TARTU ÜLIKOOL
Loodus- ja täppisteaduste valdkond
Arvutiteaduse instituut
Informaatika õppekava

Triin Schaffrik

Pildiandmestiku loomine semantilise difusioonimudeliga

Bakalaureusetöö (9 EAP)

Juhendaja: Joosep Kivastik

Tartu 2023

Pildiandmestiku loomine semantilise difusioonimudeliga

Lühikokkuvõte: Semantiline difusioonimudel on närvivõrk, mis võimaldab piltide genereerimist segmentatsiooni põhjal. Võimalus pilte genereerida nende manuaalselt tegemise asemel võimaldaks hoida kokku aega segmenteeritud pildiandmestiku loomisel. Bakalaureusetöö eesmärgiks on genereerida semantilise difusioonimudeli abil pildid segmentatsioonimudeli treeninghulka ning hinnata, kas need parandavad mudeli tulemusi. Töös genereeritakse 1640 pilti ADE20K treeningandmestiku põhjal. Genereeritud pildid lisatakse samale treeninghulgale, millega treenitakse DeepLabv3+ segmentatsioonimudel. DeepLabv3+ mudelite tulemusi hinnatakse täpsuse, saagise ning F1-skoori põhjal ning võrreldakse omavahel. Tulemuseks on semantilise difusioonimudeli poolt loodud pildid ning nende piltidega treenitud mudeli võrdlus puhtalt ADE20K treenitud mudeliga.

Võtmesõnad:
Semantiline difusioonimudel, segmenteeritud pildiandmestik, DeepLabv3+ närvivõrk

CERCS: P175 Informaatika, süsteemiteooria, P176 Tehisintellekt

Creating Image Dataset With Semantic Diffusion Model

Abstract: Semantic diffusion model is a neural network that allows image generation based on segmentation. The ability to generate images instead of manually creating them would save time in creating segmented image datasets. The goal of this bachelor's thesis is to generate images using the semantic diffusion model and evaluate whether they improve the results of the segmentation model. 1640 images are generated based on the ADE20K training dataset. The generated images are added to the same training set used to train the DeepLabv3+ segmentation model. The results of the DeepLabv3+ models are evaluated based on accuracy, recall, and F1-score and compared with each other. The outcome includes the images created by the semantic diffusion model and a

comparison of the model trained with these images with the model trained purely on the ADE20K dataset.

Keywords:

Semantic diffusionmodel, segmented image dataset, DeepLabv3+ neural network

CERCS: P175 Informatics, systems theory, P176 Artificial intelligence

Sisukord

Kasutatavad mõisted ja terminid

ASPP	Laiendatud ruumiline püramiidkoondumine (*ingl. Atrous Spatial Pyramid Pooling*)
CNN	Konvolutsiooniline tehisnärvivõrk on põhiliselt pilditöötluseks kasutatav tehisnärvivõrgu tüüp [**?**] (*ingl. Convolutional neural networks*)
FID	Fréchet' alguskaugus on meetod kahe tõenäosusliku jaotuse erinevuse võrdlemiseks [1]. Väiksem FID näitab väiksemat erinevust kahe pildi vahel [1]. (*ingl. Fréchet inception distance*)
GAN	Generatiivne võistlusvõrk (*ingl. Generative adversal network*)
Gaussi müra	Gaussi müra on statistiline müra, kus piksli tõelisele väärtusele on lisatud juhuslik väärtus [2]. (*ingl. Gaussian noise*)
LPIPS	Õpitud tajutav piltiosade sarnasus on mõõdik, mida kasutatakse, et hinnata sarnasust kahe pildi vahel (*ingl. Learned Perceptual Image Patch Similarity*)
Markovi ahel	Markovi ahel on mudel, mida kasutatakse tehisnärvivõrkude ehitamiseks [3]. Markovi ahelas on iga lüli sõltuv vaid sellele eelnevast [4]. (*ingl. Markov chain*)
SDM	Semantiline difusioonimudel (*ingl. Semantic diffusion model*)
Stiimulõpe	Stiimulõpe on masinõppe meetod, kus mudel õpib kasutades keskkonnast saadud tagasisidet [5]. (*ingl. Reinforcement learning*)
U-Net	U-Net on sügava õppe arhitektuur, mis on nime saanud oma U-kujulise struktuuri järgi [6].
VAE	Variatsiooniline autokodeerija (*ingl. Variational autoencoder*)

Sissejuhatus

Autonoomsete sõidukite juures on oluline osa väliskeskkonna tajumine ja sealt objektide ära tundmine. Need kasutavad hulga erinevaid sensoreid, kuid nendest alati ei piisa, et sõit sujuvalt kulgeks ning objektide ära tundmiseks tuleb kasutada ka kaamerapilti [7]. Pildilt objektide tuvastamiseks kasutatakse tehisnärvivõrke [7], mida tuleb treenida märgendatud pildihulgaga segmenteeritud pildihulgaga. Pildiandmestiku märgendamine on aga aeganõudev protsess, mille käigus tuleb pildid teha ning seejärel need manuaalselt märgendada. Käsitsi fotode tegemine võtab kas väga kaua aega või ei ole nende varieeruvus eriti suur, mistõttu on treenitud mudel kallutatud. Veelgi kauem võtab aega fotode märgendamine, mida tuleb teha manuaalselt iga foto puhul eraldi. On olemas küll abivahendeid piltide segmenteerimiseks, näiteks Meta AI poolt loodud *"Segment Anything"* [8], kuid isegi see ei kaota ära manuaalselt tehtavat tööd. *"Segment Anything"* mudeli keskmine IoU (*ingl. Intersection over union*) on küll erinevatel andmehulkadel üle 90% [8], kuid treenides segmentatsioonimudelit soovime, et treeningandmed oleks võimalikult täpsed.

Töö eesmärgiks on semantilise difusioonimudeli (*ingl. semantic diffusion model*) [9] abil luua üldine segmenteeritud pildiandmestik, mis võtab sisendiks segmenteeritud pildi ja genereerib selle põhjal realistliku väljundpildi, mis on võrdeline käsitsi tehtud fotole. Mudeli genereeritud pildid ei ole kunagi samasugused [9] ning sealt tuleb variatsioon pildiandmestikku. Samuti võimaldab mudel genereerida ühest segmenteeritud pildist mitu väljundit, mis vähendab oluliselt tööhulka.

Antud töö eesmärgiks on samuti testida, kas loodud pildid on piisavalt realistlikud, et neid kasutada tuvastusmudelite treenimiseks. Mudeli loojate poolt läbi viidud uuringus selgus, et semantilise difusioonimudeli poolt genereeritud pilte eelistatakse kuni 94% juhtudest võrreldes seniste alternatiividega [9, 10]. Mudeli tulemused on küll paremad võrreldes teiste mudelitega, kuid pole kindel, et kvaliteet on piisav, et asendada tõeliseid fotosid.

Samuti genereeritakse ühest segmentatsioonist mitu väljundpilti. Töös vaadeldakse, kas väljundpildid on piisavalt erinevad, et lisada pildiandmestikule lisandväärtust.

Tulemuste hindamiseks treenitakse DeepLabv3+ närvivõrk [11], mis on loodud piltide segmenteerimiseks. Valitud sai DeepLabv3+, sest on näidanud kõrget täpsust semantilises segmentatsioonis erinevatel andmestikel [11]. Treenitakse kaks närvivõrku, neist esimene vaid ADE20K treeninghulgaga ning teine sama treeninghulgaga, millele on lisatud genereeritud pildid. Närvivõrku võrreldakse seejärel tõeliste piltidega treenitud närvivõrguga, et vaadata, kas segmenteerimistulemustes on märgata vahet.

Töö esimeses peatükis käsitletakse töö teoreetilist tausta, kirjeldades sealjuures lähemalt kasutatavat mudelit, selle struktuuri ja ka teisi sarnaseid mudeleid. Teises peatükis kirjeldatakse töö metoodikat: ülevaadet kasutavast pildiandmestikust, segmenteeritud piltide eeltöötlusest ning andmestiku genereerimisprotsessist. Kolmandas peatükis hinnatakse mudeli tulemusi varasemalt kirjeldatud meetoditel.. Bakalaureusetöö viimases peatükis võetakse kokku tulemused.

Projekt on leitav GitHubist lingil https://github.com/TriinSchaffrik/Creating-dataset-via-SDM.

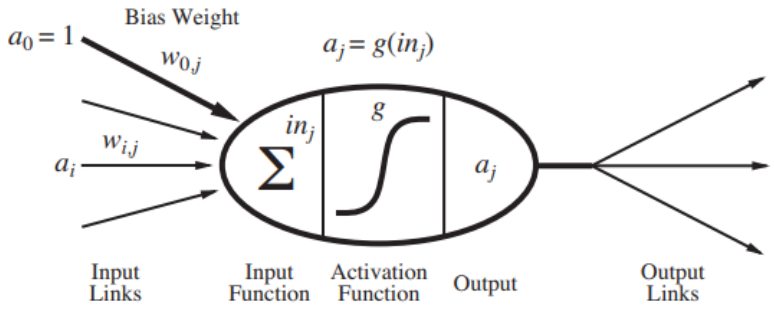
1 Teoreetiline taust

Käesolevas peatükis on toodud teoreetiline taust difusioonimudelitest üldiselt ning töös kasutatavast semantilisest difusioonimudelist. Veel on toodud võrdluseks populaarsemad alternatiivid, mida samuti kasutatakse piltide genereerimiseks.

1.1 Tehisnärvivõrgud

Tehisnärvivõrgud (*ingl. artificial neural networks*) on sügavõppe algoritmid, mis on inspireeritud inimese aju struktuurist [12]. Need võimaldavad märgata mustreid andmetes ning teha nende põhjal ennustusi. Tänapäeval kasutatakse neid mitmetes valdkondades, sealhulgas piltide segmenteerimiseks ja genereerimiseks [13].

