

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

“Stable Diffusion modeļa adaptācija latviešu valodā”

BAKALaura DARBS

Autors:

Mihails Tumaševičs,

stud. apl. nr. mt19031

Darba vadītājs:

Dr.dat. Kārlis Freivalds

RĪGA 2023

ANOTĀCIJA

Bakalaura darba mērķis ir izpētīt attēlu ģenerējošā modeļa Stable Diffusion darbības principus un adaptēt to lietošanai latviešu valodā.

Stable Diffusion ir dziļās mašīnmācīšanās attēlu ģenerēšanas modelis, kas spēj ģenerēt vizuāli pievilcīgus attēlus no padotās teksta ievades. Stable Diffusion atbalsta tikai angļu valodu, kā arī neeksistē Stable Diffusion balstīts modelis, kas atbalsta latviešu valodu. Tādēļ šī darba ietvaros tika izveidots Stable Diffusion balstīts modelis, kas spēj ģenerēt attēlus no latviešu valodā padotā teksta, kā arī spēj attēlot Latvijai specifiskus kultūras konceptus un objektus.

Atslēgas vārdi: Stable Diffusion, mašīnmācīšanās, dziļā mašīnmācīšanās, attēlu ģenerēšana

ABSTRACT

The aim of this final paper is to conduct research on Stable Diffusion image generation model as well as adapting it to Latvian.

Stable Diffusion is a deep learning image generation model that can generate visually appealing images from the given text prompt. Currently, Stable Diffusion supports only English as well as there is no Stable Diffusion based model that supports Latvian. Therefore, in the scope of this paper a new model based on Stable Diffusion that can generate images from Latvian text as well as can visualize Latvian cultural concepts and objects was developed.

Keywords: Stable Diffusion, machine learning, deep learning, image generation

SATURS

IEVADS	6
APZĪMĒJUMU SARAKSTS	7
1. STABLE DIFFUSION DARBĪBAS PRINCIPS.....	9
1.1 Stable Diffusion attēla ģenerēšana no teksta virknes.....	9
1.2. Difūzijas modelis	10
1.3. Teksta virknes kodētājs.....	11
2. TEKSTA VIRKNES TULKOŠANA	12
2.1. Tulkošanas priekšrocības	12
2.2. NLLB-200 modelis	12
2.3. Tulkošanas modeļa pielietošana	13
3. DREAMBOOTH DARBĪBAS PRINCIPS.....	14
3.1. Stable Diffusion modeļu pielāgošana izmantojot Dreambooth	14
3.2. Eksistējošā modeļa izmantošana trenēšanā.....	14
3.3. Koncepta vai objekta apraksts Dreambooth trenēšanā	15
3.4. Dreambooth pielietojums citos modeļos.....	15
4.1. Stable Diffusion attēlu ģenerēšana latviešu konceptiem un objektiem	17
4.2. Precīzās iestatīšanas priekšrocības.....	18
4.3. Izvēlētie koncepti modeļa apmācīšanai	18
5. DATU KOPAS IZVEIDE.....	19
5.1. Datu savākšana.....	19
5.2. BLIP moduļa pielietojums Stable Diffusion apmācīšanā	20
5.3. References klases attēlu ģenerēšana	21
6. MODEĻA PIELĀGOŠANAS PROCEDŪRAS VIDES SAGATAVOŠANA	22
6.1. Mākoņa skaitļošanas izmantošana modeļa trenēšanas procesam	22
6.2. RunPod servera konfigurēšana	23

6.3. Stable Diffusion WebUI konfigurēšana.....	24
6.4. Dreambooth instalēšana Stable Diffusion WebUI vidē	25
7. MODEĻA PIELĀGOŠANA	27
7.1. Pielāgošanas parametru konfigurēšana	27
7.2. Jaunā Dreambooth modeļa izveide	27
7.3. Dreambooth pielāgošanas parametru konfigurēšana	28
7.4. Pielāgotā modeļa optimāla kontrolpunkta noteikšana	29
8. MODEĻA PUBLICĒŠANA HUGGING FACE MĀJASLAPĀ	32
8.1. Modeļa kontrolpunkta pārveidošana par modeli.....	32
8.2. Jaunā modeļa repozitorija izveidošana Hugging Face mājaslapā.....	33
8.3. Modeļa ielāde.....	34
9. MODEĻA TESTĒŠANA.....	35
9.1. Modeļa konceptu testēšana	35
9.2. Vājākie ģenerēšanas modeļa aspekti.....	38
9.3. Stiprākais ģenerēšanas modeļa aspekts.....	39
REZULTĀTI.....	40
SECINĀJUMI.....	41
IZMANTOTĀ LITERATŪRA.....	42
PIELIKUMI.....	44

IEVADS

Pēdējos gados sakarā ar datortehnikas attīstību, datoru jaudas palielināšanos, plašāku datu apjoma pieejamību un jauno algoritmu parādīšanos ir strauji attīstījusies mašīnmācīšanās nozare. Tika izstrādāti vairāki dziļās mašīnmācīšanās modeļi, kurus cilvēki izmanto ikdienā, iespējams, par to nezinot. Mašīnmācīšanās modeļi tiek izmantoti lai paveiktu visdažādākās funkcijas kā, piemēram, galda spēļu spēlēšana, runas atpazīšana vai teksta ģenerēšana. Tie tiek integrēti pārlūkprogrammās, operētājsistēmās un mājaslapās veicot to lietošanas pieredzes uzlabošanu. Viens no tādiem modeļiem ir Stable Diffusion, kurš kļuva brīvi pieejams 2022. gadā. Stable Diffusion spēj ģenerēt attēlus no padotās teksta virknes, kas apraksta attēlu, kuram jābūt uzģenerētam.

Stable Diffusion modeļa galvenais mērķis ģenerēt vizuāli pievilcīgus, reālistiskus un kvalitatīvus attēlus. Šis modelis ir noderīgs māksliniekiem, lai ātri ģenerētu attēlus, kurus izmantot kā iedvesmu jauniem mākslas darbiem vai ideju vizualizācijai, šis modelis noder arī datu zinātniekiem ar jaunu datu ģenerēšanu lielos apjomos, kā arī Stable Diffusion ir pielietojams vispārējiem lietotājiem, ģenerējot personalizētus attēlus personīgo projektu izstrādāšanai.

Stable Diffusion modelis ir pielāgots angļu valodas pārziņiem un spēj ģenerēt attēlus tikai no teksta, kurš ir uzrakstīts angļu valodā. Kopš šī modeļa izlaišanas, tika izveidoti vairāki jaunie dziļās mašīnmācīšanās modeļi, kas tika bāzēti uz Stable Diffusion un kuri ir spējīgi ģenerēt attēlus izmantojot teksta virknes uzrakstītas citās valodās. Tomēr uz šo brīdi nepastāv neviens latviešu valodai pielāgots Stable Diffusion modelis.

Šī bakalaura darba mērķis ir izstrādāt Stable Diffusion attēlu ģenerējošo modeli, kas spēs ģenerēt attēlus no latviešu valodā padotā teksta, kā arī spēs attēlot Latvijai specifiskus kultūras konceptus un objektus.

APZĪMĒJUMU SARAKSTS

Stable Diffusion – dziļās mašīnmācīšanās attēlu ģenerēšanas modelis

Mašīnmācīšanās – datorzinātne, kura fokusējas uz datu un algoritmu izmantošanu lai imitētu cilvēka mācīšanas procesu.

Neironu tīkls - skaitļošanas sistēma, kas sastāv no liela daudzuma skaitļošanas elementu, saukti par neironiem, kas noteiktā veidā savienoti viens ar otru, un kopā veic kādu noteiktu uzdevumu.

Dziļā mašīnmācīšanās – mašīnmācīšanās apakškopa, kurā neironu tīkliem ir vairāki slāņi un kuras apmācības process mēģina simulēt cilvēka smadzenes, lai apgūtu lielu datu apjomu.

Attēlu ģenerēšana – jauno attēlu izveide no eksistējošām datu kopām.

Apslēptie dati (Latent data) – dati, kas nav skatāmi un bieži vien galalietotājs tos vairs neizmanto, izmantojot operētājsistēmu vai standarta datorprogrammu.

Difūzijas modelis (Diffusion model) – ģenerēšanas modelis, kura galvenais mērķis iemācīties atjaunot attēla sākotnējā stāvokli pēc Gausa trokšņa pielietošanas.

Apslēptais difūzijas modelis (Latent diffusion model) – difūzijas modelis, kas Gausa troksni apstrādā apslēpto datu veidā.

Gausa troksnis - statistiskais troksnis ar Gausa sadalījumu.

Embedings (Embedding) – 768 vērtību vektors, kas apraksta teksta marķieri.

ResNet slānis - pārpalikuma konvolūciju neironu tīkli (Residual Convolutional Neural Network)

Google Colab - Google Research produkts, kas tiek izmantots, lai rakstītu un palaistu Python un citu programmēšanas valodu programmas pārlūkprogrammā.

chrF++ vērtējums – mašīnmācīšanās tulkojumu vērtējums, kurš tiek noteikts salīdzinot tulkošanas modeļa tulkojumu ar pārbaudes tulkojumiem

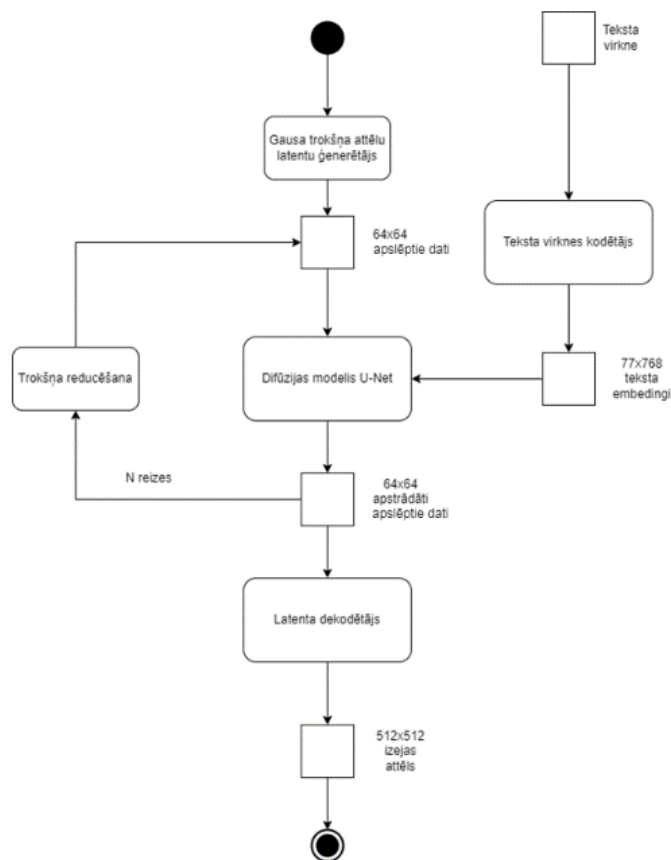
Precīzā iestatīšana (Fine tuning) – pieeja mašīnmācīšanās modeļa pielāgošanai, kurā iepriekš apmācīta modeļa svāri tiek apmācīti, izmantojot jaunus datus.

Tekstuālā inversija (Textual Inversion) – precīzās iestatīšanas metode, kas nodrošina difūzijas modeļa apmācību jaunajiem konceptiem izmantojot nelielu attēlu skaitu (3-5 attēli).

