejamu resursu un pētījumu kā lielajās pasaules valodās. Augstas kvalitātes runas sintēze latviešu valodā var veicināt tehnoloģiju pieejamību plašākam sabiedrības lokam un uzlabot valodas tehnoloģiju attīstību [35].

Darba mērķis ir izstrādāt augstākas kvalitātes un precizitātes latviešu valodas runas sintēzes modeli, izpētot datu kopas priekšapstrādes un apmācības metodes. Lai sasniegtu šo mērķi, tika izvirzīti seši **darba uzdevumi**:

1. Izpētīt esošo zinātnisko literatūru par angļu valodas runas sintēzes metodēm.
2. Izpētīt esošo zinātnisko literatūru par latviešu valodas runas sintēzes metodēm.
3. Izpētīt runas sintēzes datu kopu priekšapstrādes metodes.
4. Veikt datu kopas atlasīšanu un sagatavošanu latviešu valodas runas sintēzes modeļa izveidei.
5. Veikt latviešu valodas runas sintēzes modeļu apmācību un labākā modeļa atlasīšanu un salīdzināšanu ar esošajiem runas sintēzes rīkiem.
6. Izdarīt secinājumus un izvirzīt priekšlikumus un rekomendācijas, pamatojoties un iegūtajiem rezultātiem.

Tiek izvirzītas arī sekojošās **hipotēzes**:

1. Sintezētās balss kvalitātes un precizitātes noteikšanai apmācības laikā nepietiek ar testa kļūdas skalāro vērtību, lai noteiktu runas sintēzes modeļa veiktspēju.
2. Runas uzlabošanas modeļu izmantojums datu priekšapstrādē uzlabo kvalitātes un precizitātes rādītājus.
3. Balss toņa pārveide uz viena runātāja balss toni uzlabo kvalitātes un precizitātes rādītājus.

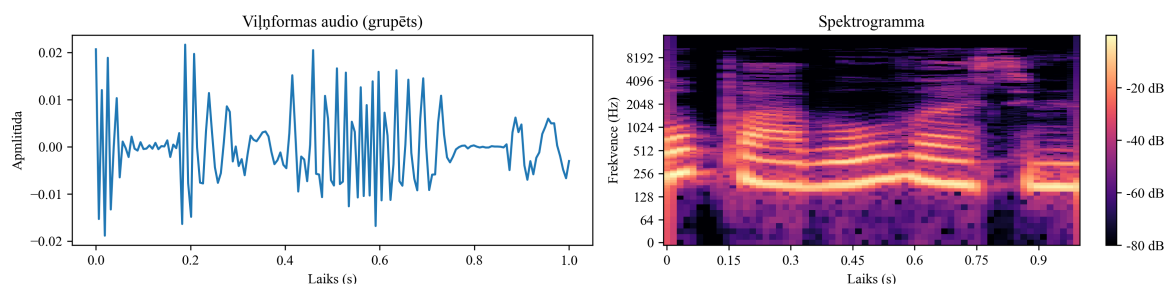
Runas sintēze latviešu valodā ne tikai paplašina tehnoloģiju pielietojuma iespējas, bet arī veicina latviešu valodas saglabāšanu un attīstību digitālajā vidē. Šī darba rezultāti var kalpot par pamatu turpmākiem pētījumiem un uzlabojumiem šajā jomā, piedāvājot inovatīvus risinājumus un pieejas runas tehnoloģiju izstrādē.

1 Saistītie pētījumi

Šajā nodaļā tiks apskatīti saistītie pētījumi, kas veido teorētisko un praktisko pamatu runas sintēzes tehnoloģiju izpētei un attīstībai. Runas sintēze ir starpdisciplināra zinātnes nozare, kas ietver akustikas, lingvistikas, digitālās signālu apstrādes, mašīnmācīšanās un neironu tīklu metodoloģiju apvienojumu. Tā kā runas sintēze ietver sarežģītu algoritmu izstrādi un datu apstrādes metožu pielietošanu, ir būtiski izprast galvenos jēdzienus un pieejas, kas veicina šo tehnoloģiju attīstību. Tiks aplūkoti arī dažādi runas sintēzes aspekti un metodes. Sākumā tiks izskatīta digitālā signālu apstrāde, kas veido pamatu runas datu iegūšanai un apstrādei. Pēc tam tiks analizētas dažādas runas sintēzes pieejas, sākot no tradicionālajām savienošanas metodēm līdz mūsdienu mašīnmācīšanās metodēm. Īpaša uzmanība tiks pievērsta dziļās mašīnmācīšanās pieejām un ģeneratīvajiem modeļiem, kas ir būtiski, lai uzlabotu runas sintēzes kvalitāti un dabiskumu.

1.1 Digitālā signālu apstrāde

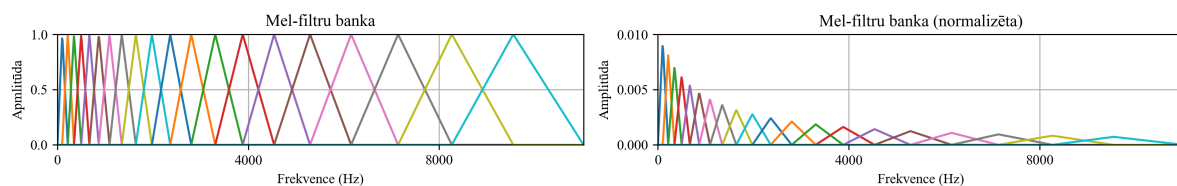
Digitālā signālu apstrāde, īpaši runas signālu apstrāde, ietver akustisko signālu pārveidi digitālos datus. Sākotnēji skaņa tiek uztverta kā analogs signāls, kas tiek pārvērsts digitālā formātā, mērot analogo signālu amplitūdu regulāros intervālos, lai iegūtu diskretu signāla reprezentāciju. Paraugu ņemšanas ātrumam jābūt vismaz divreiz lielākam par augstāko frekvenci signālā, kā to nosaka Naikvista-Šenona paraugu ņemšanas teorēma [33]. Piemēram, cilvēka balss frekvenču diapazons ir no aptuveni 80 Hz līdz 8 kHz, bet cilvēka auss uztver frekvences no 20 Hz līdz 20 kHz. Tādējādi, ja ir jāieraksta tikai balss, bieži tiek izmantotas paraugu ņemšanas frekvences 16 kHz vai 22,05 kHz, bet, lai aptvertu pilnu cilvēka dzirdes diapazonu, tiek izmantotas 44,1 kHz vai 48 kHz frekvences [36].



1. att. Runas audio fragmena viļņformas signāls un tā transformācija spektrogrammā

Runas sintēzes sistēmās liela nozīme ir arī Furjē transformācijas (FT) metodei un tās variācijai - īstermiņa Furjē transformācijai (STFT). Šīs metodes palīdz analizēt runas frekvenču saturu īsos laika posmos, nodrošinot laika-frekvences reprezentāciju, ko sauc par spektrogrammu (skatīt 1. attēlu). Pēc tam spektrogrammām veic pēcapstrādes soļus, piemēram, lineārās skalas spektrogrammu pārvēršanu mel-skalā, kas pielāgo tās cilvēka

dzirdes uztverei, izceļot frekvences, kuras cilvēka auss uztver jutīgāk [36]. Papildus iespējams veikt mel-skalas normalizāciju, kas izlīdzina spektrālās enerģijas sadalījumu starp dažādiem mel-filtriem, nodrošinot, ka katra filtra laukums zem liknes ir vienāds. Šāda normalizācija palīdz uzlabot filtru reakciju vienmērīgumu, kas ir īpaši svarīgi modeļa apmācībā. Normalizācijas izmantošana padara modeļa apmācību stabilāku un veicina labāku runas sintēzes kvalitāti. Abu mel-filtru banku salīdzinājums redzams 2. attēlā.

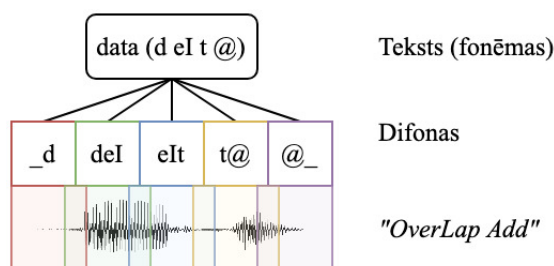


2. att. Vienkāršas un normalizētas mel-filtru bankas salīdzinājums

1.2 Savienošanas metode

Savienošanas metode ir viena no pirmajām runas sintēzes metodēm, kurā kā viena no galvenajām tiek izmantota tieši difonos bāzēta balss sintēze. Šīs metodes pamatā ir difonu savienošana - runas segmenti glabājas datu bāzē, un inferences laikā TTS sistēma meklē runas segmentus, balstoties uz ievades tekstu. Runas segmenti tiek savienoti, lai izveidotu viļņformas runas audio. Difons ir runas segments, kur katrs segments sastāv no diviem fonētiskajiem elementiem. Tas sākas fonēmas stabilajā vidusdaļā un beidzas nākamās fonēmas stabilajā vidusdaļā. Izmantojot difonus kā pamatelementus, sintēzē apvienošanas punkti tiek novietoti fonēmu stabilās daļās, kas atvieglo izlīdzināšanas operāciju veikšanu sintēzes laikā un samazina iespējamās nepārtrauktības apvienošanas punktos [31, 36].

Viena no plašāk izmantotajām sistēmām ir MBROLA (*Multi-Band Resynthesis OverLap Add*), jeb daudzjoslu atkārtotā sintēze, izmantojot pārklāšanās apvienojumu. Šī sistēma izmanto noteiktas valodas viena runātāja difonu ierakstu datubāzi, kas pirms tam tiek sagatavota un normalizēta. Runas sintēze notiek, atlasot sagatavotos difonu ierakstus un veicot to apvienošanu, izmantojot pārklāšanās apvienošanas algoritmu (*OverLap Add*), kā parādīts 3. attēlā. Savienošanas metodes, tai skaitā MBROLA, sintezētās runas kvalitāte bieži tiek vērtēta kā augsti saprotama, bet diezgan datorizēta, jo difonu datu bāze nesastāv no visiem cilvēka runas kombināciju variantiem un ģenerātajai balsij nav iespējams kontrolēt emocionalitāti [31, 4].



3. att. Vienkāršots difonu apvienošanas modeļa process [30]

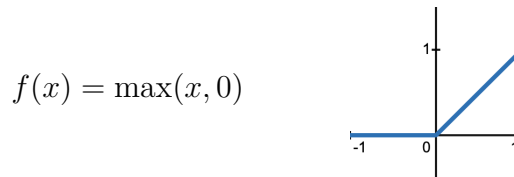
1.3 Mašīnmācīšanās metode

Mašīnmācīšanās ir mākslīgā intelekta metode, kurā tiek veidoti modeļi, kas tiek apmācīti, izmantojot jebkāda veida datus. Mūsdienās biežāk tiek izmantota tieši dziļā mašīnmācīšanās, kas ir mašīnmācīšanās paveids kurā modeļa izmērs un parametru skaits tiek palielināts, lai tiktu iegūts labāks tā darbības rezultāts. Ar šīs metodes palīdzību ir strauji attīstījušās dažādas jomas, tai skaitā - runas sintēze.

Mašīnmācīšanos var iedalīt vairākās mācīšanās paradigmās. **Pārraudzītās mācīšanās** (Supervised Learning) mērķis ir atrast optimālākos modeļa parametrus, kas pārvērs ievadi izejā, pamatojoties uz apmācības datu kopu jeb korpusu, kur treniņu piemērs ir ievades un izvades datu pāris. Pārraudzītās mācīšanās algoritms analizē un nosaka pamatā esošos paraugus un attiecības starp treniņu piemēru ievadi un izeju, un spēj ģenerēt izeju neredzētiem piemēriem. Uzraudzītā mācīšanās ir viena no pamatmācīšanās paradigmām, un to plaši izmanto regresijai un klasifikācijai [36]. **Nepārraudzītā mācīšanās** (Unsupervised learning) atšķirībā pārraudzītās atšķiras ar to, ka tā analizē un kategorizē nemarķētas datu kopas pēc paraugu kopējām iezīmēm. Visbiežāk šādu pieeju izmanto datu kategorizēšanai, anomāliju atklāšanai, latentu mainīgo tipa iekodēšanai un cita veida modeļiem [36]. **Stimulētā mācīšanās** balstās uz vides atgriezenisko saiti jeb mācīšanās signālu. Pēc lēmumu pieņemšanas modelim tiek dota atbilde jeb atlīdzības vērtība par to, cik pareizs vai nepareizs bija šis pieņemtais lēmums. Modelis mācās, cenšoties maksimizēt iegūto atlīdzības vērtību. Stimulētā mācīšanās tiek plaši izmantota spēlēs, pašbraucošajās automašīnās, kā arī audio apstrādē [36].

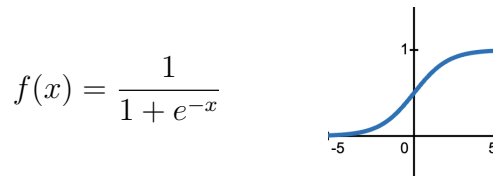
Mašīnmācīšanās pamatā ir neironu tīkli, kas imitē cilvēku smadzeņu neironus. Neironu tīkli sastāv no savstarpēji savienotām neironu mezglu grupām, kas ir sakārtotas slāņos: ieejas slānis, vairāki slēptie slāņi un izvades slānis. Katrs neirons saņem ieejas signālu no iepriekšējiem saistītajiem slāņiem, apstrādā tos, izmantojot **aktivizācijas funkciju**, un nodod rezultātu tālāk nākamajam slānim. Tas tiek darīts, lai tīkls nebūtu lineārs un tiktu ieviesta papildus sarežģītība. Dažas no izplatītākajām aktivizācijas funkcijām ir [11]:

- **ReLU** - saglabā pozitīvās vērtības un negatīvās vērtības pārvērš par 0, kā redzams 4. attēlā.