Alljärgnev lõik on refereeritud Stuart J Russell jt raamatust ”Artificial intelligence a modern approach third edition” [12]. Tehisnärvivõrk koosneb neuronitest, sarnaselt inimeste närvisüsteemile. Neuronit on kirjeldatud joonisel 1. Iga neuron saab sisendid (a_i) kas teistelt neuronitelt või väliskeskkonnast ja igal saadud sisendil on oma kaal (w_{ij}). Neuronis summeeritakse sisendid võttes arvesse ka nende kaalusid. Saadud tulemusele in_j rakendatakse seejärel aktiveerimisfunktsiooni g, mis määrab, kas neuron väljastab signaali a_j või mitte. Närvivõrgu treenimisel kohandatakse kaale ning vabaliikmeid, mis muudab ennustamise täpsemaks. Tehisnärvivõrk koosneb hulgast neuronitest, mis on



Joonis 1. Tehisnärvivõrgu neuron [12].

jagatud kihtidesse.

Erinevad neuronite arvud, paigutused ja ühendused moodustavad erinevaid tüüpe tehisnärvivõrke, selles töös kasutatakse nii difusioonimudelit kui ka konvolutsioonilist närvivõrku.

1.2 Difusioonimudelid

Difusioonimudelid on genereerivad mudelid, mis tähendab, et mudel loob andmeid, näiteks heli- või pildifaile, mis sarnanevad neile, millega seda mudelit treeniti, kuid saadud andmed pole kunagi ühegi treeningfaili täpsed koopiad [14].

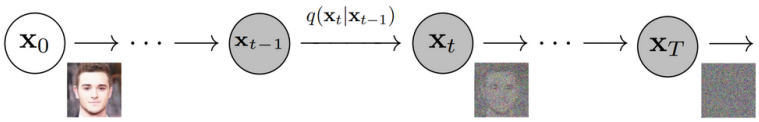
Järgnevad lõigud põhinevad Jonathan Ho jt artiklil [4]. Difusioonimudelid on treenitud kasutades Markovi ahelat, mille igal sammul lisatakse andmetele juhuslikku Gaussi müra kuni lõpuks ongi alles vaid müra. Seejärel õpib mudel pilti stiimulõppega taastama ennustades ning pöörates tagasi lisatud müra.

Mudeli treenimise esimene etapp ehk difusiooni protsess on T sammuga Markovi ahel, kus igal sammul lisatakse andmetele Gaussi müra. Protsessi kirjeldab joonis 2. Markovi eeldus tähendab siinkohal seda, et iga samm on sõltuv vaid eelnevast. Müra lisatakse sõltuvalt standardhälbest β_t ja eelnevast sammust x_{t-1}, saades uue muutuja x_t läbi tõenäosusfunktsiooni $q(x_t|x_{t-1})$, mida saab kirjeldada järgnevalt:

$$q(x_t|x_{t-1}) = N(x_t; \mu_t = \sqrt{1-\beta_t}x_{t-1}; \Sigma_t = B_t I).$$

I tähistab siin ühikmaatriksit, mis näitab seda, et igale piksile rakendatakse sama standardhälvet [15]. On näha, et mida väiksem on β_t seda vähem on lisatud järgmisele muutujale müra [4]. β_t võib olla konstant või funktsioon, näiteks lineaarne või koosinusfunktsioon, mis sõltub sammude arvust.

Difusiooni protsessi tulemuseks saadakse T pilti, millest igal pildil on üha rohkem müra [16]. Difusioonimudelit treenitakse vastupidise difusiooni protsessiga. Alustatakse eelnevalt kirjeldatud protsessi viimasest muutujast x_t ning mudel üritab eemaldada

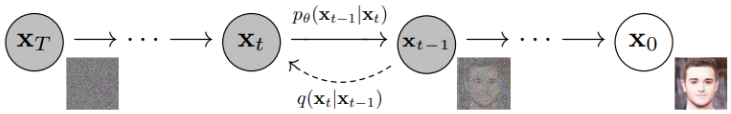


Joonis 2. Difusiooni protsess [17].

sellelt Gaussi müra nii, et saadakse muutuja x_{t-1}. Seda T korda rakendades saadakse tulemuseks algne pilt. Kui oleks teada $q(x_t|x_{t-1})$ saaks muutujast x_t käivitada täpselt vastupidist difusiooni protsessi, et saada tagasi algsed andmed. Kahjuks sõltub $q(x_t|x_{t-1})$ tervest andmete jaotusest, seega kasutatakse funktsiooni $p_\theta(x_{t-1}|x_t)$, kus

$$p_\theta(x_{t-1}|x_t) := N(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)).$$

Funktsiooni $p_\theta(x_{t-1}|x_t)$ optimeeritakse, et lähendada μ_t ja Σ_t parameetritele, mis saadi difusiooni protsessil sammul t. Eelnev lõik on refereeritud Jonathan Ho jt artiklist [4].



Joonis 3. Vastupidine difusiooni protsess [4].

Joonisel 3 on näitlikustatud funktsioonide kasutamist ja vastupidise difusiooni protsessi.

1.3 Semantiline difusioonimudel

Semantiline difusioonimudel võtab sisendiks segmenteeritud pildi ning genereerib sellele realistliku vaste [9]. See on justkui vastupidine protsess semantilisele segmenteerimisele. Semantiline segmenteerimine on protsess, kus fotol tähistatakse iga piksel vastavalt selle kuulumisele mingisse kategooriasse [18]. Näiteks eristatakse pildil taevas, hooned,

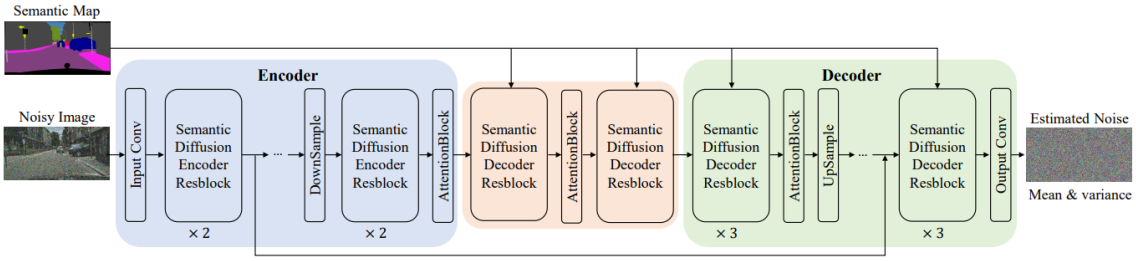
maapind jne. Joonisel 4 on näidatud fotot ning selle segmenteeritud vastet. On näha, et iga objekti kategooria on erinevat värvi, kuid objektid ise pole eraldi märgistatud. Semantilist segmentatsiooni kasutatakse näiteks isesõitvatel autodel.

Joonis 4. Tavaline ja segmenteeritud pilt [19].

Vaadeldav mudel töötab vastupidiselt kirjeldatule. Aluseks on segmenteeritud pilt, kus igale pikslile vastab mingi objekti kategooria ning mudel genereerib selle põhjal realistliku pildi [9]. Semantiline diffusioonimudel on loodud tuginedes varasemalt loodud juhendatud difusiooni mudelile (*ingl. guided diffusion*) [9, 20].

Semantiline diffusioonimudel on närvivõrk, mis on struktuurilt U-Net ning seda kujutab täpsemalt joonis 5 [9]. Joonisel näidatud protsess kujutab ühte difusioonisammu. Kodeerija saab sisendiks mürase pildi ning närvivõrk ennustab sisendpildil olevat müra. Lisaks kodeerijale on mudelis dekodeerija. Eelnenud mudelil [20] kombineeritakse semantiline kaart sisendiks oleva mürase pildiga protsessi alguses. Semantilise difusioonimudeli puhul aga sisestatakse see dekodeerijasse ja sellega suunatakse pildi genereerimisprotsessi. Autorite sõnul kasutab selline lähenemine paremini ära semantilist infot, mille tagajärjel on genereeritud pildid kvaliteetsemad ning semantiliselt asjakohasemad. [9]

Erinevalt varasematele mudelitele kasutab SDM (semantiline difusioonimudel) klassifitseerija abil juhendamise asemel klassifitseerijavaba juhendamist. Juhendamist, nii klassifitseerijaga kui ka klassifitseerijavaba, kasutatakse, et väljund oleks realistlikum ning täpsem semantilisele kaardile. Sel moel väljundi realistlikuse tõus toob kaasa aga ka



Joonis 5. Semantilise difusioonimudeli struktuur [9].

varieeruvuse languse. Klassifitseerijavaba juhendamine tähendab, et pole vaja treenida eraldi klassifitseerija mudelit, vaid treenimise käigus õpitakse funktsioon, mis juhendab sarnaselt klassifitseerijale väljundpildi genereerimist. Klassifitseerija abil juhendatud mudel tõstab FID (Fréchet' alguskaugus) 33.0 pealt 12.0 peale [21].

Töös on kasutatud just semantilist difusioonimudelit, sest see on näidanud tunduvalt paremaid tulemusi võrreldes teiste generatiivsete mudelitega. Mudeli autorite poolt läbi viidud uuringus, kus võrreldi semantilist difusioonimudelit SPADE, INADE ja OASIS mudelitega, selgus, et üle 75% kordadest hindasid küsitletavad SDM genereeritud pilte kvaliteetsemateks [9]. Lisaks on võrreldud semantilise difusioonimudeli FID ja LPIPS tulemusi alternatiivsetega ning tulemustest on näha, et SDM edestab teisi mudeleid enamike andmestike korral [9].

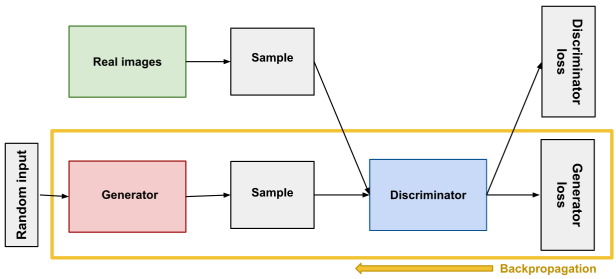
1.4 Seotud tööd

Selles peatükis kirjeldatakse alternatiive semantilisele difusioonimudelile. Piltide genereerimiseks kasutatakse tihti generatiivseid võistlusvõrke (GAN) [22], mis võimaldavad sarnaselt semantilisele difusioonimudelile luua pilte semantilise kaardi põhjal. Lisaks on võimalik kasutada ka variatsioonilisi autokoodreid (VAE) ja voopõhiseid generatiivseid mudeleid, kuid GAN mudelid on näidanud paremaid tulemusi just piltide genereerimisel ning on seetõttu teistest eelistatumad [23].

