

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Kerly Klee 164085IABB

**KARIKATUURIDE LOOMINE
NÄRVIVÕRGU STILIÜLEKANDE JA
GENEREERIVATE VASTANDVÕRKUTEGA**

Bakalaureusetöö

Juhendaja: Martin Rebane
MSc

Tallinn 2019

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Kerly Klee

20.05.2019

Annotatsioon

Lõputöö eesmärgiks on uurida karikatuuride ja masinõppe algoritmide tausta ja masinõppe meetodite põhjal luua mudelid karikatuuride loomiseks ning analüüsida mudelite võimekust karikatuure luua. Mudelite loomiseks on töös kasutatud kahte erinevat masinõppe meetodid.

Karikatuure genereerivate mudelid loodi kahel erineval viisil kasutades genereerivaid vastand võrke ja närvivõrgu stiili ülekannet. Mudelite tulemusi võrreldakse ja viiakse läbi katseid nende tulemuste parandamiseks.

Mudelite loomisel on kasutatud baaskeelena Pythonit. Lisaks on kasutuses masinõpet toetavad Pythoni laiendused TensorFlow ja Keras. Mudel närvivõrgu stiili ülekandega osutus antud töös karikatuuride loomise edukamaks kui genereeriva vastandvõrgu mudel.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 37 leheküljel, 4 peatükki, 32 joonist.

Abstract

Creating Caricatures with Neural Style Transfer and Generative Adversarial Networks

The goal of the thesis is to study the background of caricatures and machine learning algorithms and using machine learning method create models that could create caricatures and analyse models results. Two different machine learning methods have been used to create the models.

The caricature generating models were created in two different ways using generative adversarial networks and neural style transfer. The results of the models are compared and attempts are made to improve the results.

Python has been used as the base language for the models. In addition, Python extensions that support machine training TensorFlow and Keras are used. Experiments showed that in given paper neural style transfer model has better results than generative adversarial network model.

The thesis is in Estonian and contains 37 pages of text, 4 chapters, 32 figures.

Lühendite ja mõistete sõnastik

AI	<i>Artificial intelligence</i> , tehisintelligent – masinate simulatsiooniprotsess mõelda nagu inimene.
ANN	<i>Artificial Neural Network</i> ehk tehisnärvivõrk – arvutitöötlus süsteemid, mis on inspireeritud bioloogilisest närvisüsteemist.
CNN	<i>Convolutional neural network</i> , konvolutsiooniline närvivõrk – masinõppemeetod, mida rakendatakse siis kui sisend andmeteks on pildid. Disainitud inimese aju järgi, et see töötleks informatsiooni samamoodi.
ReLU	<i>Rectified linear function</i> ehk mittenegatiivne lineaarfunktsioon – konvolutsioonilise kihi lõpus rakendatav funktsioon, mis annab väljundiks 0 kui tulemus on väiksem kui 0. Vastasel juhul on funktsiooni väljundiks sisend.
Närvivõrgu stiili ülekanne	<i>Neural Style transfer</i> , Optimeerimistehnika, mille abil võetakse sisendiks kolm pilti ja väljundiks on sama stiiliga pilt
Tunnuste kaart	<i>Feature map</i> , konvolutsioonilise närvivõrgu kihi väljund, mis saadakse filtri kasutamisel pildil ja arvutades vastava punkti tulemuse.
GAN	<i>Generative Adversarial Networks</i> , genereeriv vastandvõrgustik - mille idee seisneb kahe närvivõrgu genereeriva ja detektiivse närvivõrgu vastandamises.
Andmevoo graafid	Dataflow graphs , Arvutusgraafid, mida kasutatakse Tensorflow teekides tänu millele saavad erinevad seadmed masinõppe ahendustuumasi kasutada.
Ahendustuum	<i>Kernel</i> - Konvolutsioonilistes närvivõrkudes kasutatav pikslite rühm, mida kasutatakse üldiselt ahenduste tegemisteks.
Gramm maatriks	<i>Gram matrix</i> ehk Gram'i maatriks on närvivõrgu stiili ülekande juures kasutatavad maatriksid, mis võimaldavad võrrelda erinevate piltide stiili erinevusi.
Vabaliikmed	<i>Biases</i> ehk vabaliikmed. Teisiti öeldes ka nihe. Kasutatakse närvivõrkude treenimisel, neid kohandatakse vastavalt kao gradientidele.
Kaal	<i>Weight</i> ehk kaal, esindab konvolutsioonilise närvivõrgu filtrite mõõtmeid.