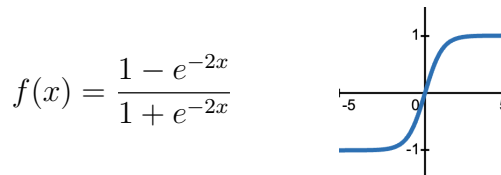
4. att. ReLU funkcijas formula un grafiks

- **Sigmoid** - visas vērtības saspiež jeb pārveido, lai tās būtu robežās no 0 līdz 1, kā redzams 5. attēlā.



5. att. Sigmoid funkcijas formula un grafiks

- **Tanh** - līdzīgi kā Sigmoid funkcijā, padotās vērtības tiek saspiešanas robežās no -1 līdz 1, kā redzams 6. attēlā.



6. att. Tanh funkcijas formula un grafiks

Lai veiktu neironu apmācību un to svaru uzlabošanu, nepieciešams modeļa **klūdas funkcijas** aprēķins, kas nosaka modeļa izvades un reālās vērtības atšķirību. Atkarībā no mācīšanās metodes, modeļa tipa un datu kopas, tiek atlasīta vispiemērotākā klūdas funkcija. Divas no galvenajām un plašāk izmantotajām klūdas funkcijām ir [11]:

- **MSE** - vidējā kvadrātiskā kļūda jeb dispersija. Izmanto regresijas uzdevumos.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

- **CE** - krustentropija. Izmanto klasifikācijas uzdevumos, kur modeļa mērķis ir klasificēt ieejas datus kategorijās. Var izmantot gan binārajā, gan vairāku kategoriju klasifikācijā.

$$CE = - \sum_x p(x) \cdot \log q(x)$$

Kā nākamais solis neirona tīkla apmācībā ir **atpakaļizplatīšanās algoritms**, kas aprēķina neironu tīkla parametru krituma vērtību un veic atpakaļizplatīšanos, izmantojot tīklu pretējā virzienā - no izvades slāņa līdz ieejas slānim. Šī algoritma pamatā ir atvasinājumu ķēdes likums, kas aprēķina tīkla neironu parciālos atvasinājumus. Ķēdes likums ļauj aprēķināt atvasinājumu viena mainīgā attiecībā pret otru, izmantojot funkciju kompozīciju, tādējādi vienkāršojot sarežģītus atvasinājumu aprēķinus. Piemēram, ja tiek dota funkcija $Y = f(X)$ un $Z = g(Y)$, tad, izmantojot ķēdes noteikumu, var aprēķināt Z atvasinājumu attiecībā pret X - $\frac{\partial Z}{\partial X} = \frac{\partial Z}{\partial Y} \cdot \frac{\partial Y}{\partial X}$

Tālāk tiek izmantoti **optimizācijas algoritmi** jeb optimizatori un iegūtie atvasinājuma funkcijas gradienti no kļūdas vērtības, lai atjaunotu katra mezgla svarus. Optimizatori virza modeli tā, lai kļūdas funkcija samazinātos, bet precizitāte pieaugtu. Darbs pie optimizatoriem turpinās, un tiek izstrādātas dažādas metodes, kas ņem vērā vairākus aprēķinātos gradientus. Divi no visplašāk izmantotajiem optimizatoriem ir SGD (*Stochastic Gradient Descent*) un Adam algoritmi.

SGD jeb stohastiskā gradienta nolaišanās, ir balstīta uz nejauši izvēlētu apakškopu no treniņu datiem, lai aprēķinātu svaru atjauninājumus, tādējādi ievērojami samazinot aprēķinu sarežģītību un laiku salīdzinājumā ar pilna gradienta metodi. SGD nodrošina ātrāku konvergenci lieliem datu kopumiem un ļauj efektīvāk trenēt dziļus neironu tīklus. $\theta_{t+1} = \theta_t - \eta \nabla_{\theta} \mathcal{L}(\theta_t; x_i, y_i)$, kur θ_t apzīmē modeļa parametrus iterācijā t , η ir mācīšanās ātrums, $\nabla_{\theta} \mathcal{L}(\theta_t; x_i, y_i)$ ir zuduma funkcijas \mathcal{L} gradients attiecībā pret parametriem θ , kas tiek aprēķināts, izmantojot viena, nejauši atlasīta, treniņa piemēru (x_i, y_i) [11].

Adam (Adaptive Moment Estimation) [16] apvieno divas labi zināmas metodes: AdaGrad [9] un RMSProp [37], lai nodrošinātu adaptīvu mācīšanās ātrumu katram parametram. Šī optimizētāja priekšrocība ir tā spēja automātiski pielāgot apmācības ātrumu, izmantojot pirmā un otrā momenta novērtējumus, kas palīdz uzturēt stabilu un ātru konvergenci. To apraksta ar formulu:

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

kur m_t un v_t attiecīgi ir pirmā un otrā momenta novērtējums (eksponenciāli mainīgais vidējais gradienta kvadrāts), \hat{m}_t un \hat{v}_t attiecīgi ir nobīdīts pirmā un otrā momenta novērtējums, θ_t ir modeļa parametri iterācijā t , η ir mācīšanās ātrums, β_1 un β_2 ir eksponenciālā samazinājuma ātruma koeficienti pirmā un otrā momentam, g_t ir gradients attiecībā pret parametriem θ iterācijā t , ϵ ir neliels skaitlis, lai novērstu dalīšanu ar nulli [11].

Balstoties uz šīm pamata konstrukcijām, tiek veidoti dažādi mašīnmācīšanās modeļi ar atšķirīgām struktūrām un pielietojumiem. Optimizatori, piemēram, SGD un Adam,

kopā ar atpakaļizplatīšanās algoritmu, kalpo par šo modeļu pamatu. Šajā kontekstā, mašīnmācīšanās arhitektūru dažādība ir plaša, un tās pielieto dažādās nozarēs un uzdevumos. Vispārīgi var izdalīt četras galvenās arhitektūras struktūru grupas - DNN, CNN, RNN un Transformeri, kur katrai no tām ir savas specifiskās īpašības un pielietojuma jomas.

DNN (Dense Neural Network) jeb blīvais neironu tīkls, saukts arī par FCN (Fully Connected Network), jeb pilnībā savienotu tīklu ir jau iepriekš aprakstītais neironu tīkls, kur katrs mezgls i slānī $l + 1$ savienojas ar mezglu j slānī l , balstoties uz svara w_{ij} . Pēc tam seko kāda no iepriekš aprakstītajām aktivizācijas funkcijām [36].

CNN (Convolutional Neural Network) jeb konvolūciju neironu tīkli sastāv no konvolūciju kodoliem vai filtriem, un izmanto slīdošā loga principu, kur vieni un tie paši parametri tiek pielietoti dažādām attēla vietām. Šī tīkla galvenie slāņi ir:

- Konvolūcijas slānis - tas veic galveno aprēķinu daļu, izmantojot kodolu vai filtru, kas pārvietojas pāri attēlam, lai izveidotu aktivizācijas karti. Šis slānis ļauj CNN uztvert un analizēt kādas lokālās iezīmes, piemēram, līnijas un leņķus pirmajā slānī un jau lielākas detaļas kā kādas formas nākamajos slāņos. Šis slānis ir apmācāms.
- Apvienošanas (*pooling*) slānis - samazina attēla izmēru, saglabājot būtiskāko informāciju, bet netiek apmācīts. Šis slānis palīdz padarīt modeli noturīgu pret nelielām attēla izmaiņām. Populārākie apvienošanas slāņu tipi ir vidējās apvienošanas slānis (Average Pooling), kas aprēķina vidējo kodola vērtību, maksimālais apvienošanas slānis (Max Pooling), kas aprēķina maksimālo kodola vērtību.
- Pilnībā savienots (*fully connected*) slānis - kalpo kā neironu tīkla “domāšanas” daļa, kas analizē iepriekšējos slāņos iegūto informāciju un veic gala klasifikāciju vai citu uzdevumu, izmantojot datu saspiešanas metodi lineārā slānī.

Šāda veida arhitektūra ļauj modeļiem tikt galā ar dažādiem attēlu apstrādes uzdevumiem, piemēram, objektu atpazīšanu, semantisko segmentāciju, attēlu aprakstu un ne tikai, jo attēls var būt arī, piemēram, audio spektrogramma, tādējādi padarot šāda veida arhitektūru piemērotu arī audio apstrādei [11].

RNN (Recurrent Neural Network), jeb rekurentais neironu tīkls ir izstrādāts datu secību apstrādei, balstoties uz “atmiņas” konceptu. Atšķirībā no dziļajiem neironu tīkliem (DNN), RNN struktūrā signāla izvade un ievade no iepriekšējiem soļiem tiek nodota nākamajiem soļiem, tādējādi iepriekšējā informācija un tās rezultāti ietekmē pašreizējās informācijas un soļu apstrādes rezultātus. Rekurentos neironu tīklus izmanto laikā mainīgu datu apstrādei, kur dati vienā laika solī ir atkarīgi no iepriekšējo laika soļu datu punktiem. Šo arhitektūru var pielietot finanšu datu prognozēšanā, mašīntulkošanā, dabiskās valodas apstrādē, kā arī runas sintēzē, kur rezultāts ir audio viļņformas. Viena no RNN būtiskākajām priekšrocībām ir spēja apstrādāt mainīga garuma secības, saglabājot

informāciju par iepriekšējām ievadēm. Tas ir īpaši svarīgi uzdevumos, kuros ir nozīmīga konteksta uztvere. Tomēr RNN struktūrai ir arī problēmas, piemēram, pazūdošie un eksplodējošie gradienti, kas apgrūtina tīkla efektīvu apmācību un mācīšanos no datiem ar ilgtermiņa atkarībām. Lai risinātu šīs problēmas, tika izstrādātas RNN variācijas, piemēram, ilgtermiņa atmiņas (LSTM) tīkls un vārtu rekurento vienību (GRU) tīkls, kas ļauj RNN efektīvāk apstrādāt ilgtermiņa atkarības un uzlabo spēju mācīties no sarežģītākām datu kopām [36], [11].

Transformeri jeb pašuzmanības (Self-Attention) neironu tīkls savienojumus organizē ar pašuzmanības mehānismu, balstoties uz līdzības mērījumiem. Šis mehānisms aprēķina uzmanības koeficientu kopumu, kas nosaka, cik lielu uzmanību katram vārdam jāpievērš, lai iegūtu noteikta vārda reprezentāciju. Transformeru arhitektūra [38] izmanto šo pašuzmanības mehānismu kopā ar tiešās padeves (Feed Forward) tīklu, lai efektīvi uztvertu sarežģītas atkarības un kontekstuālo informāciju. Atšķirībā no RNN, transformeri var apstrādāt veselus secinājumus paralēli, ievērojami uzlabojot skaitļošanas efektivitāti un mērogojamību. Šis paralēlisms kopā ar spēju modelēt ilgtermiņa atkarības, izmantojot pašuzmanību, ir padarījis transformerus par standartu dažādu uzdevumu veikšanai - dabiskās valodas apstrādei, runai un datorredzei [36].

Runas sintēzē ir ļoti tipisks laika rindu apstrādes uzdevums. Šāda tipa uzdevumus iedala trīs ietvaru grupās.

Iekodētājs (Encoder) pārveido ievades talonus, kas var būt, piemēram, teksta reprezentācija, par slēpto reprezentāciju secību. Iesākumā iekodētāju pamatā tika izmantotas RNN vai CNN struktūras, bet pēdējā laikā īpaši efektīvi un populāri ir kļuvuši Transformeru iekodētāji, jo tie var paralēli apstrādāt visus ievaddatus un uztvert saistības starp tiem. Šādā veidā iekodētājs pārveido, piemēram, tekstu par kompleksu slēpto reprezentācijas vektoru, ko tālāk var izmantot klasifikācijas vai regresijas uzdevumiem, vai, piemēram, runas ģenerēšanai un citiem risinājumiem [36]. Viens no izcilākajiem piemēriem ir BERT modelis, kas tiek trenēts, izmantojot divas galvenās pieejas. MLM (Masked Language Model) pieejā daļa no ievades vārdiem tiek aizstāti ar masku, un modelim prognozē vārdu, kas atrodas maskas vietā, tādējādi apgūstot kontekstu no abām pusēm. NSP (Next Sentence Prediction) pieejā modelis mācās noteikt, vai viena teikuma secība seko citai, tādējādi uzlabojot izpratni par teikumu attiecībām. Šādi trenēts, BERT spēj efektīvi pārveidot teksta ievades par augstas kvalitātes slēptajām reprezentācijām [8].

Dekodētājs (Decoder) pārveido iekodētās reprezentācijas atpakaļ oriģinālajā vai kādā citā izvades datu formātā, piemēram, tekstā, bildē vai audio signālā. Arī šo modeļu pamatā var tikt izmantotas RNN, CNN un Transformer struktūras. Bieži šāda veida modeļi tiek apmācīti, izmantojot skolotāja piespiedi (teacher forcing) metodi, kur iepriekšējie patiesie tokeni tiek ņemti kā ievade, lai ģenerētu pašreizējo tokenu vai arī konkurējošo (adversarial) metodi, kur pamatā ir divi konkurējoši neironu tīkli - ģenerators, kas mēģina

radīt datus, kas līdzinās patiesiem datiem, bet diskriminators mēģina atšķirt ģenerētos datus no īstajiem [36]. Lielu populatirāti iemantojis ir GPT, kas arī ir šajā ietvarā bāzēts modelis [28].