Peatükis on GAN mudelite kõrval kirjeldatud ka stabiilset difusioonimudelit, mis on kogunud üha rohkem populaarsust oma heade tulemuste tõttu piltide genereerimisel.

1.4.1 Generatiivsed võistlusvõrgud

Populaarseimaks alternatiiviks piltide genereerimisel difusioonile on generatiivsed võistlusvõrgud (*ingl. Generative adversarial networks*) [20, 24]. GAN koosneb kahest omavahel võistlevast närvivõrgust - diskriminaator ja generaator. Generatiivsete võistlusvõrkude struktuur on toodud joonisel 6. Generaatori ülesandeks on luua müra põhjal väljundpilt, mille järel diskriminaator püüab ära arvata, kas tegemist on genereeritud pildi või reaalse pildiga. Treenimise käigus õpib diskriminaator paremini ära arvama, millisesse kategooriasse pilt kuulub ning selle põhjal õpib generaator looma realistlikumat pilti. Eelnev lõik on refereeritud Ian J. Goodfellow jt artliklist "Generative adversarial networks" [23].



Joonis 6. Generatiivsete võistlusvõrkude struktuur [25].

1.4.2 Stabiilne difusioon

Stabiilne difusioon on difusioonimudel, mis võtab sisendiks teksti ja genereerib selle põhjal kvaliteetse pildi [16]. Lisaks tekstile võib sisendiks anda ka pildi, mille põhjal genereerimist alustatakse [16, 26].

Alljärgnevad lõigud tuginevad J Alammar'i artiklil [26]. Stabiilne difusioon töötab latentses ruumis, mis tähendab, et mudeli sisendiks pole enam pilt, vaid on tihendatud

info vektorina. Difusioonimudelite väljundid on sama suured kui andmed, millega neid treeniti - seega treenides mudelit hea kvaliteediga piltidega, saadakse tulemused samuti kvaliteetsed. Treenides mudelit latentses ruumis ei toimu iga piksli läbi vaatamist mistõttu on protsess tunduvalt kiirem ning võimaldab tõsta väljundi kvaliteeti.

Stabiilne difusioon on järjest populaarsust kogunud oma hea kvaliteedi, kiiruse ning ka tekstile vastavuse poolest. Mudel genereerib realistlikke väljundpilte, mis on suure varieeruvusega ning on võrdelised fotodega, mis on tehtud käsitsi. Stabiilse difusiooni mudelile saame sisendiks anda samuti ka pildi, kuid seda ei kasutata semantilise kaardina vaid lihtsalt aluspõhjana, millest edasi pilti genereerida vastavalt tekstisisendile. Selle põhjuse tõttu ei ole stabiilne difusioon antud projektis kasutatav, kuigi selle väljundite kvaliteet ja varieeruvus on suurem.

2 Metoodika

Selles peatükis on kirjeldatud töös kasutatavaid segmenteeritud pildiandmestikke ning piltide genereerimisprotsessi. Sealhulgas andmete eeltöötlust ning genereerimiseks kasutatavat mudelit.

2.1 Pildiandmestik ADE20K

Semantilise difusioonimudeli treenimiseks on kasutatud ADE20K andmehulka, mille treenimishulgas on 20 210 pildipaari [27]. Iga pildipaar sisaldab tõelist pilti ning selle segmenteeritud vastet [27]. Segmenteeritud pildil vastavad pikslite väärtused nende kategooriate indeksitele.

ADE20K andmestik on loodud 2018. aastal ning sisaldab pilte erinevatest kohtadest ning objektidest, iga pilt on segmenteeritud manuaalselt. ADE20K loomisel on kategooriaid segmenteerimise käigus lisatud, vastupidiselt teistele segmenteeritud andmestikele, kus kategooriate hulk on algusest paigas. See tagab, et pildil olevad objektid on märgistatud võimalikult täpselt. Seda kinnitab keskmine kategooriate arv pildil 9.9 võrreldes COCO andmestikuga [19], kus vastav arv on 3.5[27].

Töös on kasutatud ADE20K andmehulgaga eeltreenitud mudelit [28], kuna eeltreenitud mudelid on olemas nelja andmehulgaga: Cityscape, ADE20K, CelebAMask-HQ ja COCO-stuff. Neist andmehulkadest on ADE20K kõige üldisem ning tundus seetõttu neist sobivaim.

Cityscape on sisaldab linnafotosid ja CelebAMask-HQ sisaldab inimeste nägusid, seega pole need sobivad antud tööle oma liigse spetsiifilisuse tõttu. Kuigi COCO-stuff treeninghulgas on pildipaare ligi 96 tuhat rohkem [27], siis valituks sai ADE20K andmestikuga treenitud mudel, kuna COCO-stuff kategooriates puuduvad klassid, mida tihti piltidelt on leida, näiteks inimesed, loomad, masinad jne. Töös tahetakse kasutada andmestikku, mis on piisavalt üldine ja täielikult märgendatud.

ADE20K sisaldab endas 151 erinevat kategooriat. Igast klassist on treeninghulgas andmeid vähemalt 50 pildil [27].

2.2 Andmete eeltöötlus

Mudel võtab sisendiks pildi kujul $w \times h \times n$, kus w tähistab pildi laiust, h tähistab kõrgust ning n piksi väärtust. Antud mudel töötleb 256×256 mõõdus pilte. Kui laius või kõrgus on suuremad, siis muudetakse nende suuruseid andmete sisselugemisel. Seega piltide õigesse mõõtu viimine varasemalt pole vajalik. Piksli väärtused tähistavad objektiklassi, näiteks piksel, mis tähistab vett on kujul $(22, 22, 22)$. Mudel võtab teadmise, millise indeksile peab vastama milline kategooria treenimisprotsessist. Seetõttu on oluline, et kategooriad, millega pilte genereerima hakatakse vastaks kategooriatele, millega mudelit treeniti.

2.3 Mudel

Piltide genereerimiseks kasutatakse semantilist difusioonimudelit, mis on varasemalt treenitud ADE20K andmestikuga [28]. Nii treenimiseks kui piltide genereerimiseks kasutatud parameetrid on toodud tabelis 1. Tabelis on jäetud väärtuste lahtrid tühjaks juhul, kui neid ei ole vastavas protsessis kasutatud.

Andmestiku loomiseks on vaja, et ühele segmenteeritud pildile genereeritakse rohkem vasteid kui üks. Selleks on muudetud klassi image_train, mis parameetri num_samples põhjal genereerib igale pildile just nii mitu vastet. Näiteks kui sisendhulga suurus on 5 ning parameetri väärtus on 3, genereeritakse igale pildile 3 vastet ehk kokku 15 pilti.

Tulemused salvestatakse kausta, mille nimeks on genereerimise alustamise kuupäev. Kaustas on olemas nii algsed pildid, genereeritud pildid ning ka segmenteeritud pildid.

Genereerimisel oleks parema kvaliteedi huvides soovitatav kasutada sarnaseid parameetreid kui mudeli treenimisel. Kui difusioonisammude arv on treenimisel 1000, siis

Tabel 1. Semantilise difusioonimudeli treenimisel ja testimisel kasutatud parameetrid.

Parameetri kirjeldus	**Parameetri nimetus**	**Treenimisel kasutatud väärtus**	**Genereerimisel kasutatud väärtus**
Kategooriate arv	num_classes	151	151
Määrab, kuidas andmestik sisse loetakse	dataset_mode	ade20k	ade20k
Difusioonisammude arv	diffusion_steps	1000	1000
Väljundpildi suurus	image_size	256	256
Klassifitseerijavaba juhendamise tugevus	s		1.5
Treenitud mudeli asukoht	model_path		ema_0.9999_best.pt
Õpikiirus	lr	0.0001	
Ploki suurus	batch_size	4	
Määrab, millisest kaustast andmeid loetakse	is_training	True	False

võiks see olla sama ka pildi loomisel. Difusioonisammude arvu suurendamisel või vähendamisel Erinevusteks on varasemalt mainitud andmestiku suurus ning klassifitseerijavaba juhendamise tugevus parameetri s all. SDM loojad on leidnud, et ADE20K andmestikuga treenitud mudeli korral on tulemused parimad, kui s väärtus on 1.5 [9]. Töös kasutatud parameetrid on olemas failis sample.sh. Nvidia GeForce RTX 4090 graafikakaarti kasutades kulus ühe pildi genereerimiseks 5 minutit ja 49 sekundit. Bakalaureustöö raames genereeriti kokku 1640 pilti, mis lisati ADE20K treenimishulgale. Genereeritud piltide osakaal uuest treeninghulgast on 7.5%.

Kümme näidet genereeritud piltidest on toodud lisas 1. Visuaalsel hinnangul on genereeritud pildid üldiselt realistlikud ning kvaliteetsed. Probleeme esineb inimeste ja loomade genereerimisega ning piltidega, mis on rohkete segmentidega ning kategooriate arv pildil on kõrge. Selle tulemusel näevad pildid välja ebaloomulikud.

3 Tulemused

Hindamiseks, kas soovitud eesmärk sai saavutatud või mitte, kasutatakse DeepLabv3+ mudelit [29]. Mudel treenitakse nii reaalsete piltidega kui ka pildihulgaga, kus reaalsed fotod ning genereeritud pildid on segamini. Treenitud mudeleid hinnatakse nende täpsuse, saagise ja F1-skoori põhjal.