1. STABLE DIFFUSION DARBĪBAS PRINCIPS

1.1 Stable Diffusion attēla ģenerēšana no teksta virknes

Attēla ģenerēšanas procesa sākuma Stable Diffusion uzģenerē Gausa trokšņa attēlu un pārveido to apslēptos datos. Apslēptie dati ir dati, kas nav skatāmi, un bieži vien galalietotājs tos vairs neizmanto, izmantojot operētājsistēmu vai standarta datorprogrammu. Paralēli tam procesam ievades teksta virkne tiek padota teksta virknes kodētājam. Pēc teksta virknes apstrādes un pārveidošanas teksta embeddingos (embedding) - 768 vērtību vektors, kas apraksta teksta marķieri - Gausa trokšņa attēla apslēptie dati un teksta embeddingi tiek padoti U-Net difūzijas modelim. U-Net iterējot attēla apslēptiem datiem N reizes pakāpeniski noņem attēla Gausa troksni, tādā veidā uzģenerējot jauno attēla datus. Ģenerēts attēls ir atkarīgs no padotiem teksta embeddingiem. Tad apstrādātie apslēptie dati tiek padoti apslēpto datu dekodētājam, kas pārveido tos par attēla datiem.



1.1.att. *Stable Diffusion attēla ģenerēšanas aktivitāšu diagramma*

1.2. Difūzijas modelis

Stable Diffusion modelis pieder pie difūzijas modeļu klases. Difūzijas modeļi spēj ģenerēt jaunus datus, kas Stable Diffusion gadījumā ir attēli, izmantojot datus, kuri tika izmantoti modeļa trenēšanā.

Difūzijas modeļa galvenie posmi ir:

- Turpvērsta difūzija (Forward diffusion) – process, kurā tiek pakāpeniski pievienots Gausa troksnis datiem, kamēr tiem pazūd visas noteicošās pazīmes. Šis process notiek tikai modeļa apmācības laikā.
- Reversā difūzija (Reverse diffusion) – process, kurā iteratīvi tiek noņemts Gausa troksnis no datiem. Šis process pārvērš datus ar nejauši ģenerētu troksni par reālistiskiem datiem.



1.2.att. Difūzijas process

Difūzijas modeļa trenēšana ir ļoti lēns un resursietilpīgs process, tāpēc Stable Diffusion izmanto apslēpto datu difūzijas modeli (latent diffusion model) [1]. Attēls tiek saspriests apslēpto datu telpā (latent space), kas padara attēla izmēru 64 reizes mazāku [2]. Priekš saspiešanas Stable Diffusion izmanto Variational Autoencoder (VAE). Tas padara modeļa apmācības procesu daudz ātrāku.

Stable Diffusion neironu tīkls izmanto U-Net arhitektūru [3], kas trenēšanas paātrināšanai, attēla datus apstrādā caur vairākiem ResNet slāņiem samazinot to divreiz. Samazināts attēls tiek padots apstrādei difūzijas modelim. Pēc attēla apstrādes, tas, izejot tam pašam ResNet slāņu skaitam, tiek atgriezts pie sava sākotnēja izmēra.

1.3. Teksta virknes kodētājs

Teksta virknes kodētājs atbild par ievades teksta virknes pārveidi par embeddingu, ko kopā ar trokšņa attēlu uzņems difūzijas modelis, lai uzģenerētu teksta virknē aprakstītu attēlu.

Pirms teksta virknes pārveidošanas embeddingā, teksts tiek pārveidots par marķieri izmantojot CLIP marķētāju [4]. CLIP ir dziļās mašīnmācīšanās modelis, kas ģenerē attēlu tekstuālo aprakstu. Stable Diffusion modelis ir limitēts 75 marķieru izmantošanai.

Pēc marķiera izveides, tas tiek pārveidots par embeddingu. Stable Diffusion šim nolūkam tiek izmantots ViT-L/14 CLIP modelis [5]. Stable Diffusion modelis ir trenēts uz datiem, kas tika padoti caur CLIP modeli, tāpēc gadījumā, ja modelis tiks aizvietots, ir nepieciešams pārtrenēt pamata U-Net modeli izmantojot datus no jaunā modeļa. Katram marķierim ir savs 768 vērtību vektors, kas to apraksta, kurš tiek nodefinēts modeļa trenēšanas laikā. Embedingi nodrošina nozīmju saisti starp vārdiem, kuri tika pārveidoti par marķieriem. Tas ļauj Stable Diffusion modelim ģenerēt attēlus izmantojot noteiktus stilus, kurus ir iespējams padot modelim teksta virknē.

Embedingi kopā ar trokšņa attēlu tiek padoti U-Net difūzijas modelim, kas no trokšņa, sekojot embeddingos aprakstītai informācijai, uzģenerē attēlu.

2. TEKSTA VIRKNES TULKOŠANA

2.1. Tulkošanas priekšrocības

Tā kā Stable Diffusion U-Net modelis ir cieši saistīts ar teksta kodētāja modeli, uz kura tas tika trenēts, lai pielāgotu attēlu ģenerēšanai izmantojot citu teksta kodētāju, kas būs spējīgs interpretēt latviešu valodu, ir nepieciešams pārtrenēt U-Net modeli izmantojot to latviešu kodētāju. Šīm procesam ir nepieciešams liels trenēšanas datu apjoms, ka arī milzīgs skaitļošanas resursu apjoms lai sasniegtu attēlu kvalitāti un precizitāti līdzīgu oriģinālajam modelim. Savukārt, iztulkojot teksta virkni pirms attēla ģenerēšanas ir iespējams sasniegt oriģinālā modeļa kvalitāti patērējot mazāk resursu.

2.2. NLLB-200 modelis

Teksta virknes tulkošanai pirms padošanas attēlu ģenerējošajam modelim tika izvēlēts NLLB-200 tulkošanas modelis. NLLB-200 modelis ir 2022. gada uzņēmuma Meta publicēts dziļās mašīnmācīšanās tulkošanas modulis, kurš spēj tulkot tekstu starp 200 dažādām valodām. Šis modelis ir daļa no projekta “No Language Left Behind”, kura pašreizējais galvenais mērķis ir izveidot rīku, kurš spēs tulkot tekstu mazākumtautību valodām ar pieejamo iztulkoto teikumu skaitu zemāku par vienu miljonu.

NLLB-200 tulkošanas modelis ir brīvi pieejams piecās konfigurācijās[6]:

Modeļa nosaukums	Parametru skaits (mljrd.)	chrF++ vērtējums tulkojumam no latviešu valodas uz angļu
nllb-moe-54b	54.5	60.8
nllb-200-3.3b	3.3	60.4
nllb-200-1.3b	1.3	58.8
nllb-200-distilled-1.3b	1.3	59.4
nllb-200-distilled-600m	0.6	56.3

Tabula 2.2.1 NLLB-200 konfigurācijas

NLLB-200 arī iekļauj sevī datu kopas, kuras var izmantot jauno tulkošanas modeļu izveidošanai. Piemēram viena no valodām, kas ir pieejamā NLLB trenēšanas datu kopā ir latgaļu valoda, kas nākotnē varētu nodrošināt Stable Diffusion moduļa adaptēšanu arī tai.

2.3. Tulkošanas modeļa pielietošana

Šī darba ietvaros tika izmantotas divas NLLB-200 tulkošanas modeļa konfigurācijas. Google Colab vidē “nllb-200-distilled-600m” konfigurācija, jo tā patērē mazāk operatīvas atmiņas, un to ir iespējams izmantot lietotāji ar standarta Google Colab lietotāja ierobežojumiem. Lokālajā vidē tika izmantota “nllb-200-distilled-1.3b” konfigurācija, jo tā sniedz labākus tulkojumus nekā “nllb-200-distilled-600m” un pieprasa daudz mazāk resursu nekā lielāki modeļi, un ģenerē tulkojumus ar līdzīgu precizitāti.

NLLB-200 modeļa izmantošanai ir nepieciešams Hugging Face mājaslapas lietotāja autentifikācijas marķieris. Tāpēc pirms modeļa ielādes ir nepieciešams saglabāt pieejas informāciju programmas palaišanas vidē.

Modeļa un tā kodētāja ielāde notiek izmantojot “Transformers” satvaru. Pirms tulkošanas teksta virkne tiek pārveidota par marķieriem izmantojot NLLB-200 kodētāju, tad marķieri tiek padoti modelim tulkošanai. Tulkošanas ieejas un izvades valodas tiek definētas izmantojot BCP-47 kodu[7]. Kad tulkošana ir pabeigta, marķieri tiek pārveidoti teksta virknē.

Ievades virkne	NLLB-200 “nllb-200-distilled-1.3b” konfigurācijas tulkojums
Latvijas Universitātes ēkas bilde	Picture of the building of the University of Latvia
Rakstnieks raksta grāmatu	A writer writes a book
Students autobusā	A student on a bus
Astronauts brauc ar zirgu	Astronaut riding on horseback
Bakalaura darba jāizstrādā un jānoformē atbilstoši metodiskajiem materiāliem	The bachelor’s thesis must be developed and formatted in accordance with the methodological materials

Tabula 2.3.1 NLLB-200 tulkojumi

3. DREAMBOOTH DARBĪBAS PRINCIPS

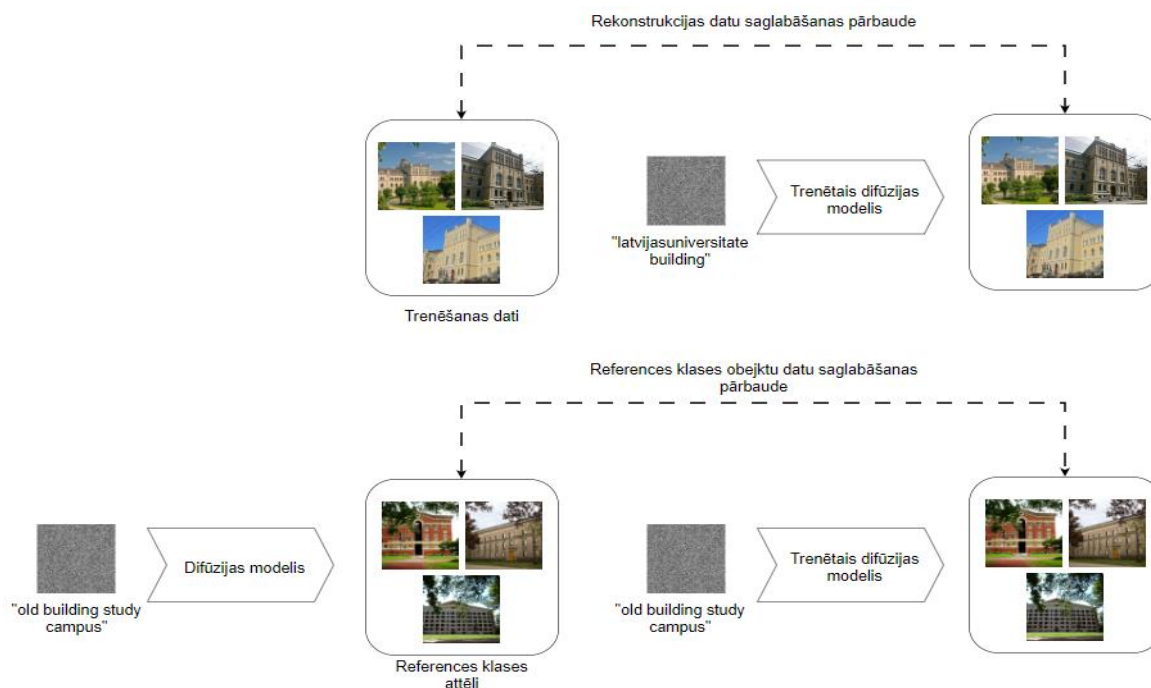
3.1. Stable Diffusion modeļu pielāgošana izmantojot Dreambooth

Šī darba ietvaros Stable Diffusion pielāgošana tika paveikta izmantojot Dreambooth modeļa trenēšanas paņēmieni. Izmantojot Dreambooth ir iespējams pievienot jaunās zināšanas par objektiem izmantojot minimālo ieejas datu apjomu.