Gradient	Närvivõrkudes kao arvutamisel esindab kui sügav on kallak kuna närvivõrk on ruumiline ja omab maksimum ja miinimum kohti.
Vastandkonvolutsioon	<i>Deconvolution</i> ehk vastandkonvolutsiooniks nimetatakse andmete eraldusvõime parandamist matemaatilise algoritmi abil, mis on mõeldud esialgse teabe resolutsiooni paremaks muutmiseks.
VGG16	<i>Very Deep Convolutional Networks for Large-Scale Image Recognition</i> ehk sügav konvolutsioonilised närvivõrgud suure mõõtmeliste piltide tundmiseks. Eeltreenitud konvolutsiooniline närvivõrk, mis koosneb kuueteistkümnest konvolutsioonilisest kihist.
VGG19	<i>Very Deep Convolutional Networks for Large-Scale Image Recognition</i> ehk sügav konvolutsioonilised närvivõrgud suure mõõtmeliste piltide tundmiseks. Eeltreenitud konvolutsiooniline närvivõrk, mis koosneb üheksateistkümnest konvolutsioonilisest kihist.

Sisukord

1 Sissejuhatus	11
2 Taust.....	13
2.1 Karikatuur ja portree	13
2.2 Masinõpe.....	15
2.3 Tehisnärvivõrgud	16
2.4 Konvolutsioonilised närvivõrgud.....	17
2.4.1 Konvolutsiooniline kiht	18
2.4.2 Ahenduskiht ja täielikult ühendatud kiht	19
2.5 Närvivõrgu stiiliülekanne	20
2.5.1 Närvivõrgu stiiliülekanne teooria	20
2.6 Genereerivad vastandvõrgud	22
2.6.1 Genereerivate vastandvõrkude teooria	23
2.6.2 Genereerivate vastandvõrkude treenimine.....	25
2.7 Tehnoloogia	27
2.7.1 TensorFlow ja Keras.....	28
2.7.2 Google graafikaprotsessor	28
3 Mudelid	29
3.1 Andmestikud.....	29
3.2 Närvivõrgu stiiliülekanne mudel.....	30
3.2.1 VGG 16 mudel	31
3.2.2 VGG 19 mudel	33
3.3 Genereeriva vastandvõrgu mudel.....	36
3.3.1 Optimeerimismeetodi valimine	37
3.3.2 Mudeli tulemused	42
3.4 Järeldused	44
4 Kokkuvõte	47
Kasutatud kirjandus	48
Lisa 1- Närvivõrgu stiiliülekanne andmestik	53

Lisa 2- Genereerivate vastandvõrkude andmestik 54

Jooniste loetelu

Joonis 1. Arnold Schwarzeneggeri karikatuur [10]	14
Joonis 2. Lihtne kolme kihiga närvivõrk [17].	16
Joonis 3. Konvolutsioonilise närvivõrgu mudel [18].....	18
Joonis 4. Konvolutsioonilise kihi filtrite efekt [23], autori markeeringuga.	19
Joonis 5. Ahenduskihi lihtne näide [19].....	19
Joonis 6. Piltide esitlused konvolutsioonilises närvivõrgus [27].....	21
Joonis 7. GAN mudeli joonis [32].	23
Joonis 8. Näide vastand konvolutsiooni ahendustuuma idee [33].	24
Joonis 9. Generaatori mudeli konvolutsioonilise närvivõrgu joonis [35].	24
Joonis 10. Kao miinimumi visuaalne näide [37] , autori markeeringuga.	26
Joonis 11. Õpisammu olulisus [37], autori markeeringuga.....	26
Joonis 12. Karikatuur Pintresti kogumist [44].....	30
Joonis 13. Karikatuur Caroly Huobolaineni kogumist [21].	30
Joonis 14. Pildi esitlus (autori joonis).	31
Joonis 15. Närvivõrgu VGG16 mudeli kadu (autori joonis).	32
Joonis 16. Tulemuste võrdlus iteratsiooni arvu põhjal (autori joonis).	32
Joonis 17. Mudeli tulemus, konteksti pilt [21], stiilipilt [48], autori markeeringuga.	33
Joonis 18. Iteratsioonide võrdlus (autori joonis), konteksti pilt [50] , stiilipilt [48].....	33
Joonis 19. 512 x 512 suuruse pildi tulemuse kvaliteet (autori joonis), konteksti pilt [51], stiilipilt [52].....	34
Joonis 20. 1000 x 1440 tulemus pildi kvaliteet (autori joonis), konteksti pilt [53], stiilipilt [54].....	35
Joonis 21. Tulemus 500 iteratsiooniga 680x780 suuruses, (autori joonis) , konteksti pilt [55], stiilipilt [56].....	36
Joonis 22. Gradientlaskumine sammuga 0,001 (autori joonis).	37
Joonis 23. Gradientlaskumise optimeerimisega tulemused 600 epoch korral (autori joonis).....	38
Joonis 24. Adam võrdlus 0.001 õpisammu ja 0.0001 õpisammu vahel (autori joonis). .	39
Joonis 25. Adam tulemused õpisamm 0,001 ja 0,0001 korral (autori joonis).	39