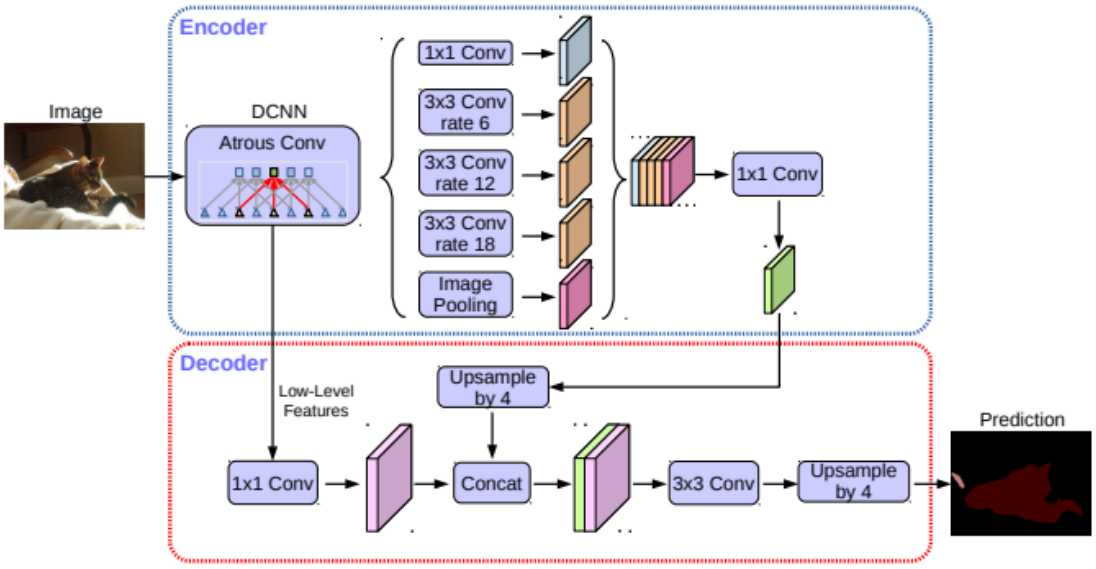
3.1 DeepLabv3+

DeepLabv3+ on semantiliseks segmentatsiooniks loodud konvolutsiooniline närvivõrk, mis võimaldab täpset pikslipõhist klassifikatsiooni [30]. Mudelis on rakendatud ASPP ning laiendatud konvolutsiooni (*ingl. Atrous Convolution*) meetodeid, misläbi on suurendatud võimekust eristada detaile isegi suurtelt piltidelt [11]. DeepLabv3+ on edasiarendus DeepLabv3'st, sest kasutab kodeerija-dekodeerija struktuuri, mis võimaldab mudelil saavutada täpsemad ja detailsemad tulemused [11]. Mudeli struktuuri ning ennustusprotsessi on kujutatud ka joonisel 7. Erinevad mudelite võrdlused on kinnitanud, et DeepLabv3+ on üks täpseimatest mudelitest piltide segmenteerimisel [31, 32].

3.1.1 Treenimine

Bakalaureusetöös on kasutatud DeepLabv3+ mudeli implementatsiooni GitHubi projektist "DeepLabv3Plus-Pytorch-ade20k"[33].

Esimene mudel (Mudel 1) treenitakse ADE20K pildiandmestikuga, mille treeninghulk koosneb 20 210 pildist ning valideerimishulk 1000 pildist. Treenitava mudeli testimiseks jagatakse ADE20K valideerimishulk pooleks nii, et valideerimishulka jääb 1000 pilti ning testhulka 1000 pilti. Teise mudeli (Mudel 2) treenimisel lisatakse treeninghulka juurde 1640 pilti, mis teeb treeninghulga suuruseks 21 850 ning valideerimis - ja testhulk jäetakse samaks. Kasutatav iteratsioonide arv on vastavalt mudelile 84 210 ning 91 000, mis ploki suurusega (*ingl. batch size*) 12 teeb epohhide arvuks 50. Treenimiseks



Joonis 7. DeepLabv3+ mudeli struktuur [11].

on kasutatud Nvidia GeForce RTX 4090 graafikakaarti, millega võttis mõlema mudeli korral protsess aega umbes 12 tundi.

Mõlemal treenimiskorral on kasutatavad parameetrid samad, et treenitud mudelite täpsust oleks võimalik omavahel võrrelda. Tabelis 2 on välja toodud parameetrid, mida kasutatakse mudelite treenimisel ning nende vastavad väärtused. Õpikiiruseks määratakse 0.003 ning selle funktsiooniks polünoomfunktsioon. Need mõjutavad kui palju iga iteratsiooni järel mudeli parameetreid muudetakse [34]. Mudeli magistraalvõrguna kasutatakse ResNet101, mis tähendab, et treenimist ei alustata tühjalt, vaid võetakse aluseks nimetatud mudel [11]. ResNet101 on sügav konvolutsiooniline närvivõrk, mis on loodud Microsofti poolt 2015. aastal [35]. Mudel ResNet101 nimi viitab sellele, et tegemist on närvivõrguga, mis võimaldab infol liikuda kihte vahele jättes ning kihtide arv kokku on 101 [35].

Treenimisel kasutatatkse kaofunktsioonina stohhastilist gradientlaskumist [33], mida ei anta ette parameetrina, vaid on implementatsiooni sisse kirjutatud. Minimaalne kao väärtus, mis Mudel 1 treenimisel saavutati oli 0.4776. Mudel 2 kaofunktsiooni väärtus

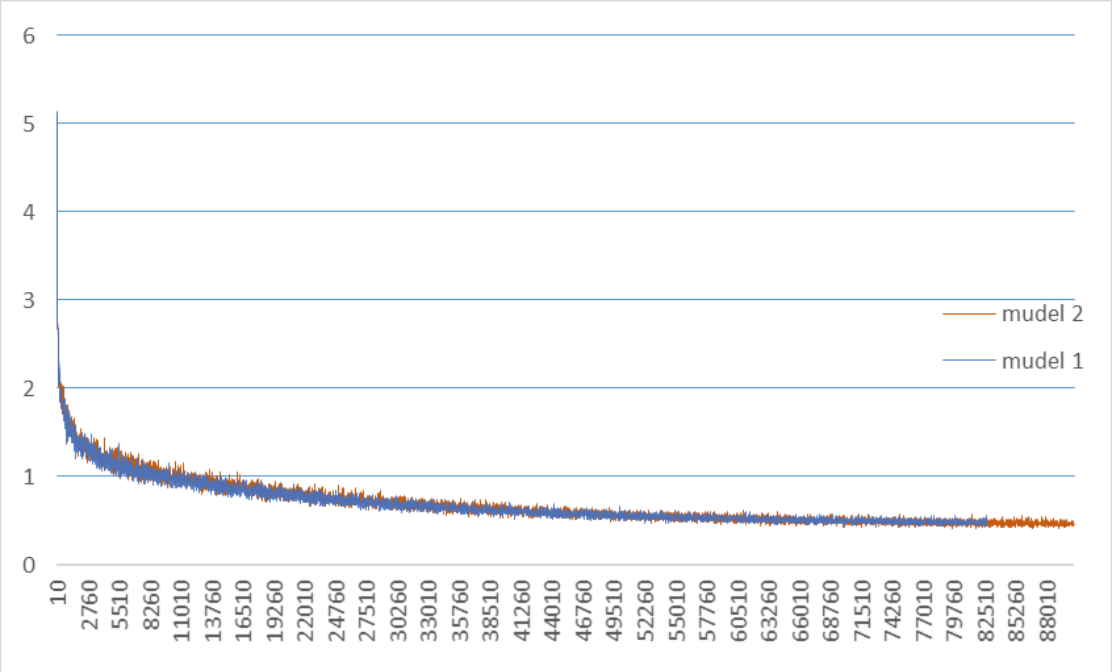
Tabel 2. Treenimisel kasutatavad parameetrid.

Parameetri nimetus	**Parameeter**	**Kasutatav väärtus**
Õpikiirus	lr	0.003
Magistraalvõrk	model	deeplabv3plus_resnet101
Treeningploki suurus	batch_size	12
Valideerimisbloki suurus	val_batch_size	12
Andmestik	dataset	ade20k
Õpikiiruse funktsioon	lr_policy	poly
Kaofunktsioon	loss_type	cross_entropy
Iteratsioonide arv	total_itrs	84 210/91 000
Epohhide arv		50

langes 0.451772'ni. Väärtused on suured, kuid 50 epohhiga stabiliseerunud ning kahanemist enam märgata ei olnud. Kaofunktsiooni väärtused vastavalt iteratsioonidele on toodud joonisel 1, kus sinine joon tähistab Mudel 1 ning punane Mudel 2 väärtuseid. Valideerimishulgal saavutati Mudel 1-ga keskmine õigsus (*ingl. accuracy*) 0.451124 ning keskmine IoU (*ingl. intersection over union*) 0.330128. Teise mudeliga saavutatud keskmine õigsus on 0.452385 ning keskmine IoU 0.326125.