Dreambooth pamata darbošanās princips ir balstīts uz to, ka modelim, uz kura pielieto Dreambooth trenēšanu, jau satur sevī zināšanas par vispārējiem objektu konceptiem. Ievadot jaunā objekta vai koncepta datus modelī, izmantojot unikālo marķieri kopā ar references klasi, modelis asociē jauno konceptu ar pazīmēm, kas jau eksistē references konceptos. Šis nodrošina modeļa apmācīšanas efektivitāti jaunajiem konceptiem, jo modelis iemācās tikai tās pazīmes, kuras atšķir konceptu no references klases.

3.2. Eksistējošā modeļa izmantošana trenēšanā

Trenēšanas laikā kā sākuma punkts tiek izmantots jau eksistējošs Stable Diffusion modeļa U-Net slānis. Modelis tiek trenēts izmantojot ievades attēlus kopā ar teksta aprakstu, kas satur unikālu identifikatoru un nosaukumu klasei, kurai pieder attēlots objekts vai koncepts (piem., “[A] ēka”). Paralēli tam tiek nodrošināta saglabāšana pamata moduļa references klases attēlu raksturīgiem datiem. Tas piesaista priekšrocību koncepta vai objekta saistītai klasei, kas nodrošina, ka objekts varēs tikt ģenerēts no vairākiem leņķiem.



3.3. Koncepta vai objekta apraksts Dreambooth trenēšanā

Trenēšanā izmantojot Dreambooth ir ļoti svarīgi izmantot unikālos vai reti sastopamus marķierus, lai identificētu konceptu, kuram pamata attēlu ģenerēšanas modelis tiek apmācīts. Galvenā problēma izmantojot jau esošos angļu valodas vārdus vai citas valodas jau eksistējošos vārdus ir tāda, ka modelim ir jāiemācās tos atdalīt no oriģinālās nozīmes un no jauna jāsavieno tos ar references konceptiem un konceptu, kas tiek padots.

Salīdzinot Dreambooth ar populāro modeļu pielāgošanas metodi, kuru izmanto noteikto konceptu ģenerēšanai, Tekstuālo Inversiju (Textual Inversion), Dreambooth trenēti modeļi spēj ģenerēt modelim uztrenētos koncepta attēlus ar augstāku kvalitāti un precizitāti.



Attēls 3.3.1 Dreambooth trenēta modeļa salīdzinājums ar tekstuālās inversijas trenēto modeli

3.4. Dreambooth pielietojums citos modeļos

Dreambooth modeļu pielāgošanas paņēmiens tiek plaši izmantots citos Stable Diffusion bāzētajos modeļos. Vairākoties Stable Diffusion pielāgo, lai tas varētu ģenerēt ļoti specifiskus objektus vai konceptus kā mājdzīvniekus un zīmēšanas stilus. Piemēram, uzņēmums “Lambda Labs” izveidoja Stable Diffusion modeli, kas spēj ģenerēt

“Avatar” filmas stila attēlus. Tomēr Dreambooth izmantošana Stable Diffusion modeļa pielāgošanai specifiskām kultūrām nav izplatīta.



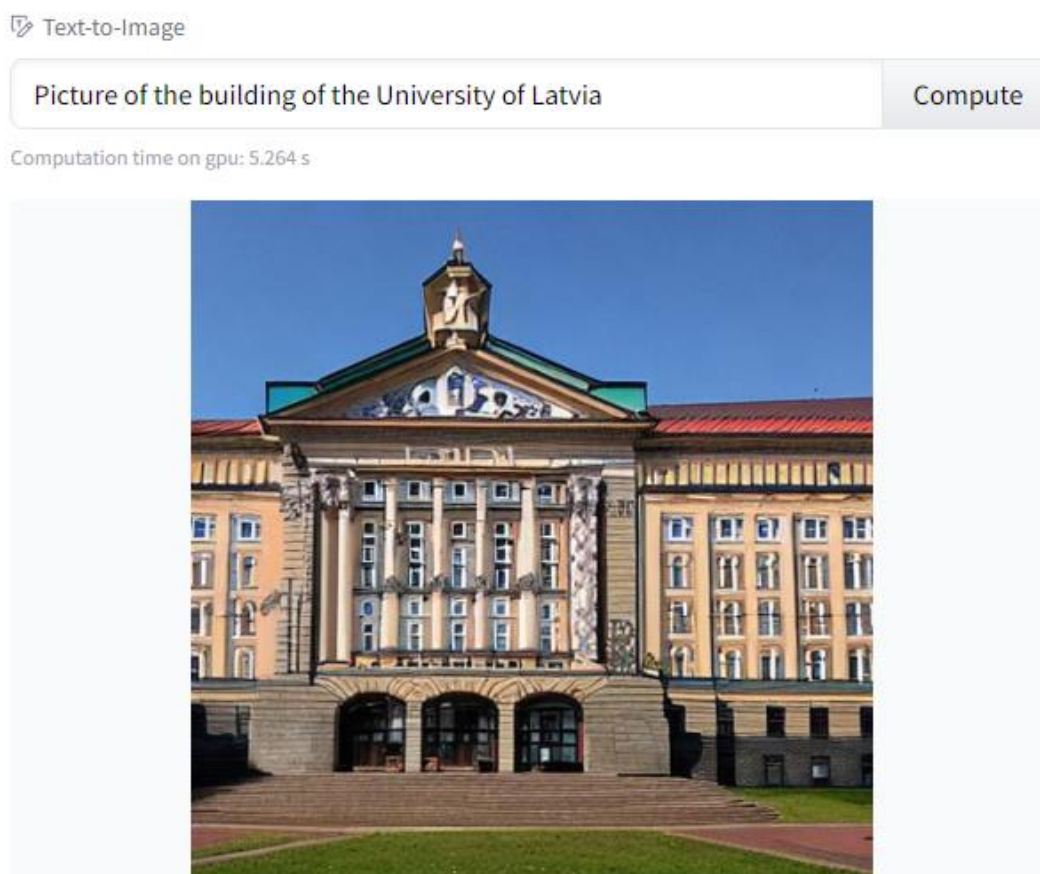
Attēls 3.4.1 Dreambooth trenēta modeļa “dreambooth-avatar” filmas stilam ģenerētie attēli

4. STABLE DIFFUSION PRECĪZĀ IESTATĪŠANA

4.1. Stable Diffusion attēlu ģenerēšana latviešu konceptiem un objektiem

Stable Diffusion modelis tika trenēts izmantojot attēlus ar pieejamiem angļu valodas attēlu aprakstiem. Šī iemesla dēļ attēlu ģenerēšanas modelis nespēj uzģenerēt attēlus, kas ir cieši saistīti ar noteikto kultūru konceptiem, kas nav plaši pieejami angļu valodā.

Lai pielāgotu Stable Diffusion modeli jauno konceptu ģenerēšanai, ir iespējams vai nu modeli pārtrenēt izmantojot datus, kas ietvers latviešu kultūras konceptus un objektus, vai izmantot modeļa precīzo iestatīšanu (Fine Tuning) – pieeja mašīnmācīšanās modeļa pielāgošanai, kurā iepriekš apmācīta modeļa svāri tiek apmācīti, izmantojot jaunus datus. Šī darba ietvaros modeļa pielāgošanai tika izvēlēta modeļa precīzā iestatīšana.



Attēls 4.1.1 Stable Diffusion 1.5 ģenerētais attēls virknei “Picture of the building of the University of Latvia”

4.2. Precīzās iestatīšanas priekšrocības

Precīzā iestatīšana tika izmantota Stable Diffusion modeļa pielāgošanai latviešu valodā, jo tai ir sekojošas priekšrocības:

- Precīzā iestatīšana patērē drastiski mazāk laika un skaitļošanas resursu, salīdzinot ar jaunā modeļa trenēšanu. Jo precīzā iestatīšana ir balstīta uz jau eksistējošo modeļu apmācības, tā ļauj izmantot zināšanas no pamata modeļa, tāpēc modeli ir iespējams trenēt tikai nodaļās, kuras ir nepieciešamas lietotājam. Šīs nozīmē arī to, ka datu apjoms, kas ir nepieciešams modeļa uztrenēšanai, ir mazāks.
- Stable Diffusion modeļa zināšanas ir ļoti plašas un daudzveidīgas. Tas veicina vieglāku un kvalitatīvāku jauno konceptu pievienošanu, jo jaunie koncepti, visdrīzāk, būs saistīti ar konceptiem, kuri jau pastāv modelī.
- Precīza iestatīšana ļauj pievienot jauno informāciju jau uztrenētam modelim vairākās iterācijās. Gadījumā, ja modeli ir nepieciešams atjaunot ar aktuāliem datiem vai pievienot papildus konceptus, ir iespējams modeli, kurš jau tika trenēts ar specializētiem datiem, papildināt ar jaunām zināšanām.

4.3. Izvēlētie koncepti modeļa apmācīšanai

Stable Diffusion modeļa pielāgošanai tika izvēlēti 10 latviešu kultūras koncepti un objekti, kurus pirms trenēšanas modelis nespēja uzgenerēt:

- Melngalvju Nams
- Latvijas Nacionālā bibliotēka
- Vanšu tilts
- Brīvības piemineklis
- Latvijas Universitātes galvenā ēka
- Piparkūkas
- Rīgas Melnais Balzams
- Latviešu rakstu zīmes
- Vispārējie latviešu Dziesmu un Deju svētki
- Rainis

5. DATU KOPAS IZVEIDE

5.1. Datu savākšana

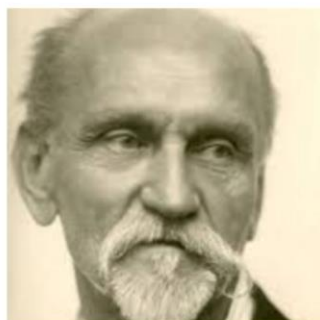
Dati ir viena no svarīgākajām daļām mašīnmācīšanās modeļu trenēšanā. Lai pielāgotu Stable Diffusion modeli izvēlētajiem attēliem tika savāktas 11 bildes katram konceptam izņemot Latvijas Universitātes galveno ēku – tai tika savāktas 22 bildes (11 gājēja skatupunkta bildes un 11 bildes no putna lidojuma).

Lai iegūtu kvalitatīvu modeli ir jā sagatavo kvalitatīvi ieejas dati. Tāpēc atlasot bildes datu kopas izveidei tika izmantotas augstākās pieejamās kvalitātes bildes. Tika nodrošināts arī tas, ka izmantotās bildes nesatur vizuālos defektus. Izmantojot zemas izšķirtspējas bildes, izstaiptas bildes vai bildes ar kustības izplūšanu tiks uztvertas kā detaļas no trenēšanas koncepta, un izvadē modelis ģenerēs bildes ar defektiem. Trenējot Stable Diffusion modeli noteikto objektu ģenerēšanai ir arī ieteicams izmantot vairākas bildes, kas attēlo objektu no vairākiem leņķiem, lai modelis spētu ģenerēt daudzveidīgākus attēlus. Netika arī izmantoti stilizēti attēli vai mākslas darbi, kas atspoguļo objektu, jo tas arī negatīvi ietekmēs ģenerēto attēlu kvalitāti.

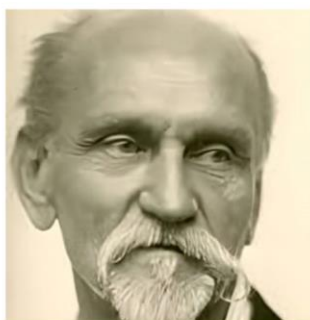
Kad bildes tika atlasītas, tās tika padotas attēlu izšķirtspējas palielināšanas modelim[9], kas turpmāk uzlaboja attēlu kvalitāti.

Kā pēdējais solis ieejas attēlu apstrādei, tās tika apgrieztas novietojot galveno objektu bildes centrā un attēlu izmērs tikai nomainīts uz 512x512px. Atlikušie attēlu defekti tika manuāli noņemti izmantojot Adobe Photoshop attēlu rediģēšanas rīku.