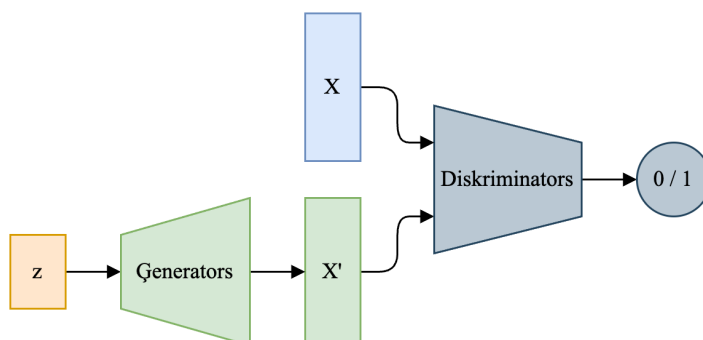
Šos abus ietvarus var apvienot kopā, lai veidotu **iekodētāja-dekodētāja** (Encoder-Decoder) ietvaru, kas tiek izmantots gadījumos, ja ievades dati ir laika rinda un izvade arī ir laika rinda, piemēram, tulkošanas vai teksta apkopošanas uzdevumos [36].

Runas sintēze pamatā ir ģeneratīvs uzdevums. Mašīnmācīšanās kontekstā tajā tiek izmantoti dziļie ģeneratīvie modeļi. Tos var iedalīt četrās visbiežāk izmantotajās grupās pēc ietvaru tipa - GAN, VAE, Plūsma un Difūzija.

GAN (Generative Adversarial Network) - ģeneratīvi konkurējošie tīkli ir ģeneratīvo modeļu veids, kas ģenerē datus no nejauša vektora, kas tiek ņemts kā paraugs no normālā sadalījumā. Šī tipa ietvaru plaši izmanto daudzos datu ģenerēšanas uzdevumos, tai skaitā audio ģenerēšanā, kur viens no zināmākajiem modeļiem ir HiFi-GAN [17]. Šo modeļu pamatā ir divu konkurējošu tīkli (skatīt 7. attēlu) – ģenerators un diskriminators, kas savā starpā sacenšas. Ģenerators mēģina radīt datus, kas līdzinās patiesiem datiem, bet diskriminators mēģina atšķirt ģenerētos datus no īstajiem. Šajā procesā tiek izmantota *minmax* spēles teorija (*game theory*), kur ģenerators mēģina maksimizēt diskriminatora kļūdas varbūtību, bet diskriminators mēģina to minimizēt, kā rezultātā GAN kļūdas funkciju var formulēt kā:

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x; \phi)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z; \theta); \phi))]$$

kur θ un ϕ attiecīgi apzīmē ģeneratora un diskriminatora parametrus, un p_{data} un p_z apzīmē patieso un normālo sadalījumu. GAN arhitektūra paredz abu tīklu trenēšanu vienlaikus, izmantojot atpakaļizplatīšanas algoritmu. Lai arī šī arhitektūra ir pierādījusi sevi gan reālistisku bilžu, gan runas audio sintēzē, šo modeļu trenēšana nav viegla, un bieži vien mēdz būt lēna un nestabila [11].

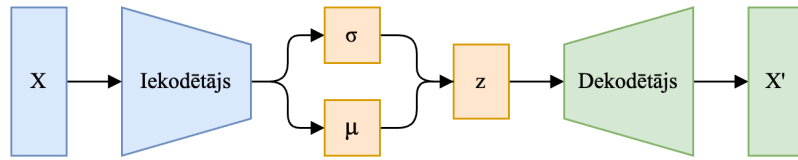


7. att. GAN arhitektūras modelis, kur z - latentais vektors, X - reālie dati, X' - ģenerētie dati un $0/1$ - diskriminatora minējums

VAE (Variational Auto-Encoder) - variacionālie autoenkoderi ir ģeneratīvo modeļu veids, kura pamatā ir divi komponenti - iekodētājs un dekodētājs (skatīt 8. attēlu). Naivā autoenkodera gadījumā iekodētājs ieejas datus saspiež latentajā vektorā, savukārt dekodētājs šo latentu vektoru cenšas rekonstruēt pēc iespējas precīzāk. Tā kā autoenkoderim latentajā telpā nav regularizācijas, latentā telpa ir ārkārtīgi neregulāra un nevienmērīga, kas nozīmē, ka daži punkti, kas ir tuvu viens otram latentajā telpā, var dekodēt uz datu punktiem, kas ir ļoti atšķirīgi datu telpā, kā arī būs punkti kuru rezultāts būs bezjēdzīgs. Arī enkodera un dekodera augstās sarežģītības dēļ, var rasties pārmācīšanās. Lai risinātu šo problēmu, VAE latentajā telpā ievieš sadalījumu, kas nozīmē, ka iekodētājs saspiež ieejas datus divos vektoros μ un σ , veidojot sadalījumu $p(\mu|\sigma)$. Dekodētājs tad ņem paraugu $z \sim p(\mu|\sigma)$ no šī sadalījuma, kuru tālāk rekonstruē kā izejas datus. Papildus tiek ieviests regularizācijas termins latentajā telpā, kas nodrošina, ka sadalījums veido nepārtrauktu un pilnīgu telpu. Tā rezultātā kļūdas funkcija ir šāda:

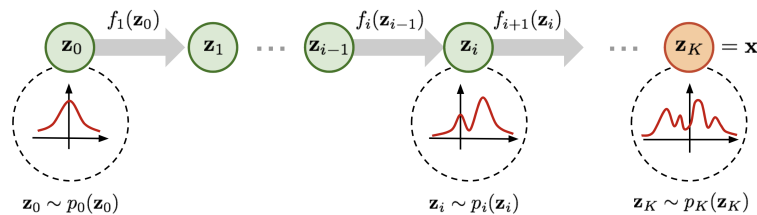
$$L = \|x - \text{dec}(z)\|^2 + \text{KL}(N(\mu_x, \sigma_x) \| N(0, 1)),$$

kur $\|x - \text{dec}(z)\|^2$ ir rekonstrukcijas kļūda, reālā un ģenerētā rezultāta Eiklīda distance un $\text{KL}(N(\mu_x, \sigma_x) \| N(0, 1))$ ir KL (*Kullback-Leibler*) novirze, kas tuvinā latentās telpas sadalījumu normālam sadalījumam [36].



8. att. VAE arhitektūras modelis, kur μ - sadalījuma vidējā vērtība, σ - sadalījuma standartnovirze, z - latentais vektors, X - reālie dati, X' - ģenerētie dati

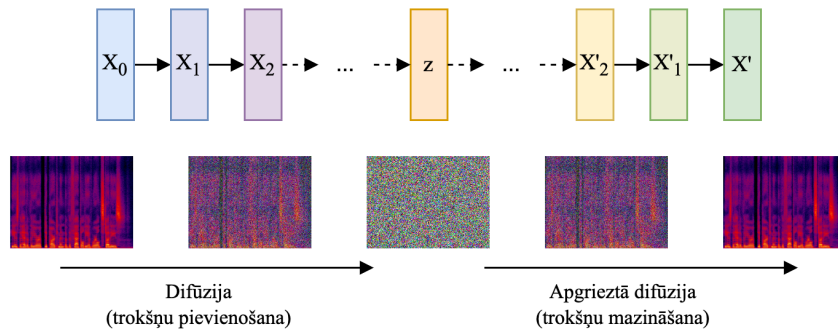
Plūsmas (Flow) - ģeneratīvo modeļu veids, kas veic datu transformāciju no vienkārša, labi zināma sadalījuma, piemēram, normālā sadalījuma, uz sarežģītāku datu sadalījumu, kas atbilst kādiem reāliem datiem. Šo modeļu pamatā ir secīga invertējamo kartēšanas funkciju ķēde $x = f_0 \circ f_1 \circ \dots \circ f_k(z)$, kur $z \sim N(0,1)$ ir datu punkts no normālā sadalījuma un x ir datu punkts reālajā sadalījumā (skatīt 9. attēlu).



9. att. Plūsmas modeļu process vienkārša sadalījuma pārveidošanai sarežģītākā [39]

Plūsmas modeļu apmācībā, tiek optimizēta logaritmiskā ticamība (*log-likelihood*), kas ir statistikas metrika, kas mēra, cik labi modelis atbilst novērotajiem datiem. Optimālās ticamības vērtības tiek iegūtas, pielāgojot modeļa parametrus tā, lai maksimizētu iespējamību, ka dati nāk no dotā modeļa sadalījuma. Šāda tipa modeļu parametri tiek apmācīti optimizējot logaritmisko ticamību (*log-likelihood*). Viens no pirmajiem runas sintēzes modeļiem, kas izmantoja plūsmu un guva labus rezultātus, bija WaveGlow [27].

Difūzijas (Diffusion) - ģeneratīvo modeļu veids, kas balstās uz diviem procesiem - uz priekšu vērsta procesa un atpakaļvērsta procesa (skatīt 10. attēlu). Uz priekšu vērsta process izmanto Markova ķēdes principu, kas pārveido datus x_0 par iepriekšējo x_T pamazām pievienojot Gausa sadalījumā bāzētu troksi pēc konkrēta trokša grafika β , kur $0 < \beta_1 < \dots < \beta_T < 1$. Atpakaļvērsta process pakāpeniski pārveido troksni $x_T \sim N(0, 1)$ atpakaļ uz datiem x_0 izmantojot apmācāmu modeli. Kā viens no galvenajiem trūkumiem ir paraugu ģenerēšanas laiktelpīgums [36]. Par spīti tam, pēdējā laikā difūzijā balstīti rīki ir guvuši lielus panākumus attēlu ģenerēšanā no teksta, kur kā viens no zināmākajiem ir Midjourney [13]. Arī runas sintēzē tiek pielietots šis process un kā viens no zināmākajiem šāda tipa modeļiem runas sintēzē ir Grad-TTS [26], kas veiksmīgi izmanto difūzijas principus ģenerējot viļņformas audio.



10. att. Difūzijas modeļa process izmantojot audio mel spektrogrammu

1.4 Rādītāji

WER un CER rādītāji ir plaši izmantoti dažādās sistēmās, sākot no teksta atpazīšanas līdz automātiskajām runas atpazīšanas (ASR) sistēmām. Šie rādītāji palīdz objektīvi novērtēt, cik precīzi un atbilstoši sistēma ir spējusi ģenerēt un atpazīt tekstu, salīdzinot to ar sākotnējo versiju. Izmantojot šos rādītājus, tiek mērīti vārdu izlaišanas, ievietošanas un aizvietošanas gadījumu skaits. Abi rādītāji tiek aprēķināti, izmantojot šādu formulu:

$$\text{CER vai WER} = \frac{S + D + I}{N}$$

kur S - aizvietošanas skaits, D - dzēšanu skaits, I - ievietošanas skaits un N - kopējais

simbolu vai vārdu skaits.

Lai aprēķinātu šos rādītājus runas sintēzes sistēmās, sintezētie audio faili jāpārverš atpakaļ teksta formā, izmantojot kādu ASR sistēmu. Pēc šī procesa pabeigšanas, tiek salīdzināts sākotnējais teksts ar ASR sistēmas atpazīto tekstu, un tiek veikts rādītāju aprēķins [30].

MOS ir plaši izmantots rādītājs telekomunikāciju un runas sintēzes jomās, lai novērtētu audio kvalitāti. Šis rādītājs balstās uz cilvēku subjektīviem vērtējumiem. Respondenti novērtē audio kvalitāti Likerta skalā no 1 līdz 5, kur 1 nozīmē zemu kvalitāti, bet 5 - augstu kvalitāti.

MOS rādītājam ir arī dažādi paveidi, kas tāpat kā MOS rādītājs, balstās uz subjektīvu vērtējumu pamata. Viens no populārākajiem paveidiem ir CMOS (Comparative MOS). Tas ir salīdzinošais vidējais viedokļa rādītājs, kas tiek noteikts, liekot respondentiem vērtēt kvalitātes atšķirību starp diviem paraugiem. Arī šī rādītāja novērtēšanai bieži tiek izmantota Likerta skala no -3 līdz +3, kur negatīvie skaitļi nozīmē, ka pirmais paraugs ir sliktāks, savukārt pozitīvie, ka labāks.

Dažādās situācijās ir iespējams šos rādītājus pielāgot, liekot respondentiem vērtēt nevis tikai kvalitāti, bet kādas citas pazīmes, piemēram, dabīgumu, trokšņus vai skaļumu. [34, 36, 30].