3.1.2 Segmentatsioon testhulgal

Treenitud mudelite testimiseks kasutatakse ADE20K valideerimishulgast eraldatud testhulka, mille suurus on 1000 pilti. Piltide segmenteerimine võttis mõlema mudelil aega alla minuti. Tabelis 3 on välja toodud mõned pildid. Pildid valiti võrreldes mudelite segmentatsioone ning valides välja suurimate erinevustega pildid. Tabeli esimeses tulbas on toodud pilt ADE20K valideerimishulgast [27], teises veerus on toodud selle tõeline segmentatsioon samast andmestikust. Kolmas ning neljas veerg näitavad vastavalt

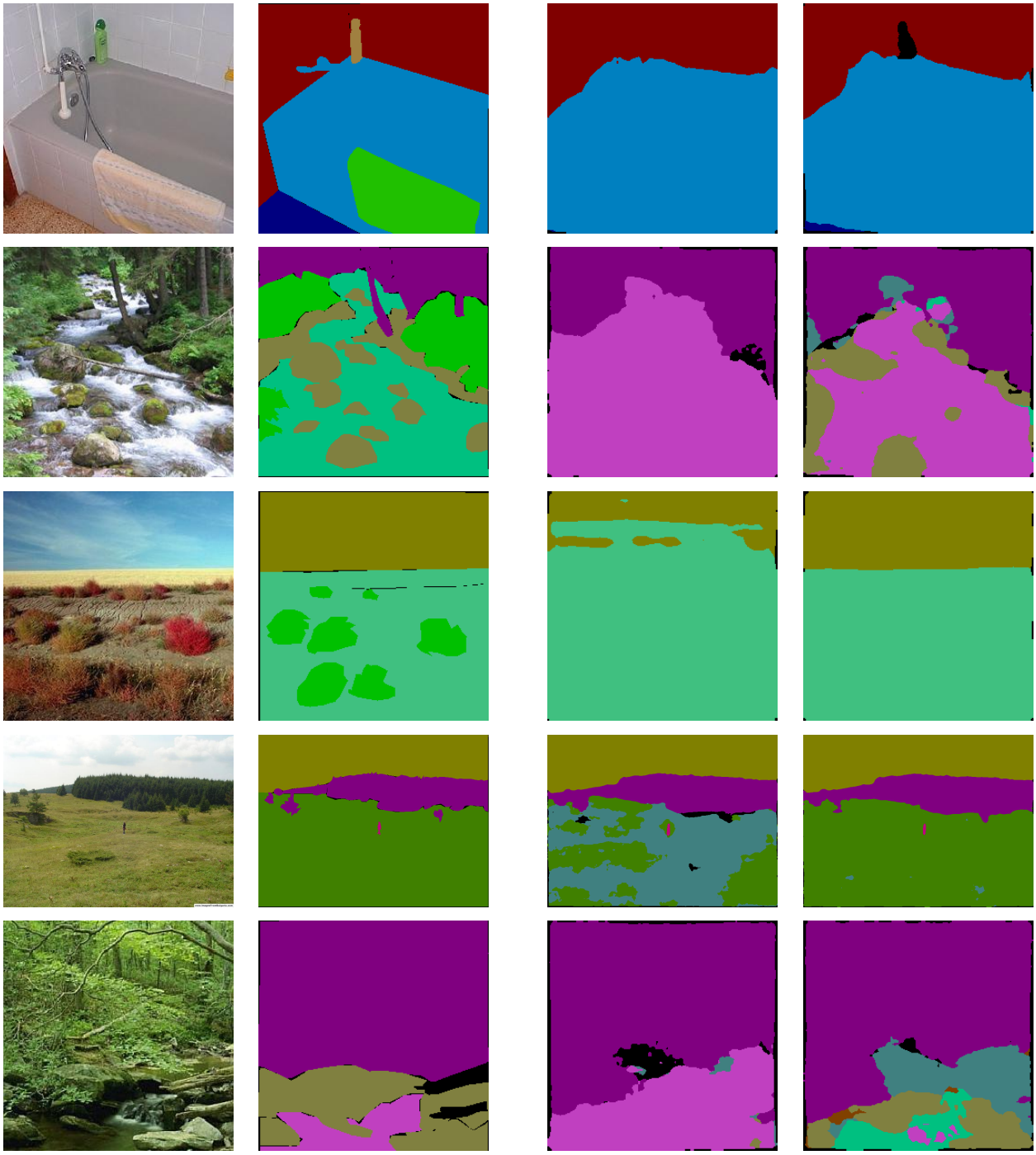


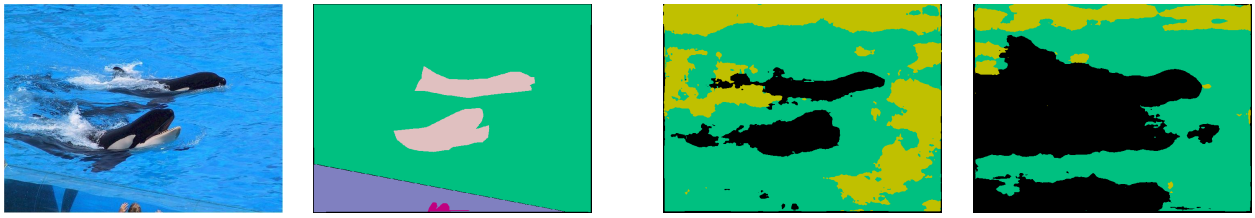
Joonis 8. Kaofunktsiooni väärtused vastavalt iteratsioonide arvule.

mudelite 1 ja 2 segmentatsioone.

Tabel 3. Treenitud mudelite tulemuste võrdlus.

Pilt	**Tõeline segmentatsioon**	**Mudel 1**	**Mudel 2**
			





Esimesel viiel pildil paistab Mudel 2 olevat parema segmenteerimistäpsusega. Mudel 2 saab paremini hakkama suurte alade tuvastamiseg, nagu taevas ja maapind. Piltidel kolmandast reast paistab silma oluline erinevus tõelise segmentatsiooniga. Tõelisel segmentatsioonil on märgitud veekogu kategoorianumbriga 22, mis vastab veele, kuid mõlemad mudelid ennustasid selle kategooriaks jõge. Nimetatud pildilt on Mudel 2 tuvastanud vees olevad kivid, mida Mudel 1 ei tuvastanud.

Oluline visuaalne erinevus on ka viiendas reas olevatel piltidel, kus Mudel 1 liigitas suurema osa maapinnast kategooria alla 14, mis tähistab maapinda, kuid Mudel 2 ja tõeline segmentatsioon tähistavad ala kategooriaga 10, milleks on muru. Mudel 2 on küll lähemal tõelisele segmentatsioonile, kuid ei saa öelda, et ka Mudel 1 ennustus oleks väär.

Viimases kahes reas toodud piltide segmentatsioonid on küllaltki erinevad. Esimesel neist on Mudel 1 ennustanud kategooriale jõgi suuremat ala kui tõelisel segmentatsioonil, samas on kategooria sama. Mudel 2 on tuvastanud pildilt suurema arvu kategooriaid, nagu vesi, kivid, maapind ja puud. Segmentide piire vaadates paistab, et Mudel 2 tuvastab paremini erinevaid objekte ning veekogu .

Vastupidiselt eelnevatele on viimasel pildil Mudel 2 täpsus langenud. Mudel 1 ei ole tuvastanud, et pildil on loomad ning on jätnud piirkonna kategooriata. Mudel 2 pole samuti loomi tuvastanud ning on jätnud kategooriata hoopis suurema piirkonna. Täpsuse langus antud kategooria puhul võib olla tingitud sellest, et semantilise difusioonimudeli poolt loodud piltidel ei ole loomad kuigi realislikud.

3.2 Täpsus, saagis, F1-skoor

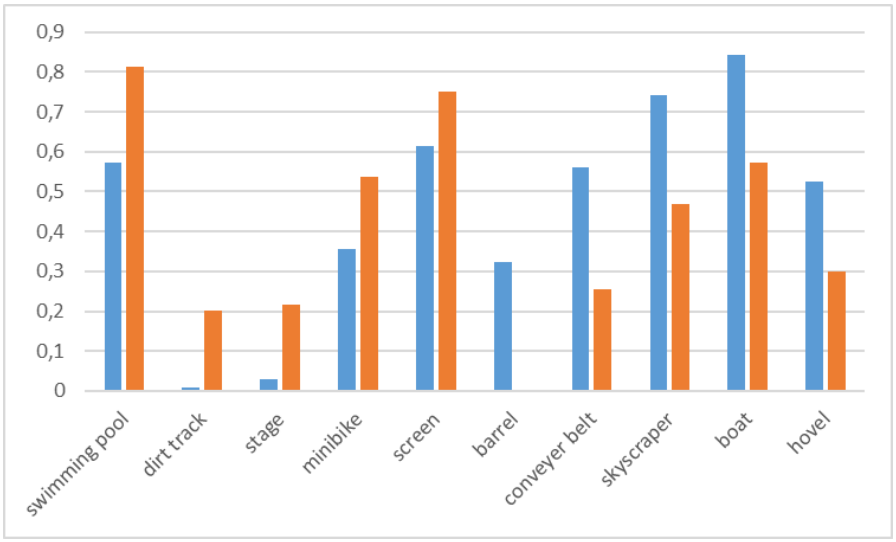
Mudelite võrdlemiseks kasutatakse täpsust (*ingl. precision*), saagist (*ingl. recall*) ning F1-skoori. Kõigi nende väärtuste arvutamiseks kasutame segadusmaatriksit.

Segadusmaatriksi rea ja veeru kombinatsioonid näitavad kui sageli mudel ennustas õigesti või valesti. Õiged positiivsed (TP) näitavad õigesti ennustatud piirkondi, valepositiivsed (FP) kujutavad endast piirkondi, kuhu ennustati vastavat kategooriat, kuid mille tegelik kategooria on teine, ning valenegatiivsed (FN) on segmendid, mille tegelik kategooria vastab ennustatavale, kuid ennustati teist kategooriat.