Attēls pirms apstrādes



Attēls pēc apstrādes

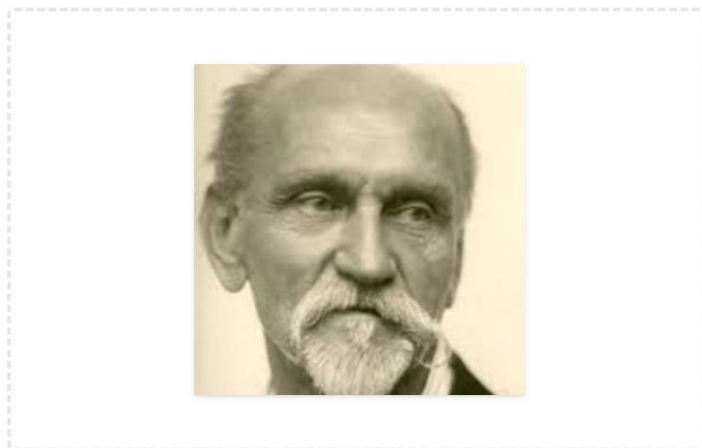


Attēls 5.1.1 Raiņa portrets pirms un pēc apstrādes

5.2. BLIP moduļa pielietojums Stable Diffusion apmācīšanā

Pēc attēlu atlasīšanas un apstrādes, katram attēlam tika ģenerēts tekstuālais apraksts izmantojot BLIP attēlu aprakstu ģenerēšanas modeli[10]. Attēlu tekstuālie apraksti Stable Diffusion modeļa pielāgošanai ir nepieciešami, lai uzģenerētu references klases, kas nodrošinās labāku modeļa trenēšanas kvalitāti.

Image-to-Text



Computation time on Intel Xeon 3rd Gen Scalable cpu: 1.800 s

an old black and white photo of a man with a beard

Attēls 5.2.1 Raiņa portreta BLIP modeļa uzģenerētais apraksts

5.3. References klases attēlu ģenerēšana

Iegūtie attēlu apraksti tika manuāli pārbaudīti katrai bildei, taču apraksti tika apstrādāti nepieciešamības gadījumos, ja aprakstā ir liekas detaļas vai dažas detaļas iztrūkst no attēla. Pēc attēlu aprakstu apstrādes tie tika padoti Stable Diffusion modelim, lai uzģenerētu references klases attēlus. Dreambooth trenēšanas laikā references klases attēli ir nepieciešami, lai modelī esošie koncepti ir saistīti ar apmācīto konceptu vai objektu. Tas nodrošinās ātrāku apmācīšanos un izejas modeļa kvalitāti. Katram no atlasītiem attēliem tika uzģenerēti 10 references attēli. Kopā modeļa trenēšanai tika izmantoti 1210 references klases attēli.



Attēls 5.2.1 References klases attēli virknei “An old black and white photo of a man with a white beard”

6. MODEĻA PIELĀGOŠANAS PROCEDŪRAS VIDES SAGATAVOŠANA

6.1. Mākoņa skaitļošanas izmantošana modeļa trenēšanas procesam

Stable Diffusion modeļa trenēšana ir skaitļošanas resursu ietilpīgs process. Neskatoties uz to, ka modeļa adaptēšanai tika izvēlēta precīzas iestatīšanas trenēšanas metode, tā pieprasa jaudīgu grafisko karti, kas lielākajai cilvēku daļai nav pieejama. Darba autoram datorā tiek izmantotā NVIDIA GeForce RTX 3070 grafiskā karte ar 8GB video atmiņu. Tā ir pielietojama modeļa palaišanai lokālajā vidē, bet Dreambooth trenēšanai tai pietrūkst video atmiņas, jo minimālais nepieciešamais atmiņas daudzums ir 16GB - tāpēc trenēšanai tika izmantota mākoņa vide.

Tīmekļa vidē ir pieejami vairāki servisi, kas piedāvā mākoņa skaitļošanas pakalpojumus, un gandrīz visi no tiem piedāvā izīrēt serverus ar jaudīgām grafiskajām kartēm mašīnmācīšanās modeļu trenēšanai. Neskatoties uz to, ka Dreambooth ir iespējams izmantot ar 16GB video kartes atmiņas daudzumu, tas pieprasa piepūles no atmiņas pārvaldības puses un pastāv iespēja, ka trenējot modeli video atmiņas patēriņš pārsniegs 16GB, saglabājot moduļa kontrolpunktu. Šī iemesla dēļ stabilai modeļa trenēšanai ir ieteicams izmantot grafisko karti ar 24GB video atmiņas.

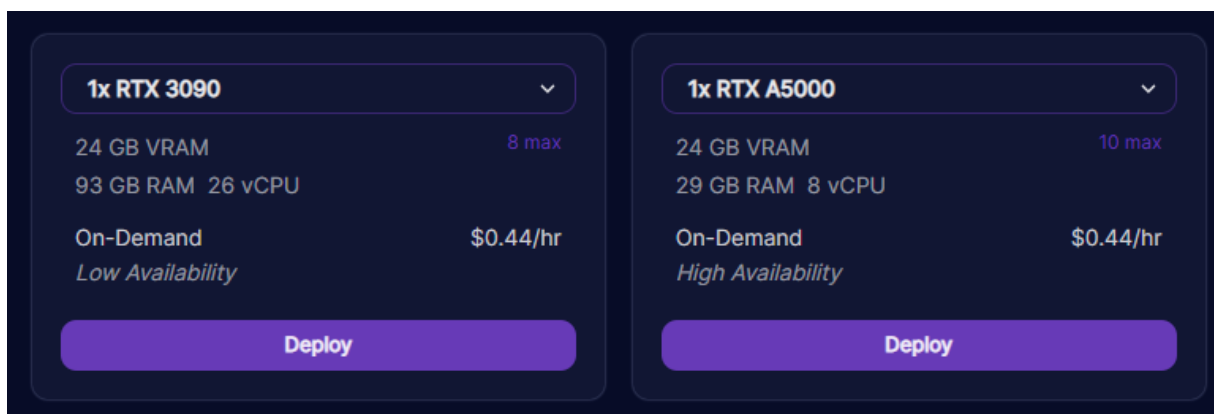
Šī darba ietvaros tika izlemts izmantot RunPod mākoņa skaitļošanas pakalpojuma sniedzēja servisu. RunPod ir viens no vislētākajiem servisiem, kas piedāvā Eiropā novietotus serverus, kuri spēj veikt Dreambooth trenēšanu.

Mākoņa skaitļošanas pakalpojuma sniedzējs	Servera ar 24GB video atmiņas grafiskās kartes cena stundā (EUR/h)
Google Cloud	0.57
Amazon AWS	0.69
RunDiffusion	2.31
Vast.AI	0.44
RunPod	0.41

Tabula 6.1.1 Mākoņa skaitļošanas servisu cenas

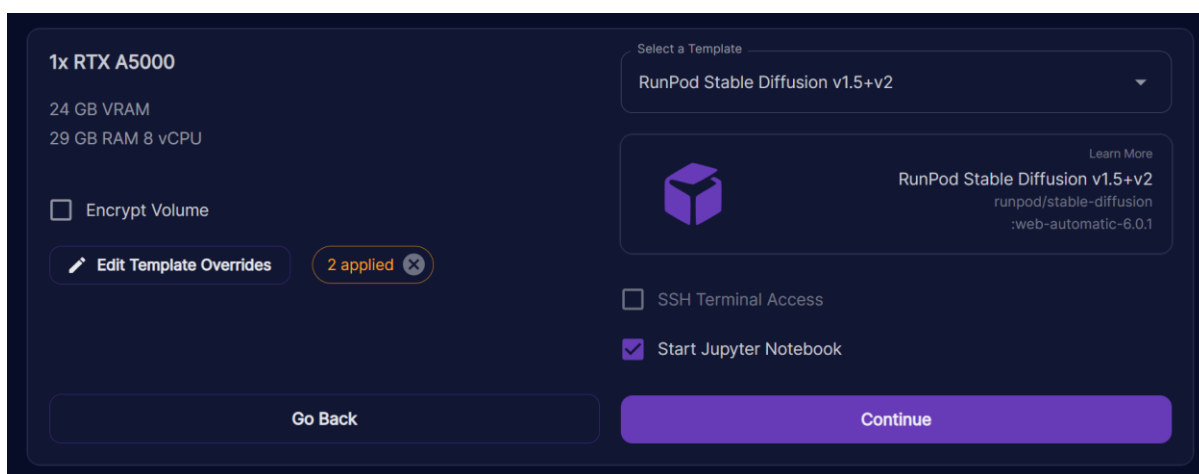
6.2. RunPod servera konfigurēšana

Atverot RunPod mājaslapu un izveidojot lietotāju ir iespējams izvēlēties skaitļošanas serveri no saraksta ar vairākām serveru specifikācijām. Dreambooth trenēšanas nolūkiem jāizvēlas jebkurš serveris ar 24GB video atmiņu. Ir ieteicams izmantot serveri ar augstu pieejamību, tāpēc tika izvēlēts serveris ar RTX A5000 grafisko karti.



Attēls 6.2.1 RunPod pieejamo serveru saraksts ar 24GB grafiskās atmiņas

Pēc servera izvēles ir iespējams konfigurēt servera noklusējuma konfigurāciju. Pie servera šablona konfigurācijas jāizvēlas “RunPod Stable Diffusion v1.5+v2”, kas nodrošinās automātisko Stable Diffusion kopā ar Stable Diffusion WebUI ielādi pēc servera pirmreizējās palaišanas. Pirms servera palaišanas ir ieteicams arī nomainīt konfigurācijas atvēlētās atmiņas daudzumu konteinerim un sējumam. Šī darba nolūkiem konteineru izmērs tika nomainīts uz 10GB un sējuma izmērs uz 100GB. Pēc servera konfigurēšanas ir iespējams palaist serveri nospiežot pogu “Turpināt”.



Attēls 6.2.2 RunPod servera konfigurācija

6.3. Stable Diffusion WebUI konfigurēšana

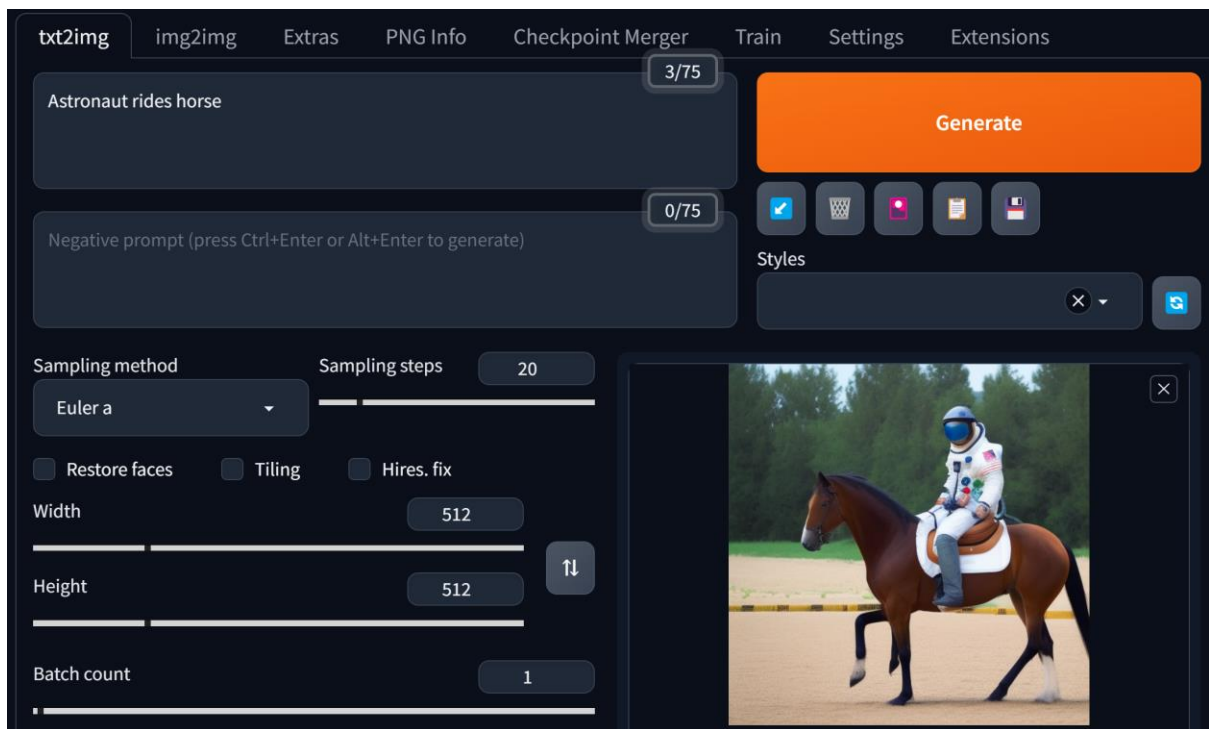
RunPod serverim ir iespējams pieslēgties izmantojot SSH savienojumu vai Jupyter Lab WEB interfeisu, kurš automātiski tiek palaists kopā ar serveri. Pieslēdzoties serverim izmantojot komandrindu ir jāatjaunina Stable Diffusion WebUI repozitorija versija izmantojot “git pull” komandu.