Joonis 26. RMSprop võrdlus 0,001 , 0,002 ja 0,0002 vahel (autori joonis).....	40
Joonis 27. 600. genereeritud piltide võrdlus erineva õpisammu vahel.	41
Joonis 28. Genereeritu tulemused 5000 epohhiga 0,0002 õpisammu puhul.	42
Joonis 29. Parimad tulemused õpisammuga 0.00012	43
Joonis 30. Närvivõrgu stiilimudeli VGG16 näite tulemused.	44
Joonis 31. Närvivõrgu stiiliülekanne VGG 19 mudeli näite tulemused	45
Joonis 32. RMSprop õpisammuga 0,0002 epohh 6800.	46

1 Sissejuhatus

Karikatuure on loodud inimeste poole juba sellest ajast peale kui inimesed kunsti looma hakkasid, tänu karikatuuridele on näha, et inimesed on osanud iseenese üle naerda juba algusest peale [1]. Karikatuure on lõbus vaadata ja need lahutavad inimeste meelt. Karikatuur kuulub visuaalse huumori alla. Karikatuuride loomiseks peab mõistma, et karikatuuri teeb naljakaks selle, et karikatuuril kujutatava inimese proportsioonid on viidud tasakaalust välja, rõhutades inimese omapäraseid näojooni või suurstades millegigi, mille poolest inimene tuntud on. Antud töös üritatakse karikatuure kui visuaalset huumorit luua masinõppe meetodite abil

Masinõppe algoritmid annavad masinale võimalus õppida tuginedes suurele hulgatele andmetele kogudele, mis võimaldab masinal ise otsuseid teha ilma, et inimene oleks detailselt asjad ette programmeerinud [2]. Antud töös on andmestikeks pildid karikatuuridest ja portreedest, mille põhjal otsuseid tehes hakkavad masinõppe meetodid karikatuure looma. Masinõppe algoritmid ja sellega seostuvad võimalused on paljude uute tehnoloogiliste lahenduste aluseks. Paljud suured ettevõtted, kellel on palju andmekogumeid näevad seda tulevikuna, sest masinõppe võimaldab luua erinevaid tehisintelligentseid lahendusi, mis suudavad inimeste kõnest soove välja lugeda, pildi või video põhjal otsuseid teha ja kas või ise kliendiga suhelda [3]. Tänapäeval, kus andmeid on suurtes hulkades ja neid on kerge koguda tänu andmebaaside arengule, on mudelid, mis suudavad tänu andmetele otsuseid teha ja ise nendega kasulikku asju korda saata väga tere tulnud.

Antud töös on lähemalt süvenetud kahte viisi, kuidas masinõppe meetodite abil karikatuure luua. Esimeseks neist närvivõrgu stiili ülekannet ja teiseks genereeriva vastand võrk, mõlemad kasutavad oma töös konvolutsioonilisi närvivõrke. Töö eesmärk on välja selgitada mõlema viisi efektiivsus karikatuuride loomisel ning optimeerida mudeleid parema tulemuste tarbeks. Valitud meetodid said valitud kuna nii genereerivad vastand võrgud kui ka närvivõrgu ülekannet on palju erinevaid võimalusi pakkuvad ja hetkel aktuaalsed meetodid.

Töös kasutatavate ingliskeelsete infotehnoloogiliste terminite tõlkimiseks eesti keelde on kasutatud Närvivõrkude ja masinõppe sõnastikku [4]. Lisaks on kasutaud IT

terministandardi sõnastikku veebiversiooni [5]. Terminid, mille kohta eesti keelseid vasteid ei leitud on ümber sõnastatud eesti keelte tuginedes mõistete tähendustele.

2 Taust

Karikatuure on võimalik luua kasutades erinevaid masinõppe meetodeid. Masinõppe meetodeid kasutades karikatuuride loomine on lõbustav ülesanne. Karikatuure on hea vaadata, sest nad on lõbusad ja pakuvad pigem nalja kui midagi muud.

Tänu karikatuuridele mõistame, et inimesed on enese ja teiste üle nalja heitnud juba ammusest ajast. Nii palju kui ka iidset inimesed erinesid meist erinevate omaduste poolest, üks asi on kindel, nad kohe kindlasti naersid teineteise üle ilmselt samadel põhjustel nagu meie, seda on näha ajaloolisest kunstist [6].