1.5 VITS

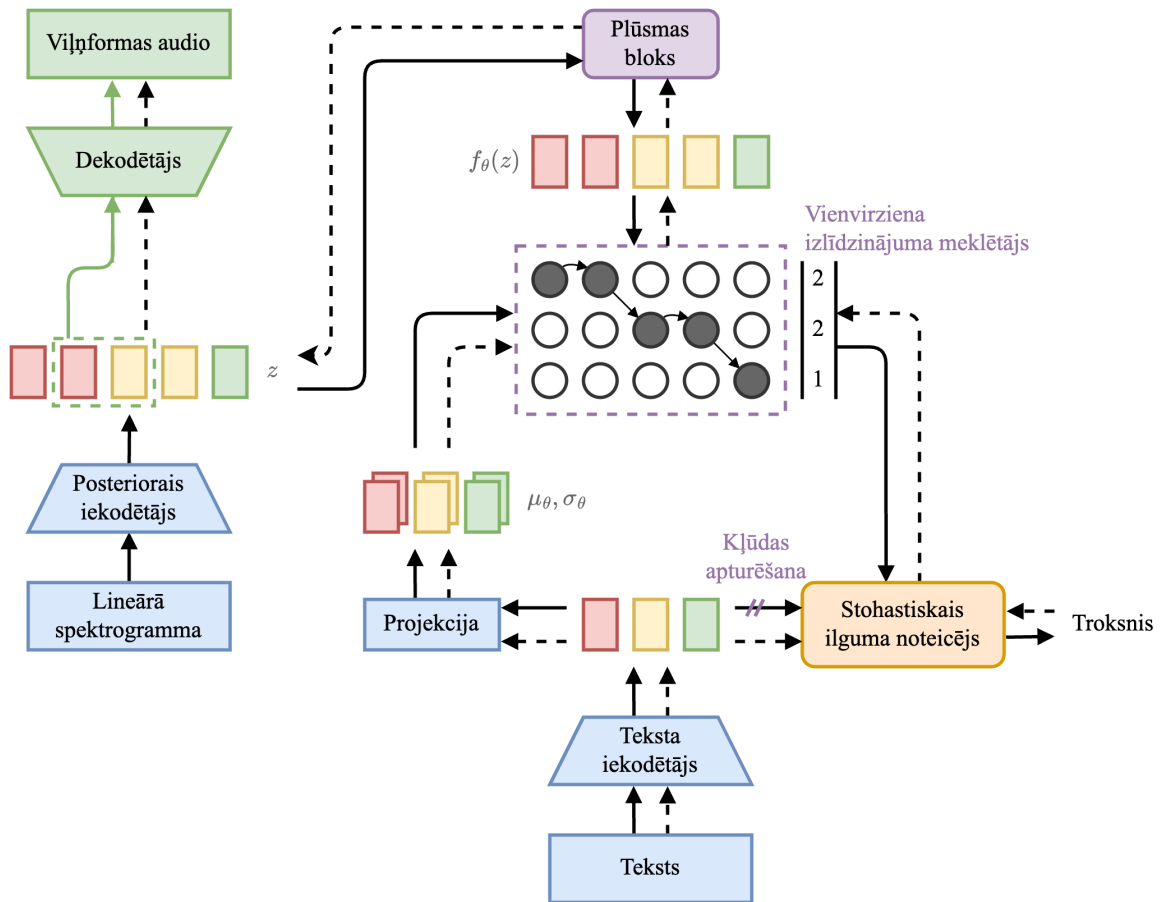
VITS (Variational Inference Text-to-Speech), jeb variacionālās inferences runas sintēzes modelis ir uzrādījis augstus rādītājus gan sintezētās runas kvalitātē, gan precizitātē [31]. Modelis apvieno elementus gan no VAE, gan GAN, gan Flow ietvaru struktūrām, lai rezultātā izveidotu modeli, kas spēj sintezēt augstas kvalitātes un dabiski skanošu runu.

VITS arhitektūra (skatīt 11. atēlu) sastāv no vairākām daļām, kas tālāk tiks apstrakstītas sīkāk:

- Teksta iekodētājs - Transformer balstīts iekodētājs, kas veic teksta iekodēšanu latentajos vektoros no kuriem tiek veidots sākotnējais varbūtības sadalījums.
- Stohastiskais ilguma noteicējs (Stochastic Duration Predictor) - šīs daļas mērķis ir paredzēt cik ilgu laiku aizņems izrunāt vienu teksta elementu - fonēmu vai burtu. Tas izmanto teksta latentu vektoru un normālā sadalījuma troksni, kas palīdz imitēt dabīgo runas mainīgumu un tā pamatā ir neirāla splaina plūsmas (Neural spline flows), [10].
- Vienvirziena izlīdzinājuma meklētājs (Monotonic Alignment Search) [14] - balstās uz dinamiskās programmēšanas algoritma, lai noteiktu katra burtā vai fonēmas latentā vektora attiecību pret runas latentajiem vektoriem. Algoritms nodrošina,

ka izlīdzinājums ir vienvirziena un bez pārlekšanas, līdzīgi kā cilvēki lasa secīgi, neizlaižot vārdus.

- Plūsmas (Flow) bloks - savieno sākotnējo varbūtības sadalījumu, kurā ir iekodēta relatīvi vienkārša linvistiskā informācija, ar posterioro varbūtības sadalījumu, kurā iekodēta relatīvi sarežģītāka runas sintēzei nepieciešamā informācija. Šis bloka pamatā ir apgriežamas transformācijas funkcijas, kuru parametri apmācības laikā tiek optimizēti.
- Posteriorais iekodētājs - veic reālās spektrogrammas iekodēšanu normālā sadalījuma latentajā telpā, izmantojot σ un μ vektorus. Šī daļa balstās uz WaveNet [24] daļām, līdzīgi kā tas ir darīts WaveGlow [27] modelī.
- Posteriorais dekodētājs - veic latentu vektoru pārvēršanu viļņformas runā. Balstīts uz HiFi-GAN [17] ģeneratoru.
- Diskriminators - veic reālās un sintezētās runas atšifrēšanu. Balstās uz HiFi-GAN [17] modelī ierosināto diskriminatora arhitektūru.



11. att. VITS arhitektūras pārskats. Ar nepārtrauktu līniju apzīmēta datu plūsma apmācības laikā. Ar raustītu līniju apzīmēta datu plūsma inferences laikā.

Apmācības laikā modelis kā ievaddatus saņem lineāro spektrogrammu, kas tiek iekodēta z latentajā vektorā izmantojot posterioro iekodētāju. No šī vektora tiek paņemta tikai maza daļa, segments, kas tālāk tiek pārvērsts viļņformas audio izmantojot dekodētāju. Tas tiek darīts apmācības ātruma uzlabošanas nolūkos. Sintezētais audio tālāk tiek nodots diskriminatoram. Apmācības laikā kā ievaddati tiek saņemti arī teksts, kas tiek iekodēts latentajā vektorā un veidota projekcija normālā sadalījuma telpā, izmantojot vidējās vērtības un standartnovirzes vektorus. Tālāk caur plūsmas bloku apstrādātais z vektors un teksta iekodējuma projekcija tiek apstrādāta, izmantojot vienvirziena izlīdzinājuma meklētāju, kur, izmantojot x algoritmu tiek aprēķināta iespējamība katram iekodētā teksta vektoram pret z vektoru un atrasts vismazāk maksājošais izlīdzinājums. Izlīdzinājuma rezultāts jeb z un q vektoru kartējums tiek nodots stohastiskajam ilguma noteicējam. Stohastiskais ilguma noteicējs savukārt apstrādā vektoru kartējumu un teksta latentu vektoru. Inferences laikā modelis kā ievadu saņem tikai tekstu un tiek izlaista lineārās spektrogrammas iekodēšanas daļa, jo inferences laikā plūsmas bloks tiek darbināts pretējā virzienā.

Modeļa apmācībai tiek izmantotas sekojošās kļūdas funkcijas modeļa svaru optimizācijai:

- FM kļūda (*Feature Loss*) - pamatā tiek izmantots MAE, kas tiek summēts pāri visām diskriminatora starpposmu iezīmju kartēm. Summu salīdzinājums tiek aprēķināts izmantojot sintezētās runas iezīmju kartes un reālas runas kartes. Šīs kļūdas aprēķināšanas laikā diskriminatora svāri ir fiksēti, tie netiek ietekmēti.
- Mel spektrogrammu kļūda - arī par pamatu tiek izmantots MAE. Kļūdas vērtība tiek rēķināta par pamatu ņemot ģenerētā un reālā audio normalizētās mel spektrogrammas un summējot attiecīgo vienību atšķirības.
- Ilguma kļūda (*Duration loss*) - mērķis ir maksimizēt teksta elementu ilgumu loģiskās ticamības variāciju zemākās zobežas (*Variational Lower Bound*) attiecībā pret doto tekstu. Šī kļūda tiek aprēķināta, ņemot vērā latentos mainīgos u un v , kas ievieš dabīgu mainību sintezētajā runā, jo dažādi cilvēki var izrunāt vienus un tos pašus vārdus ar nelielām laika atšķirībām.
- KL kļūda (*Kullback-Leibler loss*) - nobīde starp posterioro un pirmatnējo latentu vektoru sadalījumu. Šī zuduma funkcija tiek izmantota, lai novērtētu, cik labi pirmatnējais, iekodētais teksta īpašību latentais sadalījums sakrīt ar posterioro, runas īpašību sadalījumu.
- Diskriminatora kļūda - pamatā tiek izmantots MSE un balstās uz LSGAN [22] (*Least Squares Generative Adversarial Networks*), jeb mazāko kvadrātu ģeneratīvajiem konkurējošiem tīkliem. MSE zaudējuma funkcija tiek izmantota, lai izvairītos

no gradientu izzušanas problēmas, kas var rasties, ja diskriminators kļūst pārāk pārliecināts par saviem lēmumiem. Diskriminatora zaudējums tiek aprēķināts, salīdzinot diskriminatora noteiktās vērtības reālai un ģenerētai runai, tādējādi veicinot precīzāku un efektīvāku modeļa apmācību, kas uzlabo sintezētās runas kvalitāti.

- Ģeneratora kļūda - optimizē ģeneratora svarus, lai uzlabotu sintezētās runas kvalitāti. Šī kļūda tiek aprēķināta, izmantojot vidējo kvadrātisko kļūdu (MSE) starp diskriminatora prognozēm un ģenerēto runu. Lai samazinātu kļūdas vērtību, ģeneratoram jāspēj radīt runas paraugus, kas diskriminatoram šķiet autentiski. Ģeneratora kļūdas aprēķināšanas procesā diskriminatora svāri tiek fiksēti, lai nodrošinātu, ka tikai ģeneratora svāri tiek pielāgoti.

Rezultātā tiek iegūta kopējā kļūda, kas izskatās šādi:

$$L_{vae} = L_{mel} + L_{kl} + L_{dur} + L_{adv}(G) + L_{fm}(G)$$

2 Sistemātiskā literatūras analīze

Darba veikšanai nepieciešams veikt padziļinātu izpēti un salīdzinājumu vairākām ar runas sintēzi saistītām tehnoloģijām. Šajā nodālā tiks apskatīti pētījumi par labākajiem runas sintēzes modeļiem angļu valodā, esošajiem runas sintēzes rīkiem latviešu valodā, dažādas audio attīrīšanas un uzlabošanas metodes un rīki, kā arī runas stila pārnese ierunāto audio balss vienādošanai.

2.1 Runas sintēze angļu valodā

Balstoties uz kursa darbu "Angļu valodas runas sintēzes modeļu salīdzinājums", kā galvenais apmācāmais runas sintēze modelis latviešu valodai tika izvēlēts VITS. Tas izcēlās ar zemo precizitātes kļudu - CER 1.48%, kas norāda uz augstu precizitāti teksta pārveidošanā runā salīdzinājumā ar citiem modeļiem. Augstā runas sintēzes precizitāte ir būtiska, lai nodrošinātu, ka sintētiskā latviešu runa ir skaidra un saprotama, kas ir īpaši svarīgi valodai ar specifiskām fonētiskām un gramatiskām niansēm. Arī tā kvalitātes rādītāji bija vieni no augstākajiem, NISQA kvalitātes rādītājs bija 3.07 un dabiskums 4.29 [31].

2.2 Runas sintēze latviešu valodā

Runas sintēze latviešu valodā ir nozīmīgs pētījumu un attīstības virziens, kas ietver dažādu tehnoloģiju un pieeju izmantošanu, lai radītu augstas kvalitātes sintezēto runu. Šajā apakšnodalā tiek apskatīti vairāki ievērojami runas sintēzes modeļi un sistēmas, kas ir pieejamas latviešu valodā. Piemēram, AiLab Ilvars un Coqui, kas balstās uz VITS modeli, kā arī eSpeak un Hugo.lv, kas piedāvā dažādus risinājumus, kas ir pieejami gan pētniekiem, gan plašākai sabiedrībai, tādējādi veicinot runas tehnoloģiju attīstību un pielietojumu latviešu valodā.

- AiLab Ilvars - LU Matemātikas un informātikas institūta (LUMII) Mākslīgā intelekta laboratorijā radīts runas sintēzes modelis. Balstās uz mašīnmācīšanās tehnoloģiju un izmanto VITS modeli, kurš ir apmācīts uz viena runātāja datiem, vīrieša balsi. Apmācības datu kopa kopā satur 25 stundas garu audio, un ir veidots uz audiogrāmatu bāzes. Šī modeļa licence ir diezgan ierobežota - to ir iespējams izmantot tikai akadēmiskām un nekomerciālām darbībām [6].
- COQUI¹ - atvērta pirmkoda runas sintēzes sistēma, kurā pieejami dažādi mašīnmācīšanās tehnoloģijā bāzēti runas sintēzes modeļi, dažādās valodās, tai skaitā arī latviešu. Šī sistēmas arhitektūra ir ļoti elastīga un plaša dokumentācija padara to

¹<https://github.com/coqui-ai/TTS>

par vērtīgu rīku pētniekiem un izstrādātājiem, kuri strādā pie runas sintēzes projektiem, atvieglojot pielāgotu TTS risinājumu izveidi. Latviešu valodā ir pieejams VITS viena runātāja modelis sievietes balsī. Sīkāka informācija par modeļa apmācību nav atrodamā, pieejama tikai informācija par to, ka modelis apmācīts izmantojot Common Voice datu kopu [19]. Bet, ņemot vērā to, ka Common Voice datu kopā ir salīdzinoši maz viena runātāja ieraksti un nav detalizētāka informācija par datu priekšapstrādi, šī informācija varētu būt apšaubāma.

- eSpeak - atvērtā pirmkoda runas sintēzes sistēma, kas pārveido rakstīto tekstu izrunātos vārdos. Izstrādāts, lai atbalstītu vairākas valodas, tostarp latviešu valodu. eSpeak ir pazīstams ar savu kompakto izmēru un salīdzinoši augstas precizitātes balss izvadi. Tā pamatā ir dažādas sintēzes metodes, ka piemēram, formantu sintēzes metode, kas modelē cilvēka balss trakta rezonanses frekvences, un arī jau iepriekš aprakstītā difonu savienošanas metode. Neskatoties uz sintezētās runas robotisko toni, eSpeak ir ļoti daudzpusīgs un pielāgojams, ļaujot lietotājiem modificēt balsis un veidot jaunas. To plaši izmanto dažādās lietojumprogrammās, tostarp palīgtechnoloģijās cilvēkiem ar redzes traucējumiem, un to var integrēt programmatūrā dažādās operētājsistēmās [1].
- Hugo.lv¹ - Latvijas valsts pārvaldes valodas tehnoloģiju platforma, kas ir brīvi pieejama visiem Latvijas iedzīvotājiem. Šī platforma nodrošina automatizētu tulkošanas, runas atpazīšanas un runas sintēzes pakalpojumus, kā arī piedāvā dažādus rīkus daudzvalodu atbalstam e-pakalpojumos [2]. Tomēr, neskatoties uz plašo funkcionalitāti, nav pieejama detalizēta informācija par izmantotajām runas sintēzes tehnoloģijām. Klausoties sintezētās runas paraugus var tikai izdarīt minējumu, ka šī varētu būt difonu savienošanā bāzēta metode, jo veido līdzīgu skanējumu, kā MaryTTS [32] sintezētā runa.