Pēc repozitorija atjaunošanas ir ieteicams rediģēt relaunch.py skriptu, lai būtu iespējams manuāli pārtraukt vai palaist WebUI procesu izmantojot komandrindu. Skriptā mūžīgu ciklu “while true” ir jāaizvieto ar “while (n<1)”. Ir arī nepieciešams instalēt pytorch un xformers satvarus izmantojot “pip3 install torch==1.13.1 torchvision torchaudio --index-url https://download.pytorch.org/whl/cu117” un “pip install https://huggingface.co/MonsterMMORPG/SECourses/resolve/main/xformers-0.0.19-cp310-cp310-manylinux2014_x86_64.whl” komandas.

Tad ir jāielādē Stable Diffusion modelis, kas tiks izmantots Stable Diffusion pielāgošanai latviešu konceptiem un objekti. Šī kursa darba ietvaros tika izmantots Runwayml Stable Diffusion 1.5 modeļa atzarošanas variants “v1-5-pruned”. Komandrindā ir jāatver Stable Diffusion WebUI “models” mapes apakšmapi “Stable diffusion”. Palaizot “wget https://huggingface.co/runwayml/stable-diffusion-v1-5/resolve/main/v1-5-pruned.ckpt” [13] komandu tiks ielādēts Stable Diffusion modelis. Papildus var arī ielādēt VAE modeļa atzarošanas variantu izmantojot “wget https://huggingface.co/stabilityai/sd-vae-ft-mse-original/resolve/main/vae-ft-mse-840000-ema-pruned.ckpt” komandu.

Kad visas palaistās komandas ir paveiktas ir nepieciešams restartēt serveri. Kad serveris ir veiksmīgi pārstartēts, ir ieteicams atvērt Jupyter Lab vidi un palaist WebUI lietotni izmantojot komandrindu, pirms tam apstādinot lietotnes instanci, kas tika automātiski palaista pie servera startēšanas. Tas ļaus sekot lietotnes atklūdošanas izvadei izmantojot komandrindu.

Pēc Stable Diffusion WebUI lietotnes veiksmīgās pārstartēšanas, lietošanas interfeisam ir iespējams pieslēgties izmantojot RunPod opciju “Pieslēgties caur HTTP [Ports 3000]”. Pieslēdzoties lietotnes interfeisam ir jāatver lietotnes iestatījumi, un sadaļā “Stable Diffusion” ir nepieciešams nomainīt SD VAE opciju uz iepriekš ielādēto VAE modeli. Ja instalācija un konfigurācija tika veikta korekti, sadaļā “txt2img”, ievadot teksta virkni un nospiežot “Ģenerēt” pogu, Stable Diffusion modelim ir jāuzģenerē attēls un tam ir jāattēlojas WebUI interfeisā.



Attēls 6.3.1 Stable Diffusion uzģenerētais testa attēls Stable Diffusion WebUI vidē

6.4. Dreambooth instalēšana Stable Diffusion WebUI vidē

Šī darba ietvaros tika izmantots “d8ahazard/sd_dreambooth_extension” paplašinājums[12], kas ietver sevī papildus Dreambooth optimizēšanu, kas palielina trenēšanas ātrdarbību, kā arī pielieto mazāku video atmiņas apjomu. Lai ieinstalētu šo paplašinājumu Stable Diffusion WebUI vidē, caur RunPod mājaslapu jāpalaiž Jupyter Lab vide un jāatver komandrinda “stable-diffusion-webui” mapes apakšmapē “extensions”. Palaižot “git clone https://github.com/d8ahazard/sd_dreambooth_extension” komandu komandrindā tiks ielādēts Dreambooth paplašinājums.

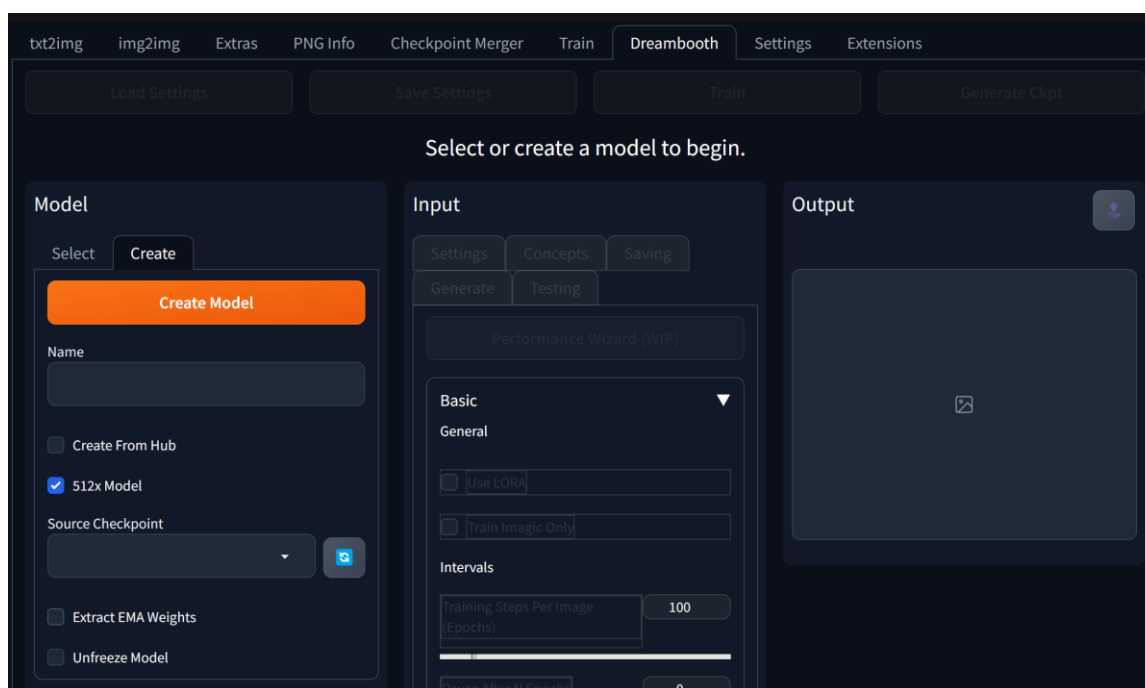
Kad paplašinājums tiks ielādēts visdrīzāk tas nedarbosies, jo tā saistītie satvaru nepieciešamās versijas atšķiras no Stable Diffusion WebUI nepieciešamajām satvaru versijām, kas izraisīs atkarību konfliktu kļūdu. Lai to salabotu teksta rediģēšanas lietotnē, ir jāatver paplašinājuma mapes teksta failu “requirements.txt” un jāpieraksta satveru versijas, kuri ir nepieciešami Dreambooth paplašinājuma lietošanai. Tad ir jāatver Stable Diffusion WebUI pamata mapes failu “requirements_versions.txt” un jāatjaunina satveru versijas, tā lai tie atbilstu Dreambooth paplašinājuma satvaru versijām. Šī darba izstrādes laika

“requirements_versions.txt” failā vajadzēja atjaunot “fastapi”, “gitpython” un “transformers” satvaru versijas.

Kad Stable Diffusion WebUI nepieciešamo satvaru versiju saraksts ir atjaunots, ir nepieciešams ieinstalēt Dreambooth paplašinājuma nepieciešamos satvarus. To ir iespējams paveikt Dreambooth paplašinājuma mapē caur komandrindu palaižot “pip install -r requirements.txt” komandu. Pēc paplašinājumu ieinstalēšanas ir nepieciešams restartēt RunPod serveri.

Gadījumā, ja Stable Diffusion WebUI lietotnes palaišanas laikā pēc pievienotām izmaiņām joprojām tiek izmesta kļūda par satveru trūkšanu, ir jāveic RunPod servera konteineru datu atiestatīšanu. Pēc tā jāieinstalē visi nepieciešamie satvari no jauna.

Ja Dreambooth paplašinājums tika veiksmīgi ieinstalēts, Stable Diffusion WebUI interfeisā ir jāparādās jaunajai sadaļai ar nosaukumu Dreambooth.

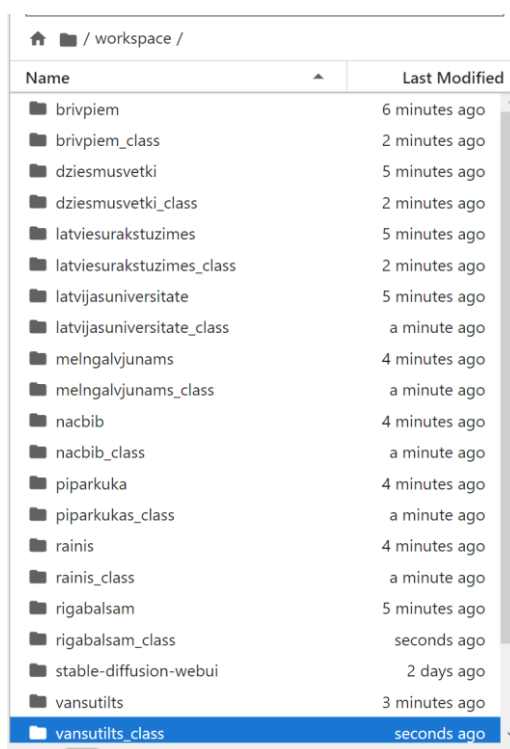


Attēls 6.4.1 Dreambooth sadaļa Stable Diffusion interfeisā

7. MODEĻA PIELĀGOŠANA

7.1. Pielāgošanas parametru konfigurēšana

Pirms Stable Diffusion modeļa pielāgošanas ir nepieciešams ielādēt modeļa trenēšanas datus RunPod serverī. Katram konceptam tiek izveidota mape ar koncepta marķiera nosaukumu, kā arī katram konceptam tiek izveidota mape priekš references klases datu ielādes.