Erinevalt inimese näo genereerimisest ei pea karikatuuride puhul ka kõik sada protsenti täpne olema, et oleks aru saada, et tegu on karikatuuriga.

Karikatuuride genereerimisega on varasemalt tegelenud ka näiteks Microsofti teadusinimesed [7]. Nende eesmärgiks oli luua portreest karikatuur ja nad kasutasid selleks andmestikku mis sisaldas 8000 karikatuuri ja 1 miljonit portreed. Antud töös kasutatakse samuti andmestikeks karikatuuride ja portreede kogumeid.

2.1 Karikatuur ja portree

Portree ja karikatuur kujutavad mõlemad inimest. Erinevus seisneb selles, et kui portree puhul on kõige tähtsam täpsus ja tõepärasus, siis karikatuuri puhul on kõige tähtsam, et pildil oleks kujutatud mingit tunnust ebanormaalselt.

Karikatuuride väärtus on ajaloolastele juba ammusest ajast teada, kuna just tänu karikatuuridele on võimalik näha kuidas ühiskond suhtus sellel ajal võimul olnud inimestesse ja sündmustesse [8]. Kuna karikatuure joonistatakse koomilisest vaatenurgast, rõhutatakse sellel asju, mis inimestele on ära tuntavad ja millega inimesed võiksid samastuda.

Karikatuuriks peetakse inimese näo, keha või tegevuse kujutamist nii, et selle proportsioonid on paigast ära. Tavaliselt valitakse isikust, kellest karikatuuri soovitakse teha kindel välimuse või iseloomu joon ja rõhutatakse seda liialduste abil. Tihti kasutatakse ka viiteid loomade käitumismustritele või välimusele, asendades inimese mingi kehaosa looma omaga [9]. Üleüldiselt on levinud arusaam et karikatuurid on joonistused avalikkusele, mis on mõeldud lõbustuseks ning millel kujutatav isik on keegi, kes on ühiskondlikult teatav ja ära tuntav. Karikatuuril on olemas kindel isiksuse omadus, mille kaudu tuntakse kujutatud inimene ära.



Joonis 1. Arnold Schwarzeneggeri karikatuur [10]

Kuigi joonisel (joonis 1) ei ole Arnold Schwarzeneggerit kujutatud tema päris elu proportsioonides, on siiski aru saada, et tegemist on just nimelt temaga. Kuigi on liialdatud tema musklite suuruse ja tema esihammastes oleva vahega. Samuti on võrreldes kehaga ka pea ülemääraselt väikseks tehtud. Karikatuuril kujutatava tunneb vaataja ära tänu kirjeldusele, millega on varem juba kujutatavat seostatud [11]. Tänu sellele ongi Arnold ära tuntav kuna ühiskonnas on ta teada kui näitlejast suur musklimees, omapärase naeratusega.

Portree juures on seevastu kõige tähtsam, et inimene näeks välja täpselt selline nagu ta päriselus välja näeb. Proportsioonid peavad olema paigas ja võimalikult reaalsed. Inimese näojoonte reaalsus on see, mille abil inimene on portreelt ära tuntav. Portree

juures on kõige tähtsam, et inimene oleks portreelt ära tuntav ehk kui meil on inimese portree, peab ta olema nii ära tuttav, et leiame ta rahva seast vaevata üles [12]. Kunagi pidid selle ülesandega hakkama saama kunstnikud, tänapäeval on asi natuke lihtsam, sest on olemas ka fotoportreed, mis saavad jäädvustada inimese reaalsuse pildile palju kiiremini. Foto on portree kui seda vaadates saab kohe ettekujutuse inimese näost ja pildi keskmes on inimese nägu [12]. Tänapäeval on fotograafia piisavalt arenenud, et portreedest puudust ei tule.

2.2 Masinõpe

Masinõppe algoritme kasutatakse mitmetes eluvaldkondades, näiteks ka meditsiinis. Teadmiste kogumine suurest hulgast bioloogilisest andmetest ja nendest järelduste tegemine on võtmeks, et avastata uusi haigusi või ravimeetodeid, eriti kuna inimestel endal võib olla raskusi meditsiiniliste andmetest järelduste tegemisega kuna need on varieeruvad ja mitte struktureeritud [13].

Tänu masinõppe arengule on näiteks ka võimalikuks saanud otse videost video genereerimise võimalused, erinevad keelte tõlkimise mudelid, mis suudavad otse kuuldes teksti tõlkida või kaamerat tekstil hoides juba telefonis tõlgitud teksti näidata, ruumilisi andmeid mõistvad visuaalsed mudelid, näiteks nagu isesõitvates autodes kasutatavad [14]. Masinõppe on palju uusi ja põnevaid lahendusi pakkuv ala.