Latviešu valodas runas sintēzē viens no lielākajiem trūkumiem ir kvalitatīvu datu kopu trūkums. No publiski pieejamajiem, viens no plašākajiem un kvalitatīvākajiem ir Common Voice. Tā primārais mērķis nav runas sintēzes uzdevumi, bet automātiskās runas atpazīšanas (ASR) uzdevumi. Ir arī izstrādātas datu kopas tieši priekš runas sintēzes [5], bet tās nav publiski pieejamas. Vairākām datu kopām ir norādītas saites, taču tās vairs nav pieejamas. Kā piemēru var minēt <http://runa.korpuss.lv/> Vēl no publiski atrodamajām datu kopām var pieminēt LaRko [3], bet arī šī vairāk piemērota tieši ASR uzdevumiem.

¹<https://hugo.lv>

2.3 Audio attrokšņošana un uzlabošana

Lai iegūtu augstākas kavalitātes balss audio ierakstus, ir izstrādāti vairāki modeļi, kas izmanto dziļās mašīnmācīšanās metodes. Tika apskatīti un eksperimentāli pārbaudīti vairāki nozīmīgi modeļi:

- Demucs [7] modelis balstās uz kodētāja-dekodētāja arhitektūru ar izlaistiem savienojumiem (skip-connections). Tas tiek optimizēts gan laika, gan frekvenču domēnos, izmantojot vairākas zaudējumu funkcijas. Kodēšanas posmā ir vairāki meta-slāņi jeb "kodēšanas slāņi". Katrs kodēšanas slānis sastāv no 1D konvolūcijas slāņa, ReLU slāņa, vēl viena 1D konvolūcijas slāņa un vārtu rekurentā bloka (GRU) slāņa. Katra meta-slāņa izvade tiek padota gan nākamajam kodēšanas slānim, gan atbilstošajam "dekodēšanas slānim" modeļa dekodēšanas posmā, ko sauc par "izlaistu savienojumu". Empīriski pierādījumi liecina, ka Demucs spēj noņemt dažāda veida fona troksni, tostarp stacionāru un nestacionāru troksni, kā arī telpu reverberāciju. Papildus tiek piedāvātas datu paplašināšanas metodes, kas tiek piemērotas tieši neapstrādātai viļņformai, kas vēl vairāk uzlabo modeļa veikspēju un vispārināšanas spējas.
- Resemble ¹ - ir uzņēmums, kas specializējas runas uzlabošanas tehnoloģijās, un ir izstrādājis rīku, kas izmanto neironu tīklus, lai uzlabotu runas kvalitāti. Ir pieejams arī atvērtā koda risinājums, kas ļauj izmantot Resemble uzlabošanas modeļus citās platformās. Tas izmanto UNet modeli, kas pieņem trokšņainu kompleksu spektrogrammu kā ievadi un paredz amplitūdas masku un fāzes rotāciju, efektīvi izolējot runu no sākotnējā audio. Uzlabotājs (enhancer), kas papildus uzlabo audio kvalitāti, atjaunojot audio kropļojumus un paplašinot audio joslas platumu. Tas ir latentais nosacītās plūsmas atbilstības (CFM) modelis, kas sastāv no Implicit Rank-Minimizing Autoencoder (IRMAE) un CFM modeļa, kurš paredz latentos mainīgos.
- MP-SENet [21] ir jauns runas uzlabošanas tīkla modelis, kas vienlaicīgi tieši attīra amplitūdas un fāzes spektrus. MP-SENet izmanto kodeka arhitektūru, kurā kodētāju un dekodētāju savieno ar konvolūciju papildināti transformatori. Kodētājs kodē laika-frekvenču attēlojumus no ievades trokšņainajiem amplitūdas un fāzes spektriem. Dekodētājs sastāv no paralēla amplitūdas maskas dekodētāja un fāzes dekodētāja, tieši atjaunojot tīros amplitūdas spektrus un tīri ietītus fāzes spektrus, attiecīgi iekļaujot apmācāmu sigmoīda aktivizāciju un paralēlu fāzes novērtēšanas arhitektūru. Amplitūdas spektros, fāzes spektros, īslaicīgos kompleksos spektros un

¹<https://github.com/resemble-ai/resemble-enhance>

laika domēna viļņos definēti vairāku līmeņu zudumi tiek izmantoti, lai kopīgi apmācītu MP-SENet modeli. Eksperimentu rezultāti rāda, ka piedāvātais MP-SENet sasniedz PESQ 3,50 publiskajā VoiceBank+DEMAND datu kopā un pārspēj esošās modernās runas uzlabošanas metodes

- "Adobe Podcast Enhance"¹ - ir Adobe izstrādāts rīks fona trokšņu noņemšanai un kopējās audio kvalitātes uzlabošanai un ir paredzēts tieši raidierakstu, kur pārsvarā tiek runāts, skaņas uzlabošanai. Reģistrējoties ir pieejama bezmaksas versija, kurā dienas laikā var apstrādāt līdz 1 stundai audio, bet maksas versijā, līdz 4 stundām.

2.4 Runas stila pārnese

Lai iegūtu vienādu balss toni visiem datu kopas runātājiem, tādejādi uzlabojot sintēzes modeļa rezultātus, ir izstrādāti vairāki modeļi, kas veic balss toni pārveidi. Tika apskatīti un eksperimentāli pārbaudīti sekojoši modeļi:

- Free-VC - atvērtā koda un svaru runas stila pārnese rīks, kas balstās uz VITS modeļa arhitektūru un pielāgo to runas pārveidošanas vajadzībām. Šī sistēma atšķiras no citām balss pārveidošanas sistēmām ar to, ka to nav nepieciešams atsevišķi trenēt, tā spēj darboties ar vēl neredzētiem runātājiem un balstās uz viena šāviena principu. No VITS modeļa šis modelis atšķiras ar divām galvenajām īpašībām un izmaiņām - teksta ievades vietā tiek ievadīta mērķa runātāja audio spektrogramma, kas tiek iekodēta runātāja balss iezīmju latentajā vektorā un tiek pielietots WavLM, kas iekodē bāzes runu jeb runu, kas tiks pārveidota mērķa runātāja tonī. Autoru veiktajos testos modelis MOS rādītājos redzētu uz redzētu runātāju gadījumā sasniedz 4.01, neredzētu uz redzētu - 4.08 un neredzētu uz neredzētu - 4.06 [20].
- so-vits-svc² - atvērtā koda un bāzes svaru runas stila pārnese rīks, kas arī balstās VITS modeļa arhitektūrā, bet pārveido to vēl vairāk. Izmanto vairāku modeļu apvienojumu, kur galvenās daļas ir:
 - Whisper [29] - tiek izmantots fonēmu izgūšanai no audio ieraksta.
 - HuBERT [12] - nosaka runas prosodiju, jeb runas ritmiskās iezīmes.
 - CREPE [15] - nodoršina precīzu balss toņa augstumas noteikšanu zīmantojot pilnībā konvolucionālu (Fully Convolutional) arhitektūru.

Šī sistēma, atšķirībā no Free-VC, nedarbojas uz viena šāviena principa, un ir jāam-pācā uz mērķa runātāja audio datiem.

¹<https://podcast.adobe.com/enhance>

²<https://github.com/svc-develop-team/so-vits-svc>

- RVC ¹ - šis arī ir atvērtā koda un bāzes svaru runas pārneses rīks, kas balstās VITS modeļa arhitektūrā. Šī sistēma pirms tās lietošanas ir jāapmāca, izmantojot mērķa runātāja audio. Atšķirībā no so-vits-svc, tiek pielietots iegultņu izgūšanas mehānisms, ko iegūst no mērķa runātāja balss, un tas raksturo specifiskās balss īpašības.

Modeļu salīdzināšanas rezultāti tika aprēķināti, izmantojot kosinusa attālumu, kā arī dabiskuma un kvalitātes rādītājus, kas redzami 4. tabulā.

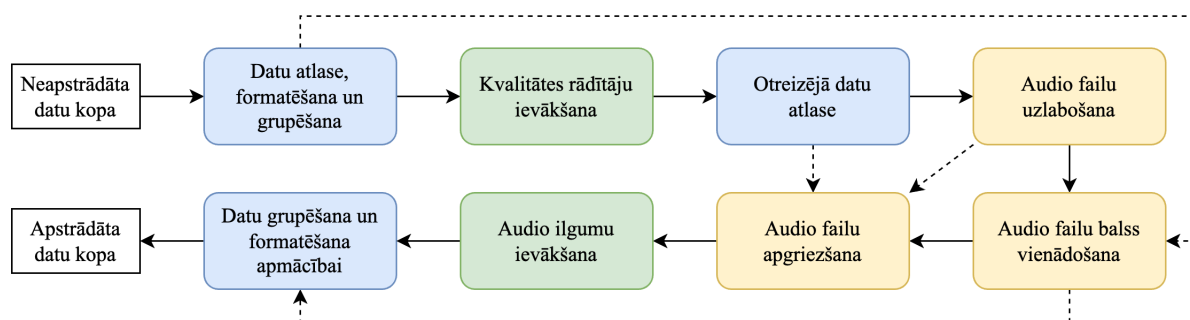
¹<https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI>

3 Metodoloģija

3.1 Datu kopas un to priekšapstrāde

Datu kopu sagatavošanas procesā tika izveidota datu priekšapstrādes plūsma, lai sagatavotu, uzlabotu un iegūtu datus apmācības procesam nepieciešamajā formātā. Datu priekšapstrādes plūsma attēlota 12. attēlā.

Šī darba ietvaros tika izmantotas divas datu kopas - Common Voice v17 ¹ latviešu valodas datu kopā un privātā Asya.ai datu kopā. Common Voice v17 ir izlaista 2024. gada 20. martā, kopējais apstiprināto audio garums ir 223 stundas, ko ierunājušu 5712 dažādi runātāji un satur 28 110 dažādus apstiprinātu tekstu fragmentus. Asya.ai datu kopas audio garums ir 86 stundas un tas satur 45 058 dažādus teikumus. Kopā tika veiktas piecas iterācijas, ar katru iterāciju uzlabojot datu apstrādes un sagatavošanas procesu. Datu apstrādē izmantotais kods ir pieejams GitHub atvērtā pirmkoda glabātnē ².



12. att. Datu priekšapstrādes plūsma. Ar zilu krāsu apzīmēti posmi kuros tika veiktas datu kopas metadatu modifikācijas, ar zaļu - papildus parametru ievākšana, ar dzeltenu - audio failu modifikācijas. Ar raustītu līniju apzīmēti alternatīvie priekšapstrādes ceļa posmi.

Pirmā iterācija – balstījās uz pilno Asya.ai datu kopu. Sākumā tika veikta teikumu dublikātu un sliktu transkriptu paraugu filtrācija. Tā kā datu kopā nav identificēti atsevišķi runātāji, visi audio faili tika pārveidoti uz viena runātāja balsi, izmantojot FreeVC ³ balss pārveidošanas modeli. Noslēgumā datu kopā tika sagatavota modeļa apmācībai, veikts treniņa un validācijas kopas sadalījums. Rezultātā iegūtās datu kopas parametri redzami 1. tabulā. Pirmās apmācības noslēgumā tika detalizētāk pārbaudīta datu kopā un atklātas transkriptu nesakritības, runātāju pārklāšanās, audio fragmenti ar fona mūziku un sliktas kvalitātes audio ieraksti.

Otrā iterācija – balstījās uz Common Voice v17 datu kopu. Sākumā tika veikta datu priekšapstrāde - statistikas ievākšana, atlasot runātājus un to audio failus, kuru

¹<https://commonvoice.mozilla.org>

²<https://github.com/krsaulitis/latvian-speech-synthesis-2024>

³<https://github.com/OlaWod/FreeVC>

kopējais ierakstu garums sasniedza vismaz 10 minūtes. Tālāk tika veikta atlasīto audio kvalitātes rādītāju ievākšana, izmantojot NISQA modeli. Runātājiem, kuru kopējais audio garums pārsniedza 30 minūtes, tika atlasītas tikai labākās kvalitātes audio. Tālāk audio faili tika apstrādāti, izmantojot pydub¹ bibliotēku – tika pielietota dinamiskā kompresija un skaļuma normalizācija. Kad audio faili bija apstrādāti, tie tika apgriezti, izmantojot to pašu bibliotēku. Tika noņemtas klusās vietas audio sākumā un beigās, kā arī iespēju robežās izņemtas datorpeles klikšķa skaņas. Nākamais solis bija audio failu ilguma pārrēķināšana un saglabāšana metadatos. Noslēgumā, līdzīgi kā pirmajā iterācijā, datu kopa tika sagatavota modeļa apmācībai. Rezultātā iegūtās datu kopas parametri redzami 1. tabulā. Pēc otrās apmācības rezultātu iegūšanas tika detalizētāk pārbaudīti atlasītie audio faili un konstatēts, ka vairāki piemēri ir diezgan sliktas kvalitātes un ar fona trokšņiem.