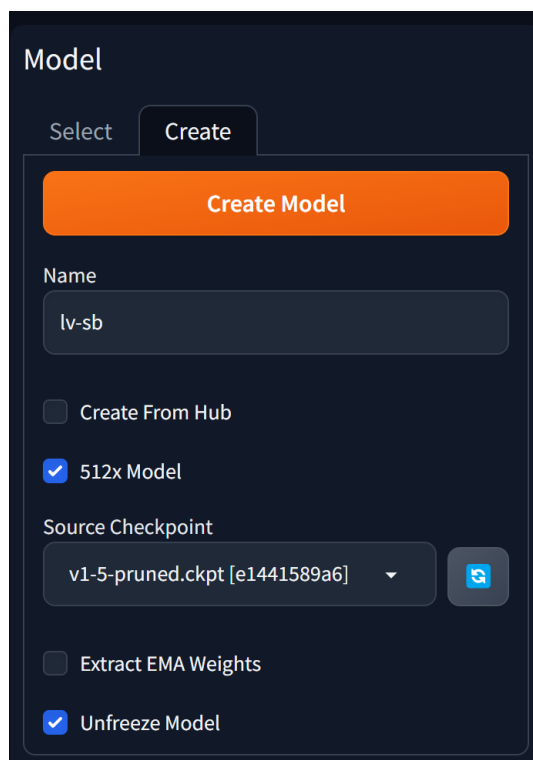


Name	Last Modified
brivpiem	6 minutes ago
brivpiem_class	2 minutes ago
dziesmusvetki	5 minutes ago
dziesmusvetki_class	2 minutes ago
latviesurakstuzimes	5 minutes ago
latviesurakstuzimes_class	2 minutes ago
latvijasuniversitate	5 minutes ago
latvijasuniversitate_class	a minute ago
meIngalvjunams	4 minutes ago
meIngalvjunams_class	a minute ago
nacbib	4 minutes ago
nacbib_class	a minute ago
piparkuka	4 minutes ago
piparkukas_class	a minute ago
rainis	4 minutes ago
rainis_class	a minute ago
rigabalsam	5 minutes ago
rigabalsam_class	seconds ago
stable-diffusion-webui	2 days ago
vansutirts	3 minutes ago
vansutirts_class	seconds ago

Attēls 7.1.1 Izveidotās mapes modeļa trenēšanas datu glabāšanai

7.2. Jaunā Dreambooth modeļa izveide

Pirms sākt modeļa trenēšanu ir jāizveido jaunais modelis, kas pārkopēs parametrus no Stable Diffusion pamata modeļa. Jāatzīmē, ka izveidotais trenēšanai modelis izmantos 512x512px izmēra attēlus, ka arī ir jāatzīmē, ka modelis ir “jāatsaldē”. Modeļa atsaldēšana nozīmē, ka trenētā modeļa jauno konceptu zināšanas tiks saglabātas modeli. Tas nodrošinās to, gadījumā, ja trenēšana tiks pārtraukta pirms modelis būs saglabāts kontrolpunktā, progress joprojām tiks paturēts izveidotajā modelī.



Attēls 7.2.1 Jaunā Dreambooth modeļa izveide

7.3. Dreambooth pielāgošanas parametru konfigurēšana

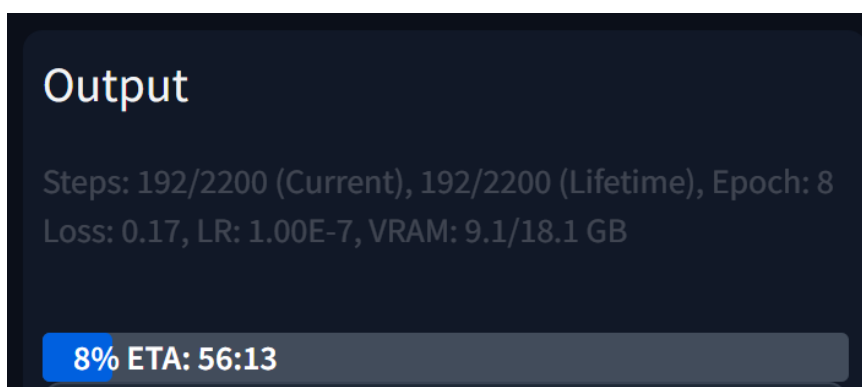
Dreambooth sadaļā “Ievade” tiek nodefinēti trenēšanas parametri. Nospiežot pogu “Veiktspējas vednis”, trenēšanas parametri tiks pielāgoti servera konfigurācijai. Stable diffusion pielāgošanai tika izmantoti sekojošie parametri:

- Trenēšanas soļu skaits katram attēlam (Diskretizācijas periodi): 200
- Modeļa kontrolpunkta saglabāšanas periods (Diskretizācijas periodi): 10
- Mācīšanas ātrums: 0.0000001
- Optimizētājs: Lion
- Jauktā precizitāte: bf16
- Atmiņas uzmanība: noklusēta
- Svaru samazināšana: 0.1

Sadaļā “Koncepti” tiek nodefinēti trenēšanas objekta dati, marķieris un koncepta apraksts. Ir iespējams nospiegt pogu “Trenēšanas vednis”, kas automātiski iestātīs parametrus koncepta trenēšanai. Lai uztrenētu modeli, katram modeļa pievienotajam konceptam tika izvēlēti sekojošie parametri.

- Datu kopas direktorijs: servera mapes direktorijs, kurā tika ievietoti attēli modeļa trenēšanai (piem. “/workspace/nacbib”)
- References klases datu kopas direktorijs: servera mapes direktorijs, kurā tika ievietoti references klases attēli (piem. “/workspace/nacbib_class”)
- Instances uzvedne: koncepta marķieris (piem. “nacbib”)
- References klases uzvedne: references klases apraksta teksts (piem. “a large modern building with a lot of windows”)
- Nolases uzvedne: virkne ar kuru tiek gaidīts ka būs iespējams uzģenerēt jauno konceptu (piem. “nacbib building”)
- References klases attēlu skaits uz katru pamata attēlu: 0

Sadaļa “Saglabāšana” ir jāatzīmē opcija “Ģenerēt .ckpt failu apmācības laikā”. Kad visi parametri ir ievadīti, nospiežot pogu “Trenēt” uzsāksies modeļa trenēšana.



Attēls 7.3.1 Dreambooth trenēšanas progressa vizualizēšana

7.4. Pielāgotā modeļa optimāla kontrolpunkta noteikšana

Pēc modeļa trenēšanas pabeigšanas tiek uzģenerēti vairāki uztrenētā modeļa kontrolpunkti. Pārbaudot katru no tiem, izvēlās kontrolpunktu, kurā attēli tiek ģenerēti viskvalitatīvāk un visprecīzāk. Uz labākā modeļa pamata tiek turpināta trenēšana citiem konceptiem.

Piemērā ar Latvijas Nacionālās bibliotēkas modeļa izveides vislabākās kvalitātes kontrolpunkts tika sasniegts pēc 1980 trenēšanas soļiem, tāpēc šis modelis tiks pielietots citu konceptu trenēšanas pamatmodulis.



Attēls 7.4.1 Latvijas Nacionālās bibliotēkas modeļa kontrolpunktu salīdzinājums 220 – 1100 soļiem



Attēls 7.4.2 Latvijas Nacionālās bibliotēkas modeļa kontrolpunktu salīdzinājums 1320 – 2200 soļiem

Ir iespējams turpināt trenēšanu un mēģināt uzlabot izvēlēto konceptu turpmāk, bet pastāv arī iespēja, ka modelis tiks pārtrenēts un sāks ģenerēt sliktākas kvalitātes attēlus vai sāks aizmirst iepriekš iemācītos konceptus. Pārtrenējot modeli, ģenerēšanas kvalitāte var pasliktināties ne tikai galvenā koncepta attēlu ģenerēšanā, bet arī vispārējā modeļa konceptiem, it īpaši konceptiem, kuri ir saistīti ar trenēto konceptu. Šī iemesla dēļ ir ieteicams saglabāt vairākus kontrolpunktus visam modelim, lai pēc trenēšanas varētu salīdzināt ģenerēšanu ar iepriekšējiem kontrolpunktiem un izvēlēties vislabāko no vairākiem kandidātiem.

Noteiktie kontrolpunkti ir labāki vienā konceptā, bet citā tie tiek sabojāti, tāpēc ir svarīgi pārbaudīt visus trenētos konceptus pie katra kontrolpunkta, lai noteiktu, kurš būs viskvalitatīvākais.



Attēls 7.4.3 Ģenerētais attēls no pārtrenētā modeļa



"a photo of
rigabalsam"
pirms
pārtrenēšanas



"a photo of
rigabalsam"
pēc
pārtrenēšanas

Attēls 7.4.4 Koncepta aizmiršana pārtrenētam modelim

8. MODEĻA PUBLICĒŠANA HUGGING FACE MĀJASLAPĀ

8.1. Modeļa kontrolpunkta pārveidošana par modeli

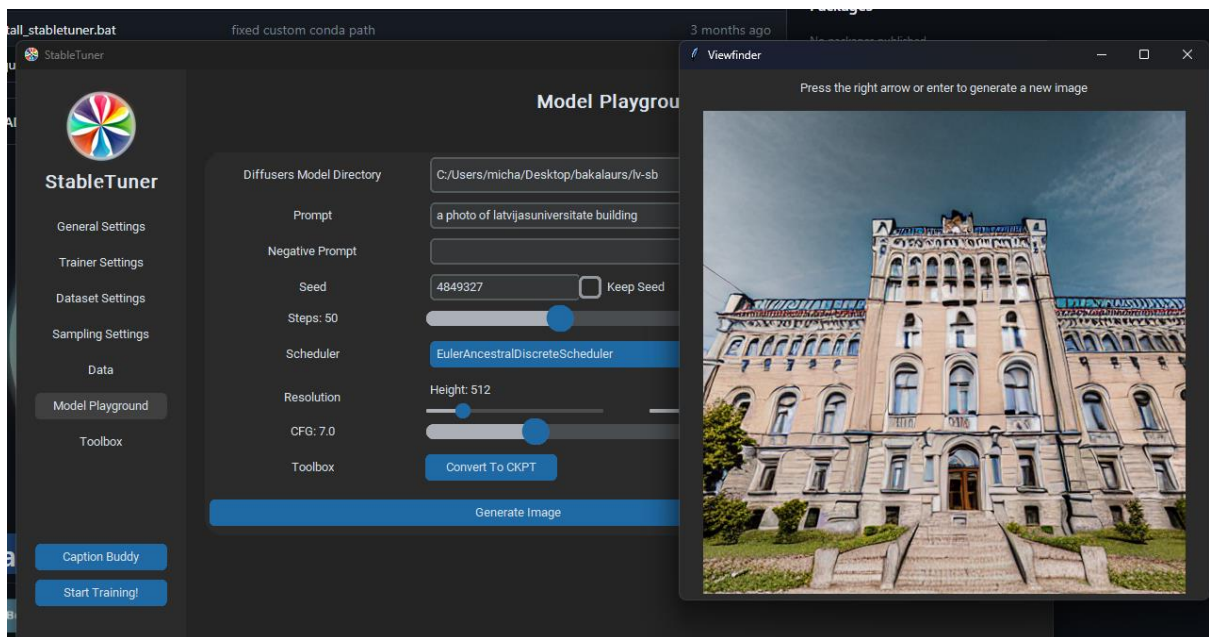
Uztrenētie modeļa kontrolpunkti tika saglabāti servera “/workspace/stable-diffusion-webui/models/Stable-diffusion/” direktorijā. Tos ir iespējams ielādēt izmantojot Jupyter Lab vidi.

Pēc kontrolpunkta ielādes, to ir jāpārveido par modeļa pirmkodu. To ir iespējams paveikt izmantojot Hugging Face “convert_original_stable_diffusion_to_diffusers” skriptu [14]. Pirms skripta palaišanas ir arī jāielādē Stable Diffusion pamata modelis, uz kura tika uztrenēts jaunais modelis, “v1-inference.yaml” specifikāciju [13]. Kontrolpunkta pārveidošanas skripts tiek palaists izmantojot komandrindas komandu “python convert_original_stable_diffusion_to_diffusers.py”.