Sügavõppe on masinõppe liik, mis tegeleb hierarhiliste abstraktsioonide modelleerimisega sisendandmetes mitmete peidetud kihtide abil [15]. Konvolutsioonilised närvivõrgud kuuluvad sügavõppe klassi.

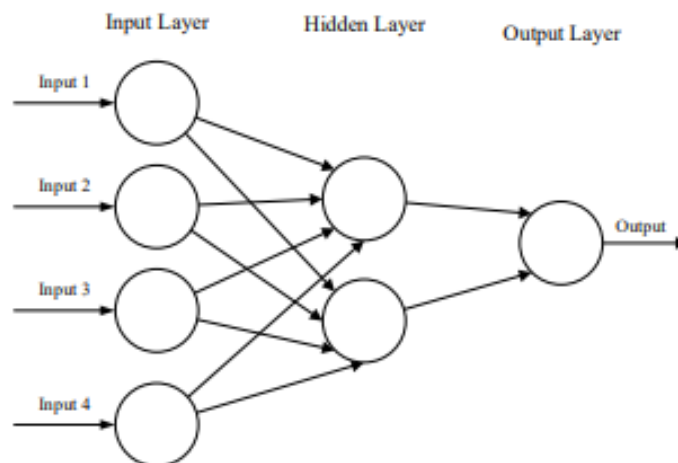
Peale esimeste programmeeritavate arvutite ilmumist on inimesed arutlenud mõtte üle, kas lõpuks arenevad nad isemõtlevateks ja saavad intelligentseteks. Tänapäeval ongi olemas tark intelligent ehk rohkem tuntud kui AI (*artificial intelligent*) [16]. Ise mõtlev tehisintelligent on võimalik tänu masinõppe algoritmidele.

Masinõppe on kiirelt arenev ja populaarsust koguv ala kuna sellega kaasnevad tohutul hulgal uusi võimalusi. Näiteks masinõppe algoritmide abil loodud malesüsteem võitis maailmameistrit Garry Kasparovit juba 1997. aastal [16]. Masinõppe meetodid lahendavad probleemi, mis on enamuses päris maailma tehisintelligentsetel rakendustel. Nimelt on raske panna masinat mõistma kontseptsioone, mida suudab hallata inimese

mõistus. Näiteks, et öösel näeb punane auto pildi pealt välja must või et auto välimus sõltub, millise nurga alt seda vaadata [16]. Tekib probleem, et analüüsi tulemused sõltuvad sellest, kuidas autot esitleda. Sügavõppe algoritmid lahendavad esitlemise probleemi, kuna nende abil on võimalik suur hulk keerulisi andmeid esitleda teisendatult lihtsamal viisil, millest masin suudab aru saada [16]. Sügavõppe meetodid suudavad otsuseid teha lihtsalt suuri andmekogumeid analüüsisid.

2.3 Tehisnärvivõrgud

Tehisnärvivõrgud ehk ANN on arvutitöötlus süsteem, mis töötab samal põhimõttel mis bioloogiline närvisüsteem. Tehisnärvivõrgud koosnevad suurest hulgast ühendatud arvutuslikest neuronitest, mis töötavad koos jaotatud kollektiivis sisendist optimaalse väljundi saamise nimel [17]. Tehisnärvivõrk koosneb sisendkihist, peidetud kihtidest ja väljundikihist (Joonis 2). Sisend laetakse üldiselt mitmemõõtmelise vektorina sisendkihti ning see jaotatakse edasi peidetud kihti. Peidetud kiht teeb seejärel otsuseid põhinedes eelmise kihil ning otsustab, millised muudatused mõjutaks tulemust [17]. Peidetud kihte on tavaliselt mitu, ning neid kihte üksteise järel rakendades saavutatakse sügavõppe.



Joonis 2. Lihtne kolme kihiga närvivõrk [17].

Konvolutsiooniline närvivõrk on analoogne tehisnärvivõrgule, kuna koosneb samuti neuronitest, mis õpivad läbi iseene optimeerimise [17]. Peamine erinevus tuleb sellest, et tehisnärvivõrgu sisendid jagatakse kahte dimensiooni kõrgus ja laius. Konvolutsiooniliste närvivõrkude puhul aga organiseeritakse sisendid kolmedimensiooniliselt kõrgusesse, laiusesse ja sügavusse [17].