Trešā iterācija – arī balstījās uz Common Voice v17 datu kopu. Atšķirībā no otrās iterācijas, tika ieviests audio failu uzlabošanas rīks Adobe Podcast Enhance², kas uzrādījis labākos rezultātus audio kvalitātes uzlabošanā (skatīt 3. tabulu). Šim rīkam tika nodoti visi atlasītie audio faili, pirms tam tos sagrupējot, lai ietaupītu laiku, jo failu augšupielāde jāveic manuāli. Pēc tam tika iegūti un atgrupēti uzlabotie audio faili. Tālāk šie uzlabotie audio tika apgriezti, veikta ilguma pārrēķināšana un sagatavošana apmācībai, līdzīgi kā otrajā iterācijā. Rezultātā iegūtās datu kopas parametri redzami 1. tabulā. Ar šo datu kopu tika iegūti noslēguma rezultāti vairāku runātāju modeļa apmācībai.

Ceturrtā iterācija – arī balstījās uz Common Voice v17 datu kopas. Atšķirībā no trešās iterācijas, iesākumā datu kopa tika attīrīta no teikumu dublikātiem. Pēc audio failu uzlabošanas tika veikta runātāja balss vienādošana, izmantojot RVC³ modeli. RVC modelis pirms tam tika apmācīts, izmantojot divus runātājus - vienu vīrieti un otru sievieti, no Common Voice datu kopas, kuriem bija vieni no ilgākajiem un labākās kvalitātes audio ierakstiem. Tālākais datu kopas process tika saglabāts tāds pats kā trešajā iterācijā. Rezultātā iegūtās datu kopas parametri redzami 1. tabulā.

Piektā iterācija – balstījās uz Asya.ai datu kopas fragmentu. Tā kā šajā datu kopā ir vairākas apakškopas ar dažādas kvalitātes audio un transkriptu precizitāti, no katras apakškopas tika manuāli pārbaudīti desmit ieraksti pēc nejaušības principa, lai veiktu kvalitātes atbilstības pārbaudi. Rezultātā tika izvēlēta viena no audio grāmatu datu kopām ar vismazāko CER, un dati papildus tika manuāli attīrīti un izlabotas kļūdas. Tālāk tika veikta audio failu apstrāde līdzīgi kā trešajā iterācijā. Tā kā visi audio ieraksti jau bija no viena runātāja, tad balss vienādošana tika izlaista. Audio faili tika apgriezti, ilgumi atjaunināti un gala kopa noformēta līdzīgi kā ceturtajā iterācijā. Rezultātā iegūtās datu kopas parametri redzami 1. tabulā.

¹<https://github.com/jiaaro/pydub>

²<https://podcast.adobe.com/Enhance>

³<https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI>

1. tabula. Katras iterācijas datu kopu parametri

Iterācija	Pamata datu kopa	Kopējais ilgums (stundas)	Ierakstu skaits	Runātāju skaits
1	Asya.ai kopējais	39.87	45058	1
2	Common Voice	42.36	50004	125
3	Common Voice	46.46	57165	132
4	Common Voice	12.07	9039	1
5	Asya.ai audio grāmata	14.73	5061	1

3.2 Rādītāji

Sistēmu apmācībai un turpmākai salīdzināšanai nepieciešami objektīvi rādītāji. Balstoties uz kursa darbā [31] veikto analīzi, svarīgi ir izvēlēties objektīvus rādītājus, kas runas sintēzes gadījumā ir WER un CER, kā arī objektīvais MOS.

3.2.1. WER un CER

Pirmie no izvēlētajiem rādītājiem ir WER un CER. Šie rādītāji palīdz objektīvi novērtēt, cik precīzi runas sintēzes sistēma ir sintezējusi tekstu atbilstoši ievadītajam tekstam.

Lai aprēķinātu šos rādītājus, vispirms nepieciešams katra testa soļa sintezētos audio datus pārvērst tekstuālā formā. Tam tika izmantota Whisper¹ [29] ASR modeļa latviskā versija, ko sagatavot palīdzēja Asya.ai² komanda. Audio faili tika nodoti ASR modelim, un rezultātā saņemti paredzamie teksta rezultāti. Tālāk tika veikta katra ieraksta WER un CER rādītāju aprēķināšana salīdzinājumā ar oriģinālo tekstu un katra soļa vidējā rezultāta aprēķināšana.

Lai iegūtu bāzlīnijas rezultātus, WER un CER rādītāju aprēķināšanas procedūra tika veikta arī oriģinālajiem audio ierakstiem. Tādējādi ir iespējams objektīvi salīdzināt sintezēto un oriģinālo audio rezultātus.

3.2.2. NISQA

Pārējie rādītāji tika iegūti, izmantojot NISQA modeli, kas ir automatizēta sistēma runas kvalitātes novērtēšanai. Šis modelis izmanto mašīnmācīšanās tehnoloģijas, lai analizētu un vērtētu runas signālu, nodrošinot objektīvu kvalitātes mērījumu. NISQA piedāvā vairākus atšķirīgus novērtējumus:

- Kopējā kvalitāte (MOS/quality) - vispārējs vērtējums par runas uztveramo kvalitāti, kas imitē manuālo MOS vērtējumu, kur vairāki klausītāji manuāli vērtē runas

¹<https://github.com/openai/whisper>

²<https://www.asya.ai>

kvalitāti.

- Dabiskums (naturalness) - novērtē, cik dabiska un cilvēka balsij līdzīga ir runas sintēzes sistēmas radītā runa. Šis rādītājs ir svarīgs, lai noteiktu, cik efektīvi sistēma var imitēt cilvēka runu.
- Krāsojums (coloration) - mēra nevēlamo skaņu vai frekvenču klātbūtni runā, kas kropļo skaņu.
- Trokšņi (noisiness) - nosaka fona trokšņu līmeņa pakāpi. Augsts trokšņa līmenis var ievērojami samazināt runas saprotamību.
- Pārtraukumi (discontinuity) - meklē un novērtē jebkādas nepārtrauktības traucējumus runā, piemēram, pārtraukumus vai skaņas defektus.
- Skaļums (loudness) - mēra, cik optimāls ir runas skaļuma līmenis, lai runa būtu komfortabli un skaidri saklausāma. Pārāk kluss vai pārāk skaļš līmenis nozīmē zemāku vērtējumu.

Visiem mērījumiem augstāks vērtējums nozīmē labāku runas kvalitāti [23, 31].

3.2.3. Kosinusa attālums

Kosinusa attālums (Cosine Distance), plaši izmantots rādītājs dažādās jomās, tostarp runas sintēzē. Tas mēra divu vektoru atšķirības, aprēķinot kosinusa leņķi starp tiem. Šis rādītājs ir īpaši noderīgs augstas dimensijas telpās, kur tradicionālais Eiklīda attālums var neefektīvi atspoguļot niansētas vektoru atšķirības. Matemātiski kosinusa attālums tiek iegūts no kosinusa līdzības (Cosine Similarity), kas tiek definēta, kā:

$$\text{Kosinusa līdzība} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

kur \mathbf{A} un \mathbf{B} ir salīdzināmie vektori, kur abu vektoru skalārais reizinājums tiek dalīts ar to moduļu reizinājumu un rezultātā tiek iegūts skaitlis no -1 līdz 1, kur -1 nozīmē pilnīgi pretējus virzienus, 0 nozīmē, ka to virzieni nav saistīti un 1, ka virzieni ir vienādi.

Kosinusa attālums tiek iegūts, atņemot kosinusa līdzību no viens, dodot vērtību diapazonā no nulles, kas norāda uz identisku virzienu, līdz divi, kas apzīmē pilnīgi pretējus vektorus. Runas sintēzes kontekstā kosinusa attālumu var izmantot, lai salīdzinātu dažādu runātāju audio signālu iezīmju reprezentācijas, tādējādi nosakot to runas modeļu līdzību vai atšķirību pakāpi.

$$\text{Kosinusa attālums} = 1 - \cos(\theta) = 1 - \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

Runātājā iegultnes dažādu audio failu salīdzināšanai ar kosinusa attāluma tika iegūtas izmantojot pyannote bibliotēku [25], kas, izmantojot mašīnmācīšanās modeli, veic runātāju diarizāciju un iegultņu vektoru noteikšanu.

3.3 Apmācības protokols

Balstoties uz sistemātiskās literatūras analīzē veiktajiem secinājumiem, modeļa apmācībai tika izvēlēts VITS modelis. Tam ir publiski pieejams gan tā pirmkods, gan arī modeļa svāri.¹ Iesākumā kods tika pielāgots apmācībai uz Rīgas Tehniskās Universitātes HPC (High Performance Computing) servera, pielāgoti konfigurācijas faili vieglākai eksperimentu trasējamībai kā arī pievienota kļūdu un citas informācijas reģistrēšanas, kā arī eksperimentu izsekošanas un uzraudzīšanas platforma "Weights and Biases"². Šajā platformā iespējams iesūtīt modeļa apmācības parametrus, kļūdas vērtības, rādītājus, kā arī audio un bilžu failus, kas tiek ģenerēti testēšanas solī.

Iesākumā tika veiktas pāris pārbaudes apmācības, pēc kurām tika veiktas izmaiņas, kļūdu labojumi. Arī pilno apmācību laikā tika konstatēts, ka trūkst kādi parametri un vēlākos apmācības posmos tie tika pievienoti, piemēram, 1. un 2. modeļa apmācība netika aprēķināts kvalitātes rādītājs testēšanas soļa laikā, līdz ar to tas arī netiek atspoguļots 2. tabulā.

Kopā tika veikti deviņi apmācības mēģinājumi un to galvenie parametri redzami 2. tabulā. Pārējie modeļa parametri tika atstāti tādi paši kā tika izmantoti oriģinālajā VITS pētījumā. Pēc stabilākās datu kopas atrašanas, kas notika trešajā apmācībā, tika veikti vairāki parametru eksperimenti. Ceturtajā apmācībā teksta ievade tika samainīta uz fonēmu ievadi, kas tika iegūtas izmantojot xx bibliotēku, līdzīgi kā tas tika darīts AiLab VITS modeļa apmācībā. Piektajā apmācības solī tika palielināts FM kļūdas koeficients, lai pārbaudītu tā ietekmi uz apmācību. Sestajā apmācības solī tika palielināts treniņa laikā sintezētais segmenta izmērs, lai pārbaudītu vai modelis labāk, vieglāk un stabilāk spēj apgūt vārdu uzbūves struktūras. Septītajā apmācības solī tika divkārtots slēpto kanālu skaits. Visu apmācību galvenās kļūdu un rādītāju grafiki atrodamas 1., 2. un 3. pielikumā.

¹<https://github.com/jaywalnut310/vits>

²<https://wandb.ai>

2. tabula. Modeļu apmācības parametru protokols.

Nr.	Runātāju skaits	Partijas izmērs	Simbolu tips	Slēpto kanālu skaits	Segmenta izmērs	FM kļūdas koeficients	Soļi	Epohi
1	1	32	Burti	192	8192	2	433000	2176
2	125	32	Burti	192	8192	2	561276	2808
3	132	64	Burti	192	8192	2	410800	464
4	132	64	Fonēmas	192	8192	2	238200	270
5	132	64	Burti	192	8192	10	167400	189
6	132	64	Burti	192	16384	2	140000	159
7	132	64	Burti	384	8192	2	145200	165
8	1	64	Burti	192	8192	2	237200	1694
9	1	64	Burti	192	8192	2	142600	1805

Pēc apmācību veikšanas tika veikta rezultātu apkopošana, atrasts katras apmācības labākais rezultātu punkts, kas tālāk tika izmantots salīdzināšanai ar citiem apmācītajiem modeļiem un jau eksistējošiem latviešu valodas runas sintēzes rīkiem, kas tika izvēlēti 2.2. apakšnodaļā. Lai veiktu savstarpējo salīdzinājumu, pēc nejaušības principa tika atlasīti 99 audio ieraksti un to transkriptu pāri no Common Voice v17 un Asya.ai grāmatu testa datu kopas - 33 no Common Voice vairāku runātāju testa kopas, 33 no Common Voice viena runātāja testu kopas un 33 no Asya.ai grāmatu testa kopas. Tas tika darīts, lai nodrošinātu līdzvērtīgāku salīdzināšanu starp apmācītajiem modeļiem. Visiem apmācītajiem modeļiem tika veikta visu transkriptu ievade un saglabāti rezultējošie audio faili. Katra modeļa sintezētajiem audio failiem tika veikta NISQA modeļa rādītāju aprēķināšana kā arī CER un WER metriku aprēķins. Papildus šo pašu procesu veica pārējiem atrastajiem runas sintēzes rīkiem latviešu valodā un veikts to savstarpējais salīdzinājums. Gala rezultāti redzami 6. tabulā.

4 Rezultāti

Rezultātā tika veiktas piecas datu sagatavošanas un priekšapstrādes iterācijas, kur katrā no tām tika atklāti kādi trūkumi un veikti labojumi vai uzlabojumi. Tika veikts runas uzlabošanas rīku un balss toņa pārveidošanas rīku salīdzinājums. Kopā tika veikti deviņi apmācības cikli. Spilgtākie piemēri un salīdzinājumi apkopoti publiski pieejamā mājaslapā¹.

Balss uzlabošanas rīku salīdzinājumam tika izmantots NISQA rādītājs un apkopotie rezultāti redzami 3. tabulā. Adobe Podcast Enhancer uzstāda visaugstākos rezultātus gandrīz visos rādītājos, izņemot trokšņainībā, kur nedaudz labāks ir Resemble balss uzlabošanas rīks. Līdz galam neizprotams ir fakts, ka gandrīz visi balss uzlabošanas rīki nedaudz, bet samazina ierunāto audio precizitāti.