Täpsus näitab kui suur osa ennustatavast kategooriast määrati pildil õigesti. Selle arvutamiseks kasutatakse järgnevat valemit:

$$Täpsus = \frac{TP}{TP + FP}$$

Joonisel 9 on kahe mudeli täpsused klasside lõikes. Nii sellel kui järgnevatel joonistel



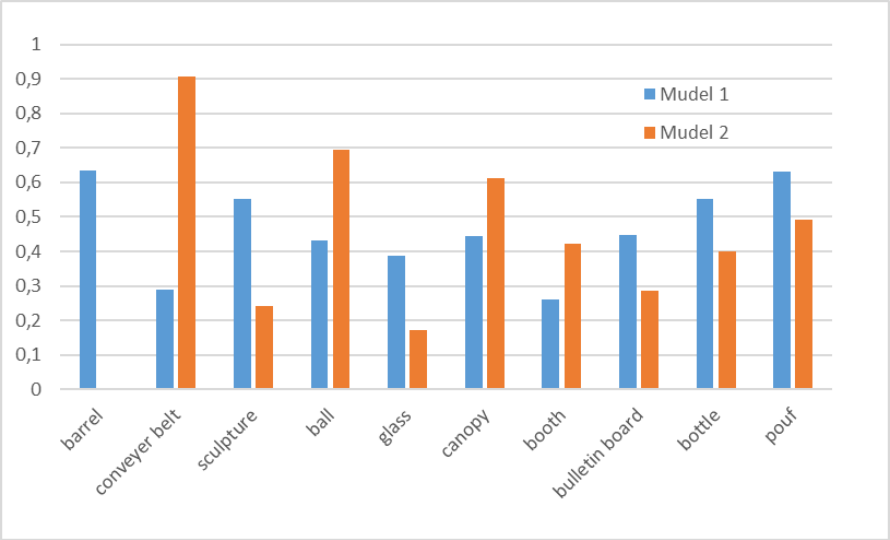
Joonis 9. Kahe mudeli täpsuste võrdlus klasside lõikes.

on Mudel 1 kujutatud sinisega ning Mudel 2 punasega. Mudelid on treenitud 151 kategooriaga, seega on joonisel välja toodud suurimate erinevustega kategooriad. Jooniselt

on näha, et mõne kategooria puhul on täpsus kasvanud, mõne puhul aga hoopis kahanenud. Täpsuse suur muutus just nende kategooriate puhul näitab, et treeninghulgas on olnud nende piltide arv väike ning piltide lisamisel on muutus märgatav. Positiivsete muutuse puhul võime järeldada, et lisatud pildid on olnud piisavalt reaalilähedased, et mudelit parandada. Suuremate kategooriate puhul, nagu sein, taevas, ehitis ja puu, on märgata täpsuse 1%-2% tõusu. Suurematekskategooriateks loeme need, mida esineb treeninghulgas rohkem kui 6000 pildil.

Saagise abil mõõdetakse kui suure osa ennustatavast kategooriast mudel pildilt tuvastas.

$$Saagis = \frac{TP}{TP + FN}$$



Joonis 10. Kahe mudeli saagise võrdlus klasside lõikes.

Joonisel 10 on toodud suurimate saagise erinevustega kategooriad. Sarnaselt täpsusele on tegemist küllaltki kitsaste kategooriatega, mida treeninghulga piltidel sisaldub vähe, mistõttu on kas positiivne või negatiivne muutus olnud suurem.

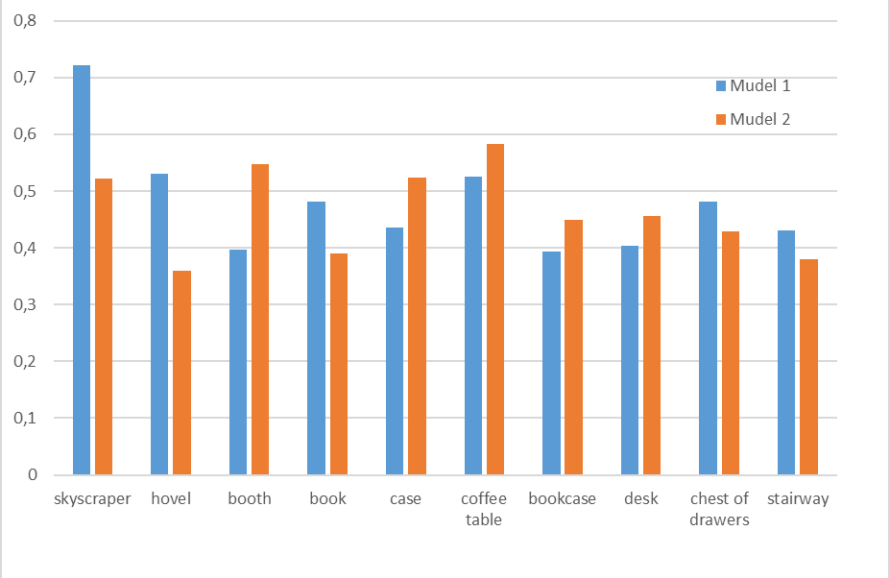
Suurematel kategooriatel on saagise väärtus hoopis piltide lisamisega kuni 1% võrra kahanenud. See võib viidata sellele, et neile kategooriatele ennustatakse suuremat seg-

menti kui varem, mis tõstab küll täpsust, kuid selle arvelt kahaneb saagis, sest piirkonda kuulub siis ka rohkem valesid väärtuseid.

F1-skoor arvestab nii täpsust kui saagist. See aitab tasakaalustada väärtuseid juhtudel kui ennustatud piirkond on väike, kuid tegelik kategooria katab suuremat ala, mistõttu tuleb suur täpsus, kuid väike saagis. Sarnane olukord võib tekkida ka vastupidiselt, kui ennustatud piirkond on suur, kuid tegelik kategooria hõlmab väikest ala. F1-skoori saame arvutada valemiga:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Joonisel 11 on välja toodud F1-skoori suurimate muutustega kategooriad. Näeme, et suurimatest muutustest on pooltel kategooriatel F1-skoor kasvanud ning pooltel kahanenud. Sarnane arvutus kõikide kategooriate lõikes näitab, et 41% kategooriatest on F1-skoor kahanenud ning 59% tõusnud. Suuremaid kategooriaid võrreldes selgub, et F1-skoor tõuseb kuni 0.5%.



Joonis 11. Kahe mudeli F1-skoori võrdlus klasside lõikes.

Tabelis 4 on keskmised väärtused mõlema mudeli kohta. Keskmised väärtused ei muutu oluliselt, kuid väike kasv on märgata täpsuses ning samas suurusjärgus langus

Tabel 4. Keskmised väärtused.

	Täpsus	Saagis	F1-Skoor
Mudel 1	0.44681	0.48702	0.23302
Mudel 2	0.44142	0.49121	0.23249

saagises. Nii täpsus kui ka saagis mõjutavad F1-skoori, kuid kuna üks langeb ning teine tõuseb, siis jääb F1-skoor muutumatuna. Kõigi kategooriate kohta saadud tulemused on leitavad lisast 4.

$$SQ = \frac{\sum_{(p,g)\in TP} IoU_{(p,g)}}{|TP|}$$

3.3 Analüüs

Kahe mudeli võrdlusest ilmnes, et genereeritud piltide lisamine andmestikule keskmiselt täpsust, saagist ega F1-skoori oluliselt ei paranda. Antud töös treenitud mudelid ei jää alla varasemalt ADE20K andmestikul treenitud DeepLabv3+ mudeliga [36]

Visuaalsel vaatlusel näeme, et kategooriatel, mis on piisavalt üldised, näiteks taevas ja põld, täpsus paraneb. Semantiline difusioonimudel suudab neis kategooriates genereerida realistlike pilte, mis tõstab DeepLabv3+ mudeli täpsust nendes kategooriates. Tehtud arvutused kinnitavad, et täpsus suuremates kategooriates tõusis 1% - 2%. Võttes arvesse kõiki kategooriaid, siis keskmine täpsus langes 0.005 võrra. Keskmise täpsuse arvutamisel pole arvestatud kategooria suurust või esinemiste hulka treening- või testandmetes.

Keskmine saagis tõusis 0.004 võrra. Kategooriaid eraldi vaadates oli näha, et 59% saagise väärtus kasvas. Samas suurematel kategooriatel on märgata hoopis saagise langust.

F1-skoor tasakaalustab mõlemat väärtust, seega kuna keskmise mõistes täpsus langes

ning saagis tõusis, siis on F1-skoor mõlema mudeli korral sama. Tulemuste põhjal saab väita, et lisatud pildid ei olnud piisavalt realistlikud, et segmenteerimismudeli tulemusi parandada. Kategooriaid eraldi hinnates saab välja tuua, et kolme suurema puhul on F1-skoor kas jäänud samaks või tõusnud. Tabel 11 näitab, et suurimad muutused on toimunud kategooriates, mille hulk treeninghulgas on olnud väike ning genereeritud piltide lisamine mõjutab segmenteerimise tulemusi rohkem kui mahult suuremate kategooriate puhul.

Kokkuvõttes saab väita, et üldisemate kategooriate puhul aitavad genereeritud pildid tulemusi tõsta, samas kui spetsiifilisemate kategooriate puhul ei ole genereeritavad pildid samaväärsed fotodega ning pigem langetavad mudeli segmenteerimiskvaliteeti.

4 Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli luua semantilise difusioonimudeliga pildiandmestik ning hinnata selle tulemusi DeepLabv3+ mudelil. Piltide käsitsi tegemise asemel nende genereerimine aitab kokku hoida aega pildiandmestiku loomisel.

Semantiline difusioonimudel võimaldab genereerida segmenteeritud pildi põhjal realistliku vaste, mis oleks suuteline asendama fotot. Piltide genereerimiseks kasutati ADE20K pildiandmestikul eeltreenitud mudelit [28]. Bakalaureusetöö käigus genereeriti 1640 pilti, mis lisati ADE20K treeninghulgale. Genereeritud pildid moodustavad uuest treeninghulgast 7.5%. Seejärel treeniti DeepLabv3+ mudel ning võrreldi tulemusi puhtalt ADE20K treeninghulgal treenitud mudeliga.

DeepLabv3+ mudelite testimiseks kasutati ADE20K valideerimishulgast eraldatud testhulka, mille suurus oli 1000 pilti. Tulemustest selgus, et lisatud pildid ei paranda keskmiselt segmentatsiooni kvaliteeti. Mahult väiksematel kategooriatel, mida lisatud pildid rohkem mõjutavad võib täpsus hoopis langeda, kuna segmenteeritud pildid ei ole piisavalt realistlikud. Mahult suuremate kategooriate puhul on tulemused pigem paranenud. Genereeritud piltidel on suuremate kategooriate segmendid realistlikumad, kuna nende hulk on olnud suur ka semantilise difusioonimudeli treenimisel.

Saadud tulemused näitavad, et vaid genereeritud piltidega loodud andmestik ei ole piisav, et treenida segmentatsioonimudelit. Samas võib piltide lisamine aidata kaasa tausta tuvastamist, nagu taevas ja maapind.