2.4 Konvolutsioonilised närvivõrgud

Konvolutsiooniline närvivõrk ehk CNN on masinõppe meetod, mida kasutatakse väga sagedasti mudelites, kus andmeteks on pildid. Konvolutsioonilised närvivõrgud saavad oma nime selle järgi, et nendes kasutatakse lineaarset konvolutsioonilist tehet [16]. Konvolutsioonilised närvivõrgud võimaldavad otsustada, kas pildil kujutatakse otsitavat seda, mida üritatakse tuvastada. Näiteks kui on vaja teada, kas pildi on auto või ei ole. Konvolutsioonilisi närvivõrke treenitakse märgendatud andmestikega [15]. Õpetamaks närvivõrku pildilt autot leidma, antakse konvolutsioonilisele närvivõrgule erinevalt märgistatud pildid. Auto näite puhul oleks siis üks kogum auto pilte märgistatud kui auto ja teine kogus pilte kus pole autod ja märgistatud muu. Neid pilte analüüsid suudab treenitud konvolutsiooniline närvivõrk teha otsuse, kas pildil on auto või ei.

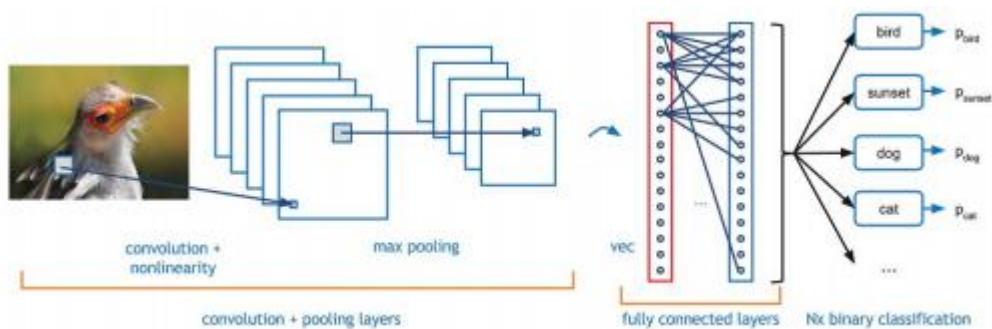
Tänu sellele on konvolutsioonilised närvivõrgud kasutusel ka meditsiinis, sest tänu oskusele pildilt asju ära tunda on võimalik üles leida kohad, kus asuvad kasvaja või muud iseärasused [13]. Suure andmestikuga treenitud konvolutsiooniline närvivõrk võib näiteks magnetresonantstomograafia pildil panna tähele asju, mida inimene oma silmaga ei pruugi näha.

Konvolutsioonilises närvivõrgus kasutatakse tavalise maatrikskorrumise asemel konvolutsioonilist lineaarset valemit (1) [16]. See võimaldab kiiremini kätte saada suurest andmete hulgast ainult olulised andmed ja need edastada järgmisele kihile.

$$s(t) = \int x(a)w(t - a)da \quad (1)$$

Võrreldes range maatrikskorrumisega on konvolutsioon efektiivsem mälu ja statistilise efektiivsuse mõttes [16]. Andmestiku puhul, mis koosneb piltidest, on hea kasutada konvolutsioonilisi närvivõrke, sest võrreldes tehisenärvivõrkudega suudab suuremast hulgast infost olulisema kiiremini ja vähesema mälu vajadusega üles leida.

Konvolutsiooniline närvivõrk koosneb tavaliselt kolmest kihist, konvolutsiooniline kiht (*convolution layer*), ahendus kiht (*pooling layer*) ning täielikult ühendatud kihist (fully connected layers) ning konvolutsioonilisi ja ahenduskihte võib olla ka rohkem kui üks (joonis 3) [17].



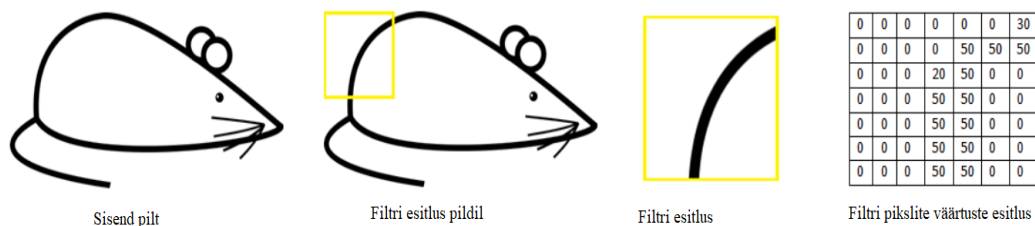
Joonis 3. Konvolutsioonilise närvivõrgu mudel [18].

Antud mudeli põhjal konvolutsiooniline närvivõrk läbib oma kihid ja otsustab saadud tulemuste põhjal, mis loomaga tegemist on.