3. tabula. Balss uzlabošanas rīku salīdzinājums balstoties uz NISQA rādītāju

Modelis	Kvalitāte			Dabiskums			Trokšņi			Precizitāte	
	Vidēji	Mediāna	STD	Vidēji	Mediāna	STD	Vidēji	Mediāna	STD	CER	WER
<i>Datu kopa</i>	3.41	3.48	0.74	3.62	3.81	0.84	3.82	4.00	0.60	0.03	0.09
Demucs	3.42	3.43	0.66	3.87	3.98	0.71	4.11	4.23	0.42	0.04	0.10
MP-SENet	3.84	3.94	0.54	4.28	4.48	0.65	4.26	4.37	0.41	0.06	0.15
Resemble	3.95	4.09	0.50	4.64	4.75	0.38	4.50	4.56	0.24	0.04	0.11
Adobe	4.53	4.61	0.37	4.85	4.90	0.29	4.47	4.58	0.35	0.03	0.10

Balss pārveidošanas rīku salīdzinājumam tika izmantots kosinusa attālums starp runātāja balss iegultnes vektoriem, kā arī pārveidotās runas dabiskums un kvalitāte. Rezultātu apkopojums redzams 4. tabulā. Vislabākos rezultātus guva RVC modelis, iegūstot vistuvāko kosinusa distanci pret mērķi, kā arī tā kvalitāte un dabiskuma rādītāji bija visaugstākie gan vīrieša, gan sievietes balsij.

4. tabula. Runātāju balss pārveidošanas rīku salīdzinājums.

Kosinusa attālums					
Modelis	Dzimums	Pret oriģinālu	Pret mērķi	Dabiskums	Kvalitāte
<i>Datu kopa</i>	vīrietis	-	0.80	4.02	4.98
	sieviete	-	0.75	4.49	5.08
FreeVC	vīrietis	0.41	0.67	3.36	4.82
	sieviete	0.45	0.75	4.14	4.88
so-vits-svc	vīrietis	0.38	0.62	3.26	4.80
	sieviete	0.40	0.58	4.11	4.89
RVC	vīrietis	0.35	0.61	3.79	4.94
	sieviete	0.38	0.56	4.39	4.95

¹<https://research.saulitis.dev/latvian-speech-synthesis-2024/>

Modeļu apmācība tika veikta vairākos posmos un 5. tabulā apkoti apmācību precizitātes un kvalitātes rādītāji to labākajā solī. Pirmās un otrās apmācības laikā šie dati vēl netika ievākti, tāpēc nav iespējams veikt objektīvu salīdzinājumu. Šo apmācību sintezētie audio skanēja ievērojami sliktāk - par to pārliecināties var rezultātu mājaslapā.¹ Tālākajos apmācības soļos tika pievienota kvalitātes un precizitātes rādītāju aprēķināšana. Viens izņēmums ir 9. apmācība, kurai tehnisku iemeslu dēļ kvalitātes rādītāju apmācības laikā neizdevās ievākt.

5. tabula. Apmācīto modeļu labākā soļa rādītāji. Ar * atzīmēti vienas balss runas sintēzes modeļi.

Nr.	Solis	CER	WER	Kvalitāte
3	327 400	0.11	0.28	4.99
4	224 600	0.18	0.38	5.00
5	164 600	0.17	0.37	5.00
6	137 000	0.15	0.34	5.01
7	115 800	0.14	0.32	4.97
8*	212 201	0.09	0.23	4.99
9*	117 000	0.15	0.33	-

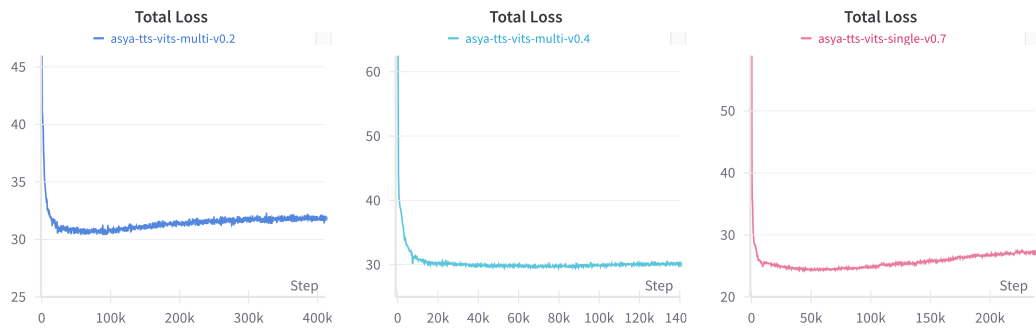
Tālākai modeļu salīdzināšanai ar esošajām sistēmām tika izvēlēti - Modelis Nr. 3 (vairāku balsu), Modelis Nr. 6 (vairāku balsu) un Modelis Nr. 8 (vienas balss). Papildus, vairāku balsu modeļiem tika atlasīta precīzākā balss pēc CER rādītāja un pievienota kopējai salīdzināšanai. Abiem vairāku balsu modeļiem 14. balss sintezēja visprecīzāko runu.

6. tabula. Runas sintēzes rīku salīdzinājums. Ar * atzīmēti vairāku balsu runas sintēzes modeļi.

Modelis	CER	WER	Dabiskums	Kvalitāte
eSpeak	0.20	0.42	1.63	1.67
Hugo.lv	0.04	0.12	2.93	4.41
Clarin	0.08	0.20	2.41	4.24
Coqui	0.07	0.18	3.43	4.21
Modelis Nr. 3 *	0.21	0.44	4.21	4.95
Modelis Nr. 3, 14. balss	0.14	0.33	4.30	4.99
Modelis Nr. 6 *	0.24	0.48	4.34	4.98
Modelis Nr. 6, 14. balss	0.17	0.39	4.26	5.02
Modelis Nr. 8	0.14	0.32	4.35	4.93

¹<https://research.saulitis.dev/latvian-speech-synthesis-2024/>

Modeļu apmācības kopējās kļūdas grafiku salīdzinājums redzams 13. attēlā. Savukārt, apskatot 1., 2. un 3. pielikumā pievienotos apmācības grafikus, var redzēt, ka runas kvalitāte kāpj ļoti strauji un ir iespējams gūt it kā labi skanošus audio, bet ar ļoti zemu precizitāti. Klausoties sintezētos audio var saprast, ka runā latviski, bet nevar saprast kas tiek pateikts.



13. att. Kopējais kļūdas grafiks, no kreisās uz labo pusi - 3. modelis, 6. modelis un 8. modelis

5 Secinājumi

Šī bakalaura darba mērķis bija izstrādāt augstākas kvalitātes un precizitātes latviešu valodas runas sintēzes modeli, izpētot datu kopas priekšapstrādes un apmācības metodes. Lai sasniegtu šo mērķi, tika izvirzīti šādi darba uzdevumi: izpētīt esošo zinātnisko literatūru par angļu valodas runas sintēzes metodēm, izpētīt esošo zinātnisko literatūru par latviešu valodas runas sintēzes metodēm, izpētīt runas sintēzes datu kopu priekšapstrādes metodes, veikt datu kopas atlasīšanu un sagatavošanu latviešu valodas runas sintēzes modeļa izveidei, veikt latviešu valodas runas sintēzes modeļu apmācību un labākā modeļa atlasīšanu, un salīdzināšanu ar esošajiem runas sintēzes rīkiem, izdarīt secinājumus un izvirzīt priekšlikumus un rekomendācijas, pamatojoties uz iegūtajiem rezultātiem. Mērķis tika sasniegts, veicot piecas datu sagatavošanas un priekšapstrādes iterācijas, kā arī deviņus apmācības ciklus. Rezultātā tika veikts runas uzlabošanas rīku un balss toņa pārveidošanas rīku salīdzinājums, kā arī modeļu apmācības un salīdzinājums. No paveiktā darba izriet sekojoši secinājumi:

1. Latviešu valodā ir pieejamas vairākas runas sintēzes sistēmas, kuras ir brīvi pieejamas. Tās demonstrē augstu precizitāti CER un WER rādītājos, bet to skaņas kvalitāte bieži ir salīdzinoši nedabiska un nekvalitatīva. Visaugstākos precizitātes rādītājus uzrādīja Hugo.lv runas sintēzes rīks ar CER 4% precizitāti, taču tam ir arī viens no zemākajiem dabiskuma kvalitātes rādītājiem (2.93), salīdzinot, piemēram, jaunizstrādāto modeli Nr.6., kuram tas ir 4.34.
2. Runas sintēzē datu kopu kvalitātei ir ļoti liela nozīme, un ir nepieciešams veikt vairākus priekšapstrādes soļus, lai iegūtu datus, ar kuriem varētu apmācīt precīzus un kvalitatīvus modeļus. Tika veikta audio failu filtrēšana, izmantojot ASR modeli, audio failu balss ieraksta uzlabošana, balss pārveidošana uz viena runātāja balsi un audio failu ilguma pārrēķināšana.
3. Latviešu valodā ir salīdzinoši maz datu kopu, ko varētu izmantot runas sintēzei, un nav nevienas publiski pieejamas datu kopas, kas būtu radīta tieši runas sintēzes modeļu apmācībai. Atšķirībā no runas atpazīšanas, runas sintēzei ir nepieciešami ievērojami augstākas kvalitātes audio ieraksti. Tāpēc ir nepieciešams veikt datu kopu sagatavošanu un priekšapstrādi, lai iegūtu kvalitatīvus datus apmācībai.
4. Lai gan runas sintēzes modeļiem ir iespējams iegūt augstus dabiskuma un kvalitātes rādītājus NISQA, tas nenozīmē, ka runa būs saprotama, skaidra un precīza. Ļoti svarīgi ir arī apmācības laikā izmantot rādītājus, kas informē gan par kvalitāti, gan par satura precizitāti kā WER un CER.

Jaunizveidotais modelis balss kvalitātē pārspēja visus citus tirgū pieejamos latviešu valodas runas sintēzes modeļus, ieskaitot eSpeak, Hugo.lv, Clarin un Conqui, iegūstot

kvalitātes rādītāju 5.02, vienlaikus, saglājot pieņemamu CER 0.17. Balstoties uz apkopotajiem secinājumiem, var teikt, ka integrētu un rūpīgi izvēlētu priekšapstrādes un uzlabošanas tehnoloģiju izmantošana veicina latviešu valodas runas sintēzes modeļu kvalitāti un precizitāti. No izdarītajiem secinājumiem darba sākumā, izvirzītās hipotēzes tika daļēji vai pilnībā apstiprinātas:

1. Sintezētās balss kvalitātes un precizitātes noteikšanai apmācības laikā nepietiek ar testa kļūdas skalāro vērtību, lai noteiktu runas sintēzes modeļa veikspēju. Tika apstiprināts, ka ir nepieciešami papildu rādītāji, ar kuriem noteikt apmācītā modeļa statusu, kā NISQA, WER un CER. Ņemot vērā tikai vienu vai nevienu no šiem papildus rādītājiem, rezultāts ir labs pēc kļūdas funkcijas skalārās vērtības, bet iegūtā sintezētā balss nav saprotama un izmantojama.
2. Runas uzlabošanas modeļu izmantojums datu priekšapstrādē uzlabo kvalitātes un precizitātes rādītājus. Šī hipotēze tika apstiprināta, jo modeļu precizitāte un kvalitāte uzlabojās pēc balss kvalitātes uzlabošanas modeļu izmantošanas.
3. Balss toņa pārveide uz viena runātāja balss toni uzlabo kvalitātes un precizitātes rādītājus. Šī hipotēze tika daļēji apstiprināta, jo precizitāte uzlabojās, bet kvalitātes rādītāji ne vienmēr bija augstāki. Būtu nepieciešami vēl papildus pētījumi, lai apstiprinātu šo hipotēzi pilnībā.

Pētījumā laikā tika arī uzrakstīta zinātniskā publikācija "Towards Natural-Sounding Speech To Text in English", kura tika pieņemta konferencē "5th International Conference on Deep Learning Theory and Applications" (DeLTA 2024). Raksts tiks indeksēts SCOPUS datubāzē, un to prezenēt ir plānots 10. - 11. jūlijā 2024. gadā.

Noslēgumā var secināt, ka latviešu valodas runas sintēzes modeļu izstrāde ir iespējama, bet prasa rūpīgu datu kopu sagatavošanu un priekšapstrādi, kā arī rūpīgu apmācību un modeļa izvēli. Lai iegūtu augstas kvalitātes un precizitātes runas sintēzes modeļus, ir nepieciešams izmantot vairākas priekšapstrādes un uzlabošanas tehnoloģijas, kā arī rūpīgi izvēlēties apmācības metodes un rādītājus.

6 Tālākie pētījumi

Ņemot vērā pētījuma rezultātus un secinājumus, būtu nepieciešams turpināt kvalitatīvas datu kopas izstrādi runas sintēzes apmācības nolūkam, izmantojot publiski pieejamās datu kopas. Viens no variantiem būtu apvienot vairākas mazākas datu kopas. Neskatoties uz to, noteikti nepieciešams veikt vēl detalizētāku datu kopu analīzi un rūpīgāku to attīrīšanu.

Balstoties uz straujo tehnoloģiju attīstīšanos, būtu vērts izskatīt vēl kādus jaunākus runas sintēzes modeļus, izskatot arī tos modeļus, kuriem ir publiski pieejams to pirmkods, bet ne svāri. Viens no daudzsološākajiem ir nākamās paaudzes VITS modelis, VITS 2 [18], kas veic vērā ņemamus uzlabojumus esošajam VITS modelim.