Skriptam tika padoti sekojošie parametri:

- checkpoint_path="C:/Users/micha/Desktop/lv-sb-model/lv-sb-1.3_2860.ckpt" (Kontrolpunkta atrašanas direktorijs)
- original_config_file="C:/Users/micha/Desktop/lv-sb-model/v1-inference.yaml" (Pamata Stable Diffusion modeļa specifikācijas faila atrašanas direktorijs)
- scheduler_type="euler-ancestral" (Plānotāja tips)
- image_size=512 (Ģenerēto attēlu izmērs)
- prediction_type="epsilon" (Ģenerēto attēlu paredzes algoritma tips)
- dump_path="C:/Users/micha/Desktop/lv-sb-model" (Izejas modeļa pirmkoda direktorijs)
- device="cuda:0" (Izmantotās grafiskās kartes numurs)

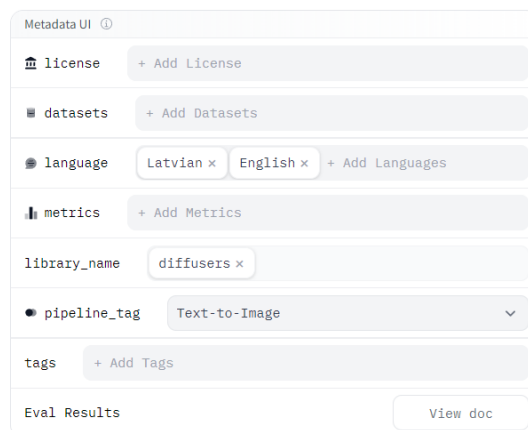
Kad skripts ir veiksmīgi izpildīts direktorijā, kas tika definēts skriptam padotos parametros, ir jāizģenerējas modeļa pirmkodam. Pirmkodu ir iespējams pārbaudīt izmantojot StableTuner lietojumprogrammu [15].



Attēls 8.1.1 StableTuner lietojumprogrammas izvide virknei “a photo of latvijasuniversitate building” izmantojot pielāgota modeļa pirmkodu

8.2. Jaunā modeļa repozitorija izveidošana Hugging Face mājaslapā

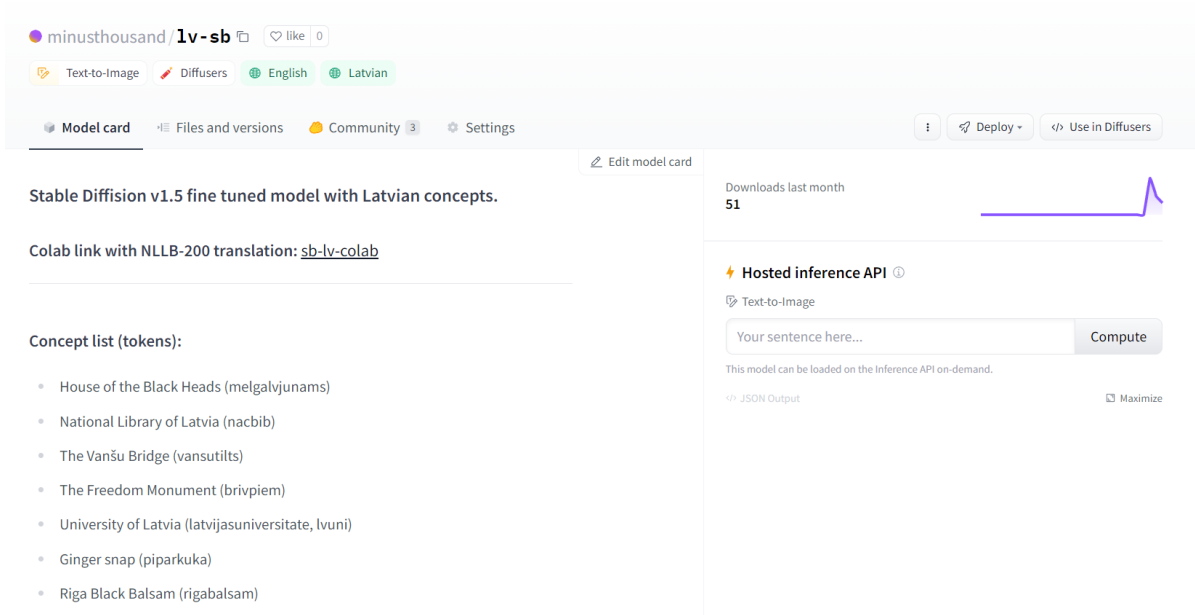
Hugging Face mājaslapā, nospiežot uz lietotājā profila bildes, pēc tam nospiežot “Pievienot modeli”, atvērsies jaunā lapa, kur ir iespējams ievadīt jaunā modeļa nosaukumu, kā arī konfigurēt tā pieejamību. Pēc modeļa un modeļa repozitorija lapas veiksmīgas izveides - jāpievieno modeļa karti, izmantojot “Izveidot modeļa karti” pogu. Metadatu sadaļā modelim pie valodas tika pievienotas valodas, kuras modelis atbalsta, pie bibliotēku nosaukuma pievienots “diffusers” un tika izvēlēts “Text-to-Image” Hugging Face modeļa plūsmas (Pipeline) tips. Metadatu konfigurēšana nodrošinās to, ka modeli būs iespējams izmantot Hugging Face WEB vidē. Nospiežot “Iesūtīt izmaiņas uz main” modeļa karte saglabāsies.



Attēls 8.2.1 Pielāgotā modeļa Hugging Face metadatu iestatījumi

8.3. Modeļa ielāde

Lai ielādētu modeli Hugging Face repozitorijā datorā, no kura notiks ielādē, ir nepieciešama “git” lietotne ar “git LFS” paplašinājumu un “Hugging Face Hub” lietotne. Lai ieinstalētu “Hugging Face Hub” lietotni komandrindā, ir jāpalaiž “python -m pip install huggingface_hub” komanda. Kad instalēšana ir pabeigta, lai ielogotos Hugging Face izmanto “huggingface-cli login” komandu, tas atļaus lietotājam rediģēt izveidoto repozitoriju. Nokopējot modeļa pirmkoda failus Hugging Face repozitorija direktoriijā un izpildot komandrindas komandas “git add .”, “git commit -m “Initial commit” un “git push” modelis tiks ielādēts Hugging Face vidē.



The screenshot shows a Hugging Face model card for 'minusthousand/lv-sb'. The model is a 'Stable Diffusion v1.5 fine tuned model with Latvian concepts'. It includes a Colab link with NLLB-200 translation: [sb-lv-colab](#). The concept list (tokens) includes: House of the Black Heads (melgalvjunams), National Library of Latvia (nacbib), The Vanšu Bridge (vansutilts), The Freedom Monument (brivpiem), University of Latvia (latvijasuniversitate, lvuni), Ginger snap (piparkuka), and Riga Black Balsam (rigabalsam). The card also features a 'Hosted inference API' section with a 'Text-to-Image' interface, a 'Compute' button, and a 'Maximize' option. The card has 51 downloads last month and a 'Deploy' button.

Attēls 8.3.1 Pielāgotā modeļa Hugging Face vide

9. MODEĻA TESTĒŠANA

9.1. Modeļa konceptu testēšana

Pielāgotā modeļa uzģenerētie attēli teksta virknēm, kas tika iztulkoti izmantojot NLLB-200 tulkošanas modeli:

Melngalvju nama ģenerētie attēli, kā arī pārējo objektu ģenerētie attēli, ir līdzīgi trenēšanai izmantotajiem attēliem. Leņķi no kuriem objekts tika attēlots atgādina oriģinālos attēlus.



Attēls 9.1.1 Melngalvju nama ēkas trenēšanas kopas attēli



Attēls 9.1.2 Teksta virknes “Melngalvju nama ēkas bilde” uzģenerētie attēli

Nacionālās Bibliotēkas attēlu ģenerēšanas kvalitāte un precizitāte ir salīdzinoši zema vispārējam ģenerēšanas modelim, tāpēc tika uztrenēts modelis, kas spēj ģenerēt labākus attēlus šim konceptam.=



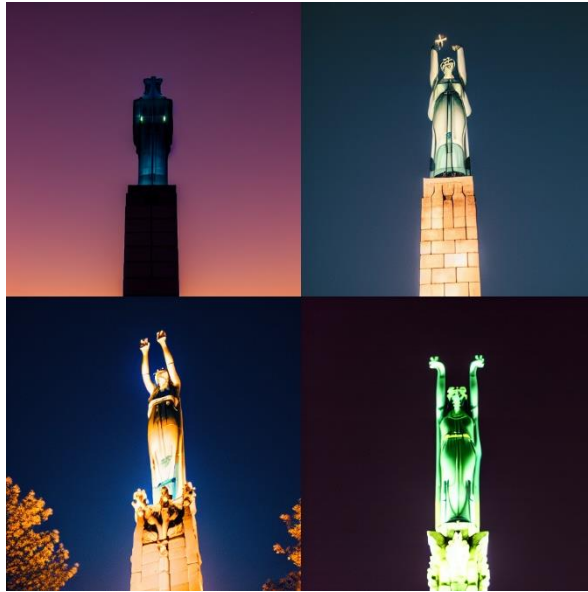
Attēls 9.1.3 Teksta virknes “Nacionālās Bibliotēkas ēkas bilde” uzģenerētie attēli



Attēls 9.1.4 Teksta virknes “Nacionālās Bibliotēkas ēkas bilde” uzģenerētie attēli modelim

“lv-sd-1.4.0_2211”

Brīvības pieminekļa ģenerētie attēli pēc silueta atgādina oriģinālas bildes, tomēr detaļa, kas netiek attēlota ģenerētajos attēlos, ir trīs zvaigznes. Neskatoties uz trenēšanas soļu skaita Dreambooth nespēja uztrenēt modeli, kas konstanti spēs attēlot arī zvaigznes.



Attēls 9.1.5 Teksta virknes “Brīvības piemineklis naktī” uzģenerētie attēli



Attēls 9.1.6 Teksta virknes “Vanšu tilts bilde saulrietā” uzģenerētie attēli



Attēls 9.1.7 Teksta virknes “Latvijas Universitātes ēkas bilde” uzģenerētie attēli



Attēls 9.1.8 Teksta virknes “Piparkūkas uz paplātes” uzģenerētie attēli



Attēls 9.1.9 Teksta virknes “Rīgas balzama pudele” uzģenerētie attēli

9.2. Vājākie ģenerēšanas modeļa aspekti

Attēli, kuri tika ģenerēti izmantojot konceptus “Dziesmu un deju svētki” un “Latviešu rakstu zīmes”, ir ģenerēti ar vissliktāko precizitāti. Modelis spēj uzģenerēt attēlus, kas vizuāli atgādina ieejas datus, bet bez papildus konteksta šīs bildes nevarētu klasificēt kā “Dziesmu un deju svētki” un “Latviešu rakstu zīmes”.



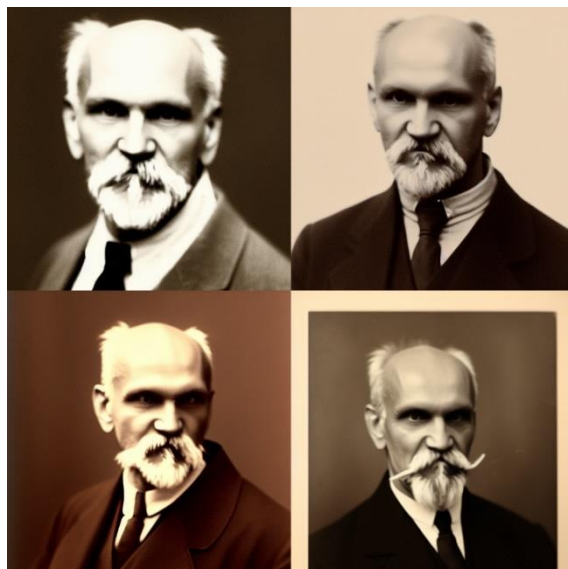
Attēls 9.2.1 Teksta virknes “Dziesmu un deju svētki” uzģenerētie attēli



Attēls 9.2. Teksta virknes “Latviešu rakstu zīmes” uzģenerētie attēli

9.3. Stiprākais ģenerēšanas modeļa aspekts

Attēli, kuri izmantoja konceptu “Rainis” tika ģenerēti ar vislabāko kvalitāti un precizitāti.