Tulemusi saaks parandada kui treenida semantiline difusioonimudel pildiandmestikul, kus soovitud kategooriaga piltide hulk on suurem. Semantilise difusioonimudeli poolt genereeritud pildid on siis ka realistlikumad nendes kategooriate ja piltidega treenitud segmenteerimismudel on täpsem.

Viidatud kirjandus

[1] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. 2017.

[2] Asoke Nath. Image denoising algorithms: A comparative study of different filtration approaches used in image restoration. In *2013 International Conference on Communication Systems and Network Technologies*, pages 157–163, 2013.

[3] Ben Lambert. *An Introduction to Markov Chain Monte Carlo*. Oxford University Press, 2018.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[5] RichardŠ Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[7] Fei Liu, Zihao Lu, and Xianke Lin. Vision-based environmental perception for autonomous driving, 2022.

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

[9] Weilun Wang, Jianmin Bao, Wengang Zhou, Dongdong Chen, Dong Chen, Lu Yuan, and Houqiang Li. Semantic image synthesis via diffusion models, 2022.

[10] Fangneng Zhan, Yingchen Yu, Rongliang Wu, Jiahui Zhang, and Shijian Lu. Multimodal image synthesis and editing: A survey. *ArXiv*, abs/2112.13592, 2021.

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018.

[12] Stuart J Russell and Peter Norvig. Artificial intelligence a modern approach third edition, 2010.

[13] Aziz Alotaibi. Deep generative adversarial networks for image-to-image translation: A review. *Symmetry*, 12:1705, 10 2020.

[14] Calvin Luo. Understanding diffusion models: A unified perspective, 2022.

[15] Adaloglou Nikolaos Karagiannakos, Sergios. Diffusion models: toward state-of-the-art image generation. *https://theaisummer.com/*, 2022. Vaadatud 20.02.2023.

[16] Edan Meyer. Stable diffusion - what, why, how? https://www.youtube.com/watch?v=ltLNYA3lWAQab_channel=EdanMeyer, 2022. Vaadatud 14.03.2023.

[17] Ryan O'Connor. Introduction to diffusion models for machine learning. May 2022.

[18] Mrinal Walia. Semantic segmentation vs. instance segmentation: Explained. https://blog.roboflow.com/difference-semantic-segmentation-instance-segmentation/, 2023. Vaadatud 28.03.2023.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[20] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021.

[21] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.

[22] S.Š. Sreeja and B. Ravindran. Generative adversarial networks (gans): An overview of theoretical contributions and applications in computer vision. *Journal of Ambient Intelligence and Humanized Computing*, 10(10):4301–4322, 2019.

[23] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018.

[25] Google. Generative adversarial networks (gans) - generator, 2023. "Vaadatud 07.05.2023".

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[27] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset, 2018.

[28] Google Drive. net_sdm_ade20k.zip, 2022. Vaadatud 30.01.2023.

[29] Hongkun Yu, Chen Chen, Xianzhi Du, Yeqing Li, Abdullah Rashwan, Le Hou, Pengchong Jin, Fan Yang, Frederick Liu, Jaeyoun Kim, and Jing Li. TensorFlow Model Garden. https://github.com/tensorflow/models, 2020.

[30] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017.

[31] Jie Shi, Qian Guo, and Xinyu Zhao. A comparative study of semantic segmentation networks for autonomous driving. *IEEE Access*, 9:95399–95410, 2021.

[32] Lakhveer Gondara, Ying Ma, and Vinh Ly. Evaluation of deeplabv3+ for land cover classification in unmanned aerial vehicle imagery. *Drones*, 5(2):1–18, 2021.

[33] Jaewan Choi. Deeplabv3plus-pytorch-ade20k. https://github.com/jwchoi384/DeepLabv3Plus-Pytorch-ade20k, 2020. Vaadatud 09.05.2023.

[34] Leslie N. Smith. Cyclical learning rates for training neural networks, 2017.

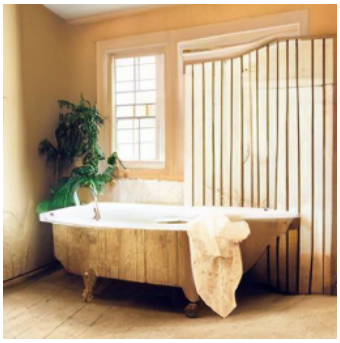
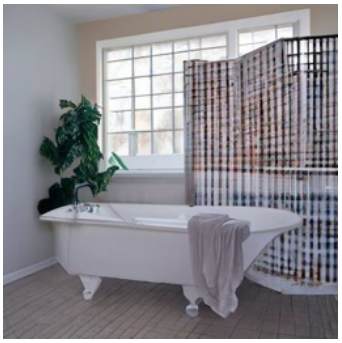
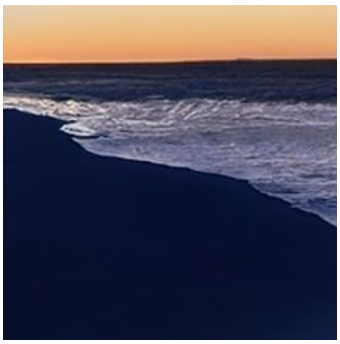
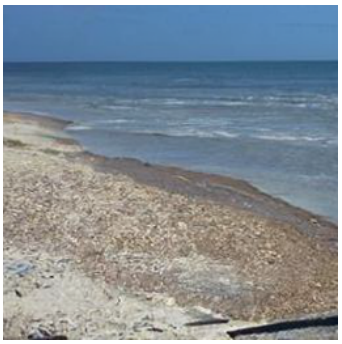
[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[36] Rawal Khirodkar, Brandon Smith, Siddhartha Chandra, Amit Agrawal, and Antonio Criminisi. Sequential ensembling for semantic segmentation, 2022.

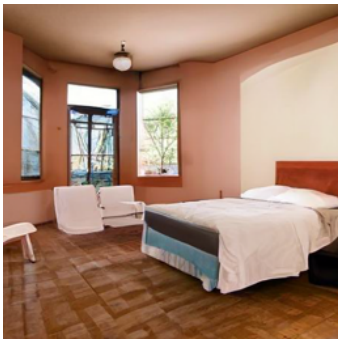
Lisad

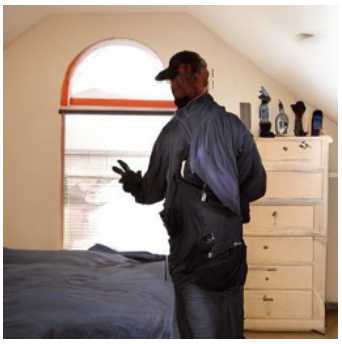
LISA 1. Näiteid semantilise difusioonimudeliga genereeritud piltidest

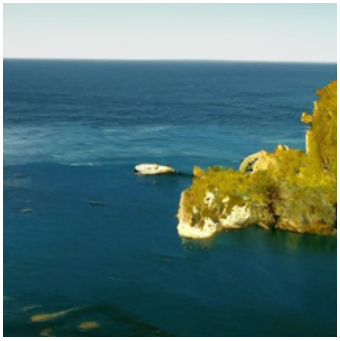
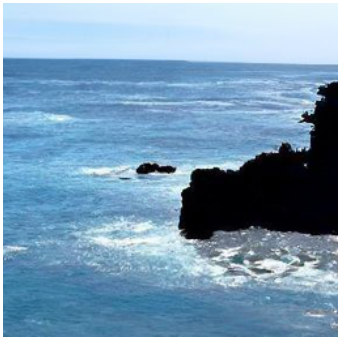
Tabelis 5 on toodud 10 näidist genereeritud piltidest. Kõiki genereeritud pilte saab leida töö GitHubi projektist.

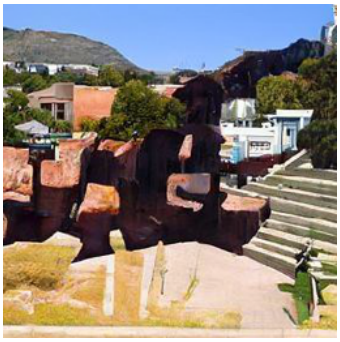
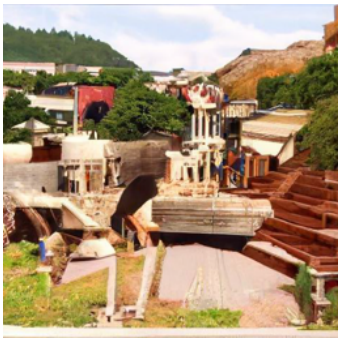
Reaalne pilt	**Näidis 1**	**Näidis 2**
		
		





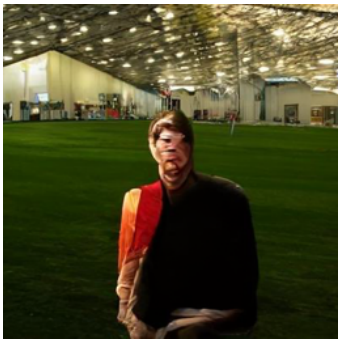











LISA 2. Täpsus, saagis ning F1-skoor kõigi kategooriate lõikes

Mudeli, mis on treenitud vaid ADE20K treeninghulgaga, väärtused asuvad tulpades täpsus 1, saagis 1 ning F1-skoor 1. Teise mudeli, mille treenimisel kasutati treeninghulka, kuhu lisati genereeritud pilte, tulemused on vastavalt tulpades täpsus 2, saagis 2 ning F1-skoor 2.