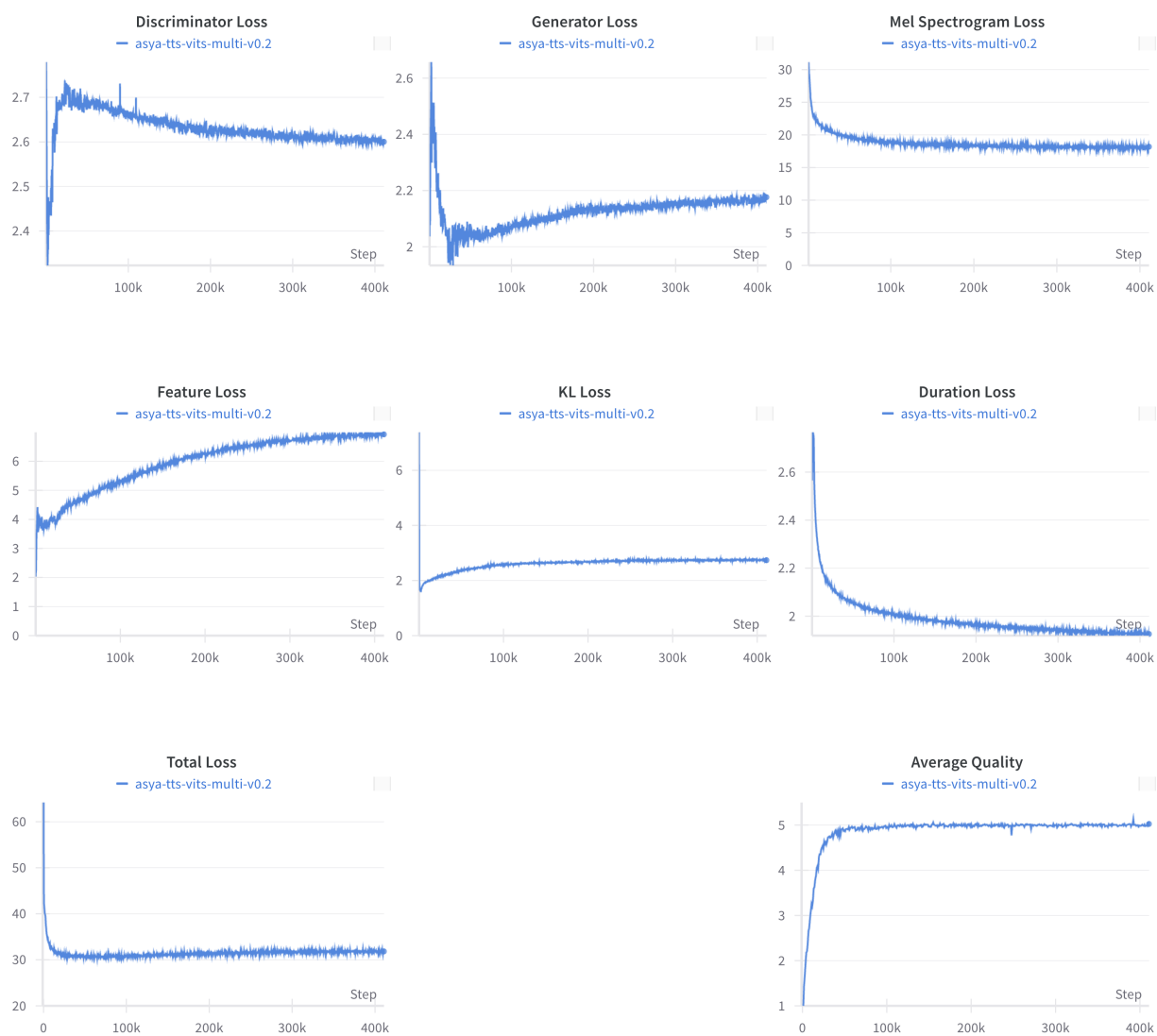
Bibliogrāfija

- [1] URL: <https://espeak.sourceforge.net/>.
- [2] URL: <https://hugo.lv/lv/About>.
- [3] I. Auziņa u. c. *Latvian Speech Corpus (LaRKO)*. Accessed: 2024-05-26. 2014. URL: <http://hdl.handle.net/20.500.12574/22>.
- [4] Nicolas D'Alessandro u. c. "MaxMBROLA: A Max/MSP MBROLA-based tool for real-time voice synthesis". *2005 13th European Signal Processing Conference*. IEEE. 2005, 1.—4. lpp.
- [5] Roberts Dargis u. c. "Development and evaluation of speech synthesis corpora for Latvian". *Proceedings of the Twelfth Language Resources and Evaluation Conference*. 2020, 6633.—6637. lpp.
- [6] Roberts Dargis un Ilze Auziņa. *Ilvars - Latvian Male VITS Text-to-Speech Model (vers. 2023)*. CLARIN-LV digital library at IMCS, University of Latvia. 2023. URL: <http://hdl.handle.net/20.500.12574/89>.
- [7] Alexandre Défossez, Gabriel Synnaeve un Yossi Adi. "Real Time Speech Enhancement in the Waveform Domain". *ArXiv abs/2006.12847* (2020). URL: <https://api.semanticscholar.org/CorpusID:219981437>.
- [8] Jacob Devlin u. c. "Bert: Pre-training of deep bidirectional transformers for language understanding". *arXiv preprint arXiv:1810.04805* (2018).
- [9] John Duchi, Elad Hazan un Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of machine learning research* 12.7 (2011).
- [10] Conor Durkan u. c. "Neural spline flows". *Advances in neural information processing systems* 32 (2019).
- [11] Ian Goodfellow, Yoshua Bengio un Aaron Courville. *Deep learning*. MIT press, 2016.
- [12] Wei-Ning Hsu u. c. "Hubert: Self-supervised speech representation learning by masked prediction of hidden units". *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), 3451.—3460. lpp.
- [13] Ibrahim Kazeem. *Midjourney v5: Create stunning AI images with the latest version*. 2023. g. nov. URL: <https://www.dxtalks.com/blog/news-2/midjourney-v5-a-guide-to-the-new-ai-image-generation-model-390>.
- [14] Jaehyeon Kim u. c. "Glow-tts: A generative flow for text-to-speech via monotonic alignment search". *Advances in Neural Information Processing Systems* 33 (2020), 8067.—8077. lpp.

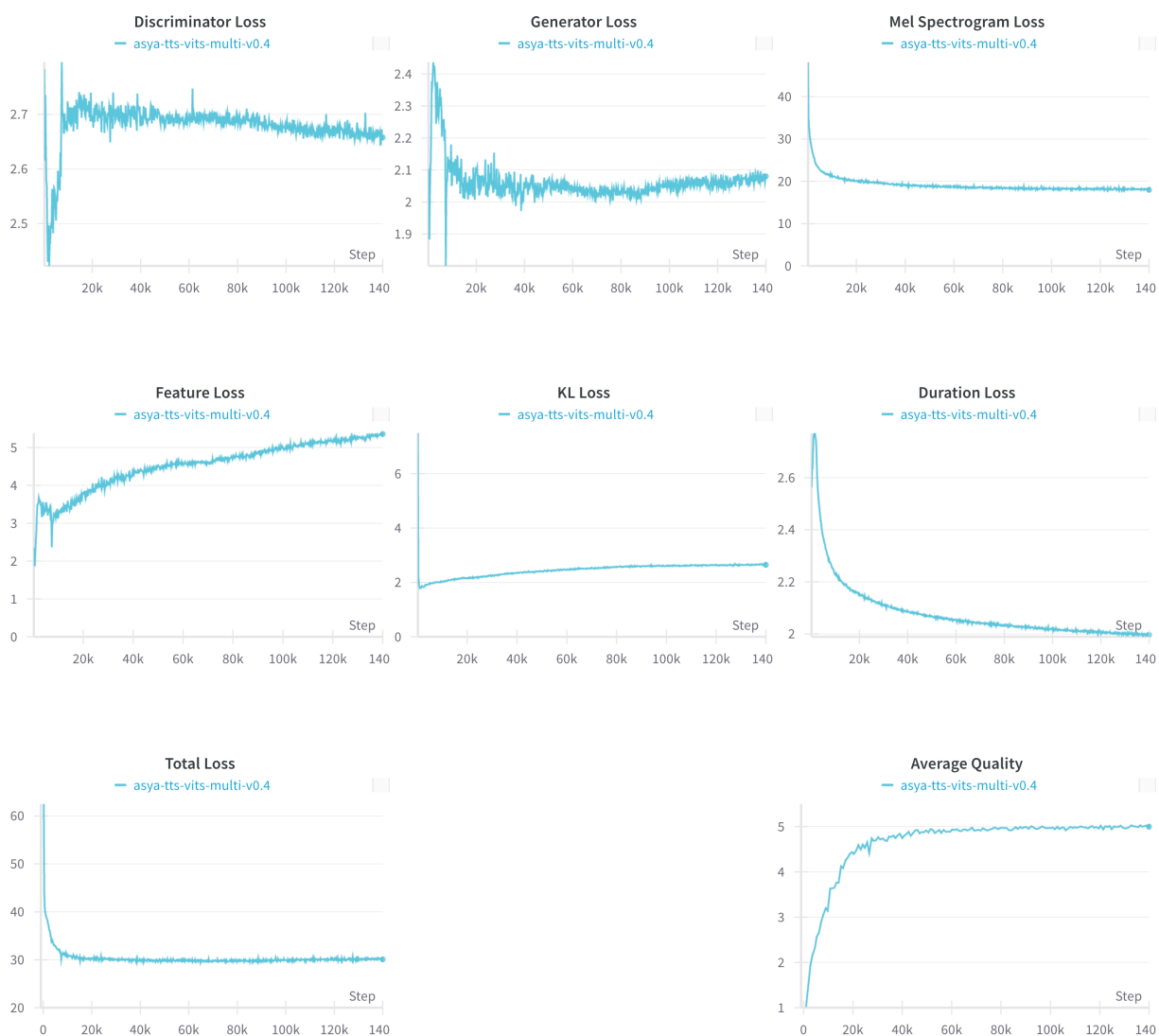
- [15] Jong Wook Kim u. c. “Crepe: A convolutional representation for pitch estimation”. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, 161.—165. lpp.
- [16] Diederik P Kingma un Jimmy Ba. “Adam: A method for stochastic optimization”. *arXiv preprint arXiv:1412.6980* (2014).
- [17] Jungil Kong, Jaehyeon Kim un Jaekyoung Bae. “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis”. *Advances in neural information processing systems* 33 (2020), 17022.—17033. lpp.
- [18] Jungil Kong u. c. “VITS2: Improving Quality and Efficiency of Single-Stage Text-to-Speech with Adversarial Learning and Architecture Design”. *arXiv preprint arXiv:2307.16430* (2023).
- [19] *Latvian-female-TTS-model-vits-encoding-trained-on-cv-dataset-at-22050hz*. URL: <https://aimodels.org/ai-models/text-to-speech-synthesis/latvian-female-tts-model-vits-encoding-trained-on-cv-dataset-at-22050hz/>.
- [20] Jingyi Li, Weiping Tu un Li Xiao. “Freevc: Towards high-quality text-free one-shot voice conversion”. *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2023, 1.—5. lpp.
- [21] Ye-Xin Lu, Yang Ai un Zhenhua Ling. “MP-SENet: A Speech Enhancement Model with Parallel Denoising of Magnitude and Phase Spectra”. *INTERSPEECH 2023* (2023). URL: <https://api.semanticscholar.org/CorpusID:258841685>.
- [22] Xudong Mao u. c. “Least squares generative adversarial networks”. *Proceedings of the IEEE international conference on computer vision*. 2017, 2794.—2802. lpp.
- [23] Gabriel Mittag u. c. “NISQA: A deep CNN-self-attention model for multidimensional speech quality prediction with crowdsourced datasets”. *arXiv preprint arXiv:2104.09494* (2021).
- [24] Aaron van den Oord u. c. “Wavenet: A generative model for raw audio”. *arXiv preprint arXiv:1609.03499* (2016).
- [25] Alexis Plaquet un Hervé Bredin. “Powerset multi-class cross entropy loss for neural speaker diarization”. *Proc. INTERSPEECH 2023*. 2023.
- [26] Vadim Popov u. c. “Grad-tts: A diffusion probabilistic model for text-to-speech”. *International Conference on Machine Learning*. PMLR. 2021, 8599.—8608. lpp.
- [27] Ryan Prenger, Rafael Valle un Bryan Catanzaro. “Waveglow: A flow-based generative network for speech synthesis”. *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, 3617.—3621. lpp.

- [28] Alec Radford u. c. “Improving language understanding by generative pre-training”. (2018).
- [29] Alec Radford u. c. “Robust speech recognition via large-scale weak supervision”. *International Conference on Machine Learning*. PMLR. 2023, 28492.—28518. lpp.
- [30] Kriss Saulitis. “Angļu valodas runas sintēzes modeļu salīdzinājums”. (2024).
- [31] Kriss Saulitis, Evalds Urtans un Vairis Caune. “Towards Natural-Sounding Speech to Text in English”. *Communications in Computer and Information Science*. Pieņemts, bet nepublicēts. Springer Nature, 2024.
- [32] Marc Schröder un Jürgen Trouvain. “The German text-to-speech synthesis system MARY: A tool for research, development and teaching”. *International Journal of Speech Technology* 6 (2003), 365.—377. lpp.
- [33] Claude Elwood Shannon. “Communication in the presence of noise”. *Proceedings of the IRE* 37.1 (1949), 10.—21. lpp.
- [34] Kai Shen u. c. “Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers”. *arXiv preprint arXiv:2304.09116* (2023).
- [35] Xiangxin Tan u. c. “A Survey on Neural Speech Synthesis”. *arXiv preprint arXiv:2106.15561* (2021).
- [36] Xu Tan. *Neural text-to-speech synthesis*. Springer Nature, 2023.
- [37] T. Tieleman. *Lecture 6.5-rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude*. 2012. URL: <https://cir.nii.ac.jp/crid/1370017282431050757>.
- [38] Ashish Vaswani u. c. “Attention is all you need”. *Advances in neural information processing systems* 30 (2017).
- [39] Lilian Weng. “Flow-based deep generative models”. (2018). URL: <https://lilianweng.github.io/posts/2018-10-13-flow-models>.

1. pielikums - 3. modeļa apmācības kļūdu un rādītāju grafiki



2. pielikums - 6. modeļa apmācības kļūdu un rādītāju grafiki



3. pielikums - 8. modeļa apmācības kļūdu un rādītāju grafiki