2.4.1 Konvolutsiooniline kiht

Peamine konvolutsioonilise kihi ülesanne on leida pildilt üles tähtsad omadused, nagu näiteks näo puhul inimese pea piirjooned, silmad, suu, kõrvad, juuksed. Konvolutsiooniline kiht suudab leitud iseloomulikud omadused meelde jätta ja pärast neid pildil tuvastada [19]. Konvolutsiooniline kiht määrab, millised neuronid ühendatakse millise osaga sisendist, tehes järeldusi võrreldes skalaarset korrutusi nende kaalude ja piirkondadega sisendmahust [17].

Konvolutsioonilise närvivõrgu sisendkiht koosneb neuronitest, mis võtavad pildi pikslid ja kodeerivad need [20]. Sisendkihile järgneb konvolutsiooniline kiht. Konvolutsiooniline kiht koosneb filtrite kogumist, mis on väiksed matriksid, mis käivad üle sisendpildi ja aktiveeruvad juhul kui nad leiavad tunnusjoone pildilt [21]. Konvolutsioonilist tehet kasutades saab kihti paigutada pikslid, mis on arvutile eredamad positiivselt ja pikslid mida närvivõrk ei pea tähtsat negatiivselt [22]. Eredamad ja tähtsamad kodeeritud pikslid on need, mis aktiveeruvad filtrite üle käimisel, kuna need esindavad tunnus jooni, mida ta otsib. Tänu sellel saab, pildilt välja filtreerida tähtsamad tunnusjooned (joonis 4).

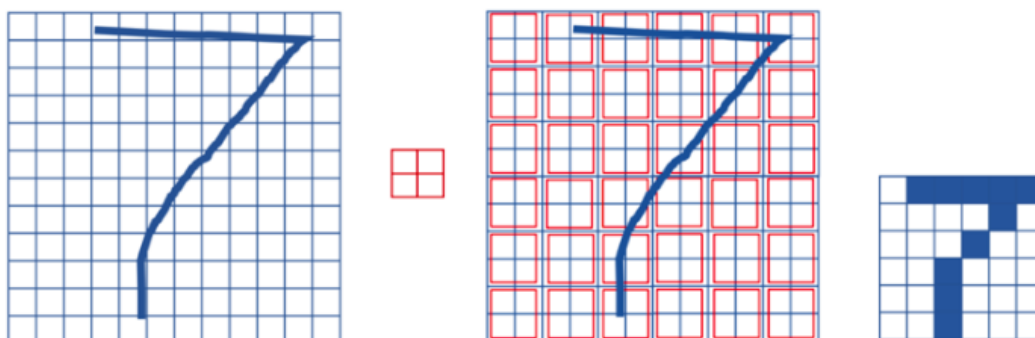


Joonis 4. Konvolutsioonilise kihi filtrite efekt [23], autori markeeringuga.

Konvolutsioonilise kihi lõpus võidakse rakendada ReLU ehk mittenegatiivne lineaarne funktsioon (*rectified linear unit*) [19]. ReLU funktsioon uurib igat sisendelementi, kui sisend on väiksem kui 0 annab ta väljundisse ka null, kui sisend on suurem kui 0 on annab ta vastava sisendi väärtuse. See on vajalik, et ebaolulised pikslid tulemusest eemaldada, kuna neid ei lähe järgmises kihis enam vaja.

2.4.2 Ahenduskiht ja täielikult ühendatud kiht

Ahendus kihis kasutatakse üldiselt ühte kahest ahendamise võimalusest, maksimum ahendamist (*max pooling*) ja keskmist ahendamist (*average pooling*). Populaarsem variant on maksimumahendamine, mille korral hoitakse konvolutsioonilisest kihist saadud sisendite maksimumväärtusi, erinevalt keskmisest ahendamisest, kus sisendite grupist arvutatakse keskmine väärtus [19]. Lihtsamalt öeldes võtab ahenduskiht konvolutsioonilisest kihist saadud sisendi ja ahendab (*downsampling*) vähendades parameetrite numbreid selles aktivatsioonis [17]. Ahenduskihis käiakse 2 x 2 maatriksitega konvolutsioonilisest kihist saadud tulemused üle ja saadud info põhjal konfigureerib väiksema maatriksi (joonis 5).



Joonis 5. Ahenduskihi lihtne näide [19].

Ahendus kihi abil vähendatakse aktivatsiooni kaardi 25% võrreldes originaalpildiga, samal ajal säilitades pildi sügavus dimensiooni [17].

Keerulisemates mudelites on üldjuhul rohkem kui üks konvolutsiooniline kiht. Sellisel juhul rakendatakse ahenduskiht peale igat konvolutsioonilist kihti [19].

Täielikult ühendatud kiht koosneb ainult nendest neuronitest, mis on läbi tulnud eelmistest kihtidest [17]. Nende neuronite põhjal tehakse väljundite järeldused. Näiteks kui on mudel loomade klassi paigutamiseks pildi järgi, otsustab täielikult ühendatud kiht, kas saadud väärtused, mis temani jõudsid, on kooskõlas linnu, koera või kassiga. Otsustab ta millisesse rühma kuulub. Täielikult ühendatud kiht tegeleb klassifitseerimisülesandega temani jõudnud sisendite põhjal.