Kategooria	**Täpsus 1**	**Täpsus 2**	**Saagis 1**	**Saagis 2**	**F1-skoor 1**	**F1-skoor 2**
wall	0.787	0.799	0.761	0.75	0.774	0.774
building	0.859	0.877	0.833	0.824	0.846	0.849
sky	0.937	0.96	0.944	0.934	0.941	0.946
floor	0.742	0.762	0.824	0.808	0.781	0.784
tree	0.83	0.832	0.782	0.774	0.805	0.802
ceiling	0.859	0.87	0.793	0.793	0.825	0.83
road	0.784	0.788	0.86	0.853	0.82	0.819
bed	0.825	0.814	0.737	0.752	0.779	0.782
windowpane	0.656	0.657	0.635	0.594	0.646	0.624
grass	0.696	0.698	0.764	0.804	0.729	0.747
cabinet	0.567	0.603	0.637	0.612	0.6	0.608
sidewalk	0.65	0.614	0.639	0.659	0.645	0.636
person	0.816	0.828	0.735	0.746	0.773	0.785
earth	0.379	0.377	0.374	0.377	0.376	0.377
door	0.367	0.368	0.464	0.504	0.41	0.426
table	0.519	0.509	0.519	0.539	0.519	0.524
mountain	0.718	0.71	0.624	0.62	0.668	0.662
plant	0.561	0.505	0.702	0.745	0.623	0.602
curtain	0.749	0.732	0.697	0.69	0.722	0.711

chair	0.532	0.539	0.539	0.546	0.535	0.543
car	0.791	0.787	0.776	0.783	0.784	0.785
water	0.653	0.645	0.562	0.635	0.604	0.64
painting	0.705	0.716	0.658	0.664	0.681	0.689
sofa	0.729	0.724	0.611	0.651	0.665	0.686
shelf	0.465	0.499	0.44	0.484	0.452	0.492
house	0.603	0.486	0.407	0.527	0.486	0.506
sea	0.816	0.803	0.616	0.707	0.702	0.752
mirror	0.51	0.476	0.622	0.658	0.56	0.552
rug	0.397	0.411	0.707	0.742	0.508	0.529
field	0.401	0.398	0.35	0.426	0.374	0.411
armchair	0.308	0.344	0.471	0.505	0.372	0.409
seat	0.417	0.47	0.42	0.46	0.419	0.465
fence	0.461	0.453	0.493	0.486	0.476	0.469
desk	0.539	0.537	0.322	0.397	0.403	0.457
rock	0.738	0.616	0.504	0.526	0.599	0.568
wardrobe	0.524	0.49	0.368	0.396	0.432	0.438
lamp	0.637	0.653	0.625	0.63	0.631	0.641
bathtub	0.871	0.826	0.656	0.771	0.748	0.797
railing	0.298	0.359	0.506	0.514	0.375	0.423
cushion	0.457	0.501	0.556	0.596	0.502	0.544
base	0.281	0.263	0.265	0.34	0.273	0.297
box	0.253	0.291	0.321	0.34	0.283	0.314
column	0.492	0.491	0.584	0.523	0.534	0.506
signboard	0.325	0.335	0.458	0.413	0.38	0.37
chest of drawers	0.47	0.413	0.495	0.447	0.482	0.429
counter	0.329	0.359	0.695	0.664	0.446	0.466

sand	0.423	0.423	0.306	0.318	0.355	0.363
sink	0.519	0.564	0.591	0.643	0.553	0.601
skyscraper	0.74	0.469	0.704	0.591	0.722	0.523
fireplace	0.707	0.687	0.569	0.608	0.63	0.645
refrigerator	0.647	0.593	0.705	0.737	0.675	0.657
grandstand	0.052	0.062	0.012	0.021	0.019	0.031
path	0.501	0.475	0.295	0.275	0.371	0.348
stairs	0.349	0.398	0.251	0.236	0.292	0.296
runway	0.916	0.864	0.547	0.504	0.685	0.637
case	0.728	0.71	0.311	0.416	0.436	0.524
pool table	0.96	0.933	0.787	0.806	0.865	0.865
pillow	0.503	0.513	0.531	0.582	0.517	0.545
screen door	0.518	0.448	0.456	0.536	0.485	0.488
stairway	0.399	0.354	0.466	0.409	0.43	0.379
river	0.137	0.142	0.152	0.095	0.144	0.114
bridge	0.525	0.477	0.346	0.314	0.417	0.379
bookcase	0.578	0.622	0.298	0.352	0.393	0.45
blind	0.202	0.226	0.547	0.575	0.295	0.325
coffee table	0.463	0.542	0.606	0.63	0.525	0.582
toilet	0.853	0.851	0.685	0.758	0.76	0.802
flower	0.284	0.317	0.476	0.441	0.356	0.369
book	0.455	0.378	0.51	0.404	0.481	0.39
hill	0.097	0.108	0.267	0.183	0.142	0.136
bench	0.394	0.407	0.675	0.545	0.497	0.466
countertop	0.39	0.402	0.43	0.501	0.409	0.446
stove	0.732	0.674	0.581	0.683	0.648	0.678
palm	0.534	0.563	0.522	0.517	0.528	0.539

kitchen island	0.432	0.433	0.201	0.241	0.274	0.31
computer	0.878	0.874	0.626	0.581	0.731	0.698
swivel chair	0.555	0.554	0.592	0.57	0.572	0.562
boat	0.842	0.574	0.71	0.725	0.77	0.64
bar	0.137	0.139	0.462	0.486	0.211	0.217
arcade machine	0.511	0.413	0.572	0.626	0.54	0.497
hovel	0.526	0.3	0.535	0.448	0.531	0.359
bus	0.805	0.834	0.648	0.67	0.718	0.743
towel	0.529	0.575	0.649	0.715	0.582	0.637
light	0.328	0.375	0.677	0.694	0.442	0.487
truck	0.381	0.427	0.43	0.365	0.404	0.394
tower	0.448	0.462	0.764	0.776	0.565	0.579
chandelier	0.656	0.707	0.714	0.698	0.684	0.702
awning	0.216	0.264	0.417	0.48	0.285	0.34
streetlight	0.075	0.141	0.332	0.343	0.122	0.2
booth	0.846	0.775	0.259	0.424	0.397	0.548
television receiver	0.532	0.462	0.744	0.755	0.62	0.573
airplane	0.531	0.535	0.432	0.402	0.476	0.459
dirt track	0.009	0.2	0.007	0.074	0.008	0.108
apparel	0.437	0.354	0.333	0.374	0.378	0.364
pole	0.09	0.135	0.263	0.274	0.134	0.181
land	0	0.01	0	0.006	nan	0.007
bannister	0.055	0.053	0.098	0.089	0.071	0.066
escalator	0.558	0.535	0.473	0.56	0.512	0.547
ottoman	0.36	0.381	0.631	0.493	0.458	0.43
bottle	0.163	0.102	0.551	0.399	0.251	0.162
buffet	0.277	0.258	0.511	0.563	0.359	0.354

poster	0.063	0.09	0.343	0.365	0.107	0.145
stage	0.031	0.216	0.012	0.07	0.017	0.105
van	0.349	0.334	0.575	0.474	0.434	0.392
ship	0.927	0.846	0.265	0.194	0.412	0.316
fountain	0.015	0.006	0.087	0.068	0.026	0.011
conveyer belt	0.561	0.255	0.288	0.907	0.381	0.398
canopy	0.086	0.16	0.446	0.613	0.145	0.254
washer	0.71	0.684	0.625	0.681	0.665	0.683
plaything	0.301	0.211	0.511	0.386	0.379	0.272
swimming pool	0.572	0.814	0.399	0.477	0.47	0.602
stool	0.058	0.089	0.368	0.301	0.1	0.137
barrel	0.324	0	0.634	0	0.429	nan
basket	0.198	0.242	0.633	0.511	0.302	0.328
waterfall	0.853	0.875	0.397	0.515	0.541	0.648
tent	0.996	0.994	0.674	0.77	0.804	0.868
bag	0.073	0.101	0.446	0.52	0.125	0.169
minibike	0.357	0.537	0.697	0.762	0.472	0.63
cradle	0.803	0.765	0.519	0.51	0.63	0.612
oven	0.081	0.214	0.071	0.17	0.075	0.19
ball	0.323	0.129	0.433	0.696	0.37	0.218
food	0.207	0.154	0.168	0.201	0.185	0.174
step	0.067	0.06	0.252	0.176	0.105	0.09
tank	0.889	0.871	0.351	0.324	0.503	0.473
trade name	0.201	0.165	0.473	0.477	0.282	0.245
microwave	0.231	0.247	0.752	0.733	0.354	0.37
pot	0.197	0.183	0.492	0.457	0.281	0.261
animal	0.256	0.217	0.745	0.664	0.381	0.327

bicycle	0.502	0.487	0.3	0.388	0.375	0.432
lake	0	0	0	0	nan	nan
dishwasher	0.485	0.541	0.551	0.652	0.516	0.591
screen	0.614	0.751	0.513	0.516	0.559	0.612
blanket	0.003	0.01	0.08	0.099	0.005	0.017
sculpture	0.365	0.209	0.551	0.24	0.439	0.224
hood	0.358	0.463	0.473	0.483	0.408	0.473
sconce	0.162	0.21	0.509	0.609	0.246	0.312
vase	0.186	0.219	0.202	0.223	0.194	0.221
traffic light	0.122	0.247	0.487	0.487	0.195	0.328
tray	0.01	0.011	0.017	0.034	0.013	0.017
ashcan	0.21	0.299	0.565	0.494	0.306	0.372
fan	0.431	0.476	0.644	0.655	0.517	0.552
pier	0.065	0.054	0.106	0.082	0.08	0.065
crt screen	0.094	0.026	0.169	0.083	0.121	0.04
plate	0.219	0.065	0.274	0.164	0.243	0.093
monitor	0.298	0.266	0.157	0.075	0.206	0.117
bulletin board	0.266	0.196	0.447	0.285	0.333	0.232
shower	0	0	0	0	nan	nan
radiator	0.407	0.486	0.925	0.902	0.566	0.631
glass	0.061	0.023	0.387	0.171	0.105	0.041
clock	0.211	0.23	0.643	0.506	0.317	0.316
flag	0.15	0.231	0.837	0.846	0.254	0.362

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Triin Schaffrik**,

(autori nimi)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

   Pildiandmestiku loomine semantilise difusioonimudeliga,

   mille juhendaja on Joosep Kivastik,

   reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Triin Schaffrik

09.05.2023