Attēls 9.3.1 Teksta virknes “Rainis” uzģenerētie attēli

REZULTĀTI

Šī darba ietvaros tika aplūkoti attēlu ģenerēšanas modeļa Stable Diffusion darbības principi, mašīnmācīšanās tulkošanas modeļa NLLB-200 pielietošana un Dreambooth modeļu trenēšanas pieeja.

Bakalaura darba rezultātā autors aplūkoja Stable Diffusion un difūzijas modeļu uzbūves un darbības principus, kā arī Stable Diffusion modeļa teksta kodētāja darbību un apmācību. Tika arī apskatīts tulkošanas modelis NLLB-200, tā izmantošanu Jupyter Notebook vidē, un pieejamo konfigurāciju pielietojamību latviešu valodas kontekstā. Autors arī aplūkoja BLIP modeļa pamatus un Dreambooth trenēšanas metodi.

Praktiskajā daļā autors izveidoja Google Colab vidi, kurā ir iespējams palaist Python valodas skriptu, kurš automātiski tulkos latviešu valodā ievadīto teksta virkni un ģenerēs no tās Stable Diffusion pielāgotā modeļa, kas tika apmācīts šī darba ietvaros, attēlus. Latviešu valodai pielāgots Stable Diffusion modelis arī tika publicēts Hugging Face mājaslapā.

SECINĀJUMI

Mašīnmācīšanās nozare pēdējos gados attīstījās ļoti strauji un tika izveidoti vairāki dziļās mašīnmācīšanās modeļi, kas atļāva datorsistēmām veikt darbības, kuras iepriekš nebija iespējamas. Viena no tādām darbībām ir attēlu ģenerēšana no teksta ievades, ko spēj paveikt Stable Diffusion modelis.

Stable Diffusion modelis ir noderīgs māksliniekiem ātri ģenerējot attēlus, kurus ir iespējams izmantot jauno mākslas darbu iedvesmai vai ideju vizualizēšanai, datu zinātniekiem, ģenerējot jaunus datus lielos apjomos, lai veiktu jauno modeļu izstrādi, ka arī Stable Diffusion ir pielietojams vispārējam lietotājam ģenerējot personalizētos attēlus personīgo projektu izstrādei.

Stable Diffusion modeļa adaptēšana latviešu valodai sniegs latviešu valodas lietotājiem iespēju izmantot attēlu ģenerēšanas modeļa priekšrocības, ģenerējot attēlus izmantojot savu dzimto valodu un ģenerējot konceptus, kas ir cieši saistīti ar latviešu kultūru.

Turpmākajos darbos var papildināt modeļa latviešu konceptu zināšanu datu kopu, kā arī uzlabojot tulkošanas modeli labāko rezultātu iegūšanai.

IZMANTOTĀ LITERATŪRA

1. Robin Rombach, High-Resolution Image Synthesis with Latent Diffusion Models, arXivLabs, 2022, 13. aprīlis:
<https://arxiv.org/pdf/2112.10752.pdf> [pēdējo reizi skatīts 23.05.2023]
2. Suraj Patil, Pedro Cuenca, Nathan Lambert un Patrick von Platen, Stable Diffusion with Diffusers, Hugging Face Blog, 2022, 22 augusts:
https://huggingface.co/blog/stable_diffusion [pēdējo reizi skatīts 23.05.2023]
3. Olaf Ronneberger, U-Net: Convolutional Networks for Biomedical Image Segmentation, arXivLabs, 2015, 18 maijs:
<https://arxiv.org/abs/1505.04597> [pēdējo reizi skatīts 23.05.2023]
4. Jong Wook Kim, Learning Transferable Visual Models From Natural Language Supervision, arXivLabs, 2021, 25. februāris:
<https://arxiv.org/pdf/2103.00020.pdf> [pēdējo reizi skatīts 23.05.2023]
5. Computer Vision & Learning research group (CompVis), Stable Diffusion, GitHub repozitorijs:
<https://github.com/CompVis/stable-diffusion> [pēdējo reizi skatīts 23.05.2023]
6. NLLB Research Team, No Language Left Behind: Scaling Human-Centered Machine Translation, Meta Research, 2022, 6. jūlijs:
<https://research.facebook.com/file/585831413174038/No-Language-Left-Behind--Scaling-Human-Centered-Machine-Translation.pdf> [pēdējo reizi skatīts 23.05.2023]
7. Meta Research, Languages in FLORES-200, GitHub repozitorijs:
<https://github.com/facebookresearch/flores/blob/main/flores200/README.md#languages-in-flores-200> [pēdējo reizi skatīts 23.05.2023]
8. Meta Research, The FLORES-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation, Meta AI, 2021, 1. jūnijs:
<https://ai.facebook.com/research/publications/the-flores-101-evaluation-benchmark-for-low-resource-and-multilingual-machine-translation> [pēdējo reizi skatīts 23.05.2023]

9. Zongcai Du, Anchor-based Plain Net for Mobile Image Super-Resolution, arXivLabs, 2021, 25. septembris:
<https://arxiv.org/pdf/2105.09750.pdf> [pēdējo reizi skatīts 23.05.2023]
10. Junnan Li Dr, BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation, arXivLabs, 2022, 15. februāris:
<https://arxiv.org/pdf/2201.12086.pdf> [pēdējo reizi skatīts 23.05.2023]
11. Nataniel Ruiz, DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation, arXivLabs, 2023, 15. marts:
<https://arxiv.org/pdf/2208.12242.pdf> [pēdējo reizi skatīts 23.05.2023]
12. Dreambooth Extension for Stable-Diffusion-WebUI, GitHub repozitorijs:
https://github.com/d8ahazard/sd_dreambooth_extension [pēdējo reizi skatīts 23.05.2023]
13. R. Rombach, A. Blattmann, D. Lorenz, P. Esser and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, 27. septembris:
<https://ieeexplore.ieee.org/document/9878449> [pēdējo reizi skatīts 23.05.2023]
14. Hugging Face, Hugging Face “convert_original_stable_diffusion_to_diffusers” skripts, GitHub repozitorijs:
https://github.com/huggingface/diffusers/blob/main/scripts/convert_original_stable_diffusion_to_diffusers.py [pēdējo reizi skatīts 23.05.2023]
15. Stable Tuner, Fine-tune your SD, GitHub repozitorijs:
<https://github.com/devilismyfriend/StableTuner> [pēdējo reizi skatīts 23.05.2023]
16. Hugging Face, Getting Started with Repositories, Hugging Face documentation:
<https://huggingface.co/docs/hub/repositories-getting-started> [pēdējo reizi skatīts 23.05.2023]

PIELIKUMI

1. Pielikums

Jupyter Notebook skripts

```
#Required to run with GPU hardware accelaration
!nvidia-smi

#Install dependencies
!pip install torch torchvision torchaudio --index-url
https://download.pytorch.org/whl/cu117
!pip install accelerate
!pip install --upgrade diffusers transformers scipy
!pip install ipywidgets
!pip install huggingface-hub

#Login to Hugging Face
from huggingface_hub import notebook_login
notebook_login()

prompt = "Latvijas Universitātes ēkas bilde"
prompt_lenght = 30

token_dict = {
    'melngalvju nams ': 'melgalvjunams',
    'melngalvju nama ': 'melgalvjunams',
    'melngalvju namā ': 'melgalvjunams',
    'melngalvju namam ': 'melgalvjunams',
    'melngalvju namu ': 'melgalvjunams',
    'nacionālā bibliotēka ': 'nacbib',
    'nacionālās bibliotēkas ': 'nacbib',
    'nacionālā bibliotēkai ': 'nacbib',
    'nacionālā bibliotēku ': 'nacbib',
    'nacionālā bibliotēkā ': 'nacbib',
    'vanšu tilts ': 'vansutilts bridge',
    'vanšu tilta ': 'vansutilts bridge',
    'vanšu tiltā ': 'vansutilts bridge',
    'vanšu tiltu ': 'vansutilts bridge',
    'vanšu tiltam ': 'vansutilts bridge',
    'brīvības piemineklis ': 'brivpiem statue',
    'brīvības pieminekļa ': 'brivpiem statue',
    'brīvības piemineklim ': 'brivpiem statue',
    'brīvības pieminekli ': 'brivpiem statue',
    'latvijas universitāte ': 'latvijasuniversitate',
    'latvijas universitātes ': 'latvijasuniversitate',
    'latvijas universitāti ': 'latvijasuniversitate',
    'latvijas universitāšu ': 'latvijasuniversitate',
    'latvijas universitātē ': 'latvijasuniversitate',
    'piparkūkas ': 'piparkuka',
    'piparkūka ': 'piparkuka',
    'piparkūku ': 'piparkuka',
    'piparkūkām ': 'piparkuka',
}
```

```

'piparkūkā ': 'piparkuka',
'piparkūkās ': 'piparkuka',
'rīgas melnais balzams ': 'rigabalsam',
'rīgas balzams ': 'rigabalsam',
'rīgas melno balzamu ': 'rigabalsam',
'rīgas balzamu ': 'rigabalsam',
'rīgas melnā balzama ': 'rigabalsam',
'rīgas balzama ': 'rigabalsam',
'rīgas melnajam balzamam ': 'rigabalsam',
'rīgas balzamam ': 'rigabalsam',
'latviešu rakstu zīmes ': 'latviesurakstuzimes',
'latvju rakstu zīmes ': 'latviesurakstuzimes',
'dziesmu svētki ': 'dziesmusvetki',
'dziesmu svētkos ': 'dziesmusvetki',
'dziesmu svētku ': 'dziesmusvetki',
'dziesmu svētkiem ': 'dziesmusvetki'
'dziesmu un deju svētki ': 'dziesmusvetki',
'dziesmu un deju svētkos ': 'dziesmusvetki',
'dziesmu un deju svētku ': 'dziesmusvetki',
'dziesmu un deju svētkiem ': 'dziesmusvetki',
'rainis ': 'rainis',
'rainim ': 'rainis',
'raiņa ': 'rainis',
}

#Remove tokens before translating

tokens_to_add = []

prompt = prompt.lower()

for key, value in token_dict.items():
    if (prompt.find(key) != -1):
        tokens_to_add.append(value)
        prompt = prompt.replace(key, '')

#Translate prompt
#Use higher parameter count nllb-200 model for better translations.
(Requires more RAM)

from transformers import AutoModelForSeq2SeqLM, AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained(
    "facebook/nllb-200-distilled-600M", use_auth_token=True,
    src_lang="lvs_Latn"
)
model = AutoModelForSeq2SeqLM.from_pretrained("facebook/nllb-200-
distilled-600M", use_auth_token=True)

inputs = tokenizer(prompt, return_tensors="pt")

translated_tokens = model.generate(

```

```

        **inputs, forced_bos_token_id=tokenizer.lang_code_to_id["eng_Latn"],
max_length=prompt_lenght
    )
    prompt = tokenizer.batch_decode(translated_tokens,
skip_special_tokens=True)[0]
    prompt

#Add tokens back to prompt

    prompt = ' '.join(tokens_to_add) + ' ' + prompt
    prompt

#Generate image

    from diffusers import StableDiffusionPipeline
    import torch

    model_id = "minusthousand/lv-sb"
    pipe = StableDiffusionPipeline.from_pretrained(model_id,
torch_dtype=torch.float16)
    pipe = pipe.to("cuda")

    image = pipe(prompt).images[0]

    image.show()

```