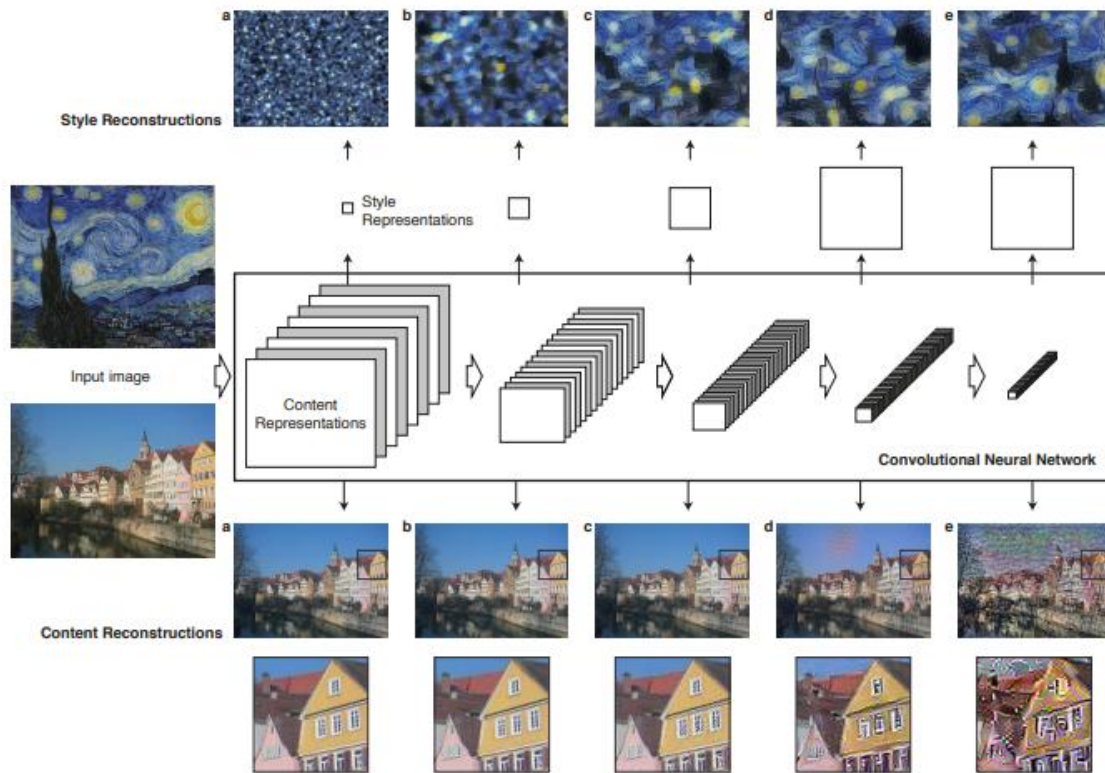
2.5 Närvivõrgu stiiliülekanne

Närvivõrgu stiili ülekanne (*neural style transfer*) on masinõppe meetod genereerimaks pilti, mis kujutaks kontekst pildisisu ja stiilipildi. Närvivõrgu stiiliülekanne kasutades saab luua kahest algsest pildist kolmanda pildi, kus tundub, et esimene konteksti pilt on tehtud teistsuguse valgustuse, teistsugusel ajal, teistsuguse ilma või hoopis teistsuguses kohas [24]. Kunstis on inimesed ära õppinud, kuidas luua erinevaid visuaalseid pilte tavalistest objektidest [25]. Maalides tavapäraselt vaasi suurte laineliste joontega saab kunstnik luua erinevaid efekte. Närvivõrgu stiiliülekanne üritatakse masinõppe abil luua erinevaid visuaalseid efekte.

Närvivõrgu stiiliülekanne põhimõte seisneb selles, et võetakse kontekstpildi (*content image*) ja stiilipildi (*style image*) ning üritatakse luua kolmas pilt, mis oleks konteksti poolest kõige väiksema erinevusega kontekstpildist ja kõige väiksema võimaliku stiilierinevusega stiilipildist [26]. Tulemusena luuakse esialgsetest piltidest kolmas pilt, sama sisu, aga teise stiiliga.

2.5.1 Närvivõrgu stiiliülekanne teooria

Närvivõrgu stiiliülekanne teostamiseks on vaja kahte funktsiooni. Ühte funktsiooni, mis kirjeldaks, kui erinevad on kahe pildi kontekstid ehk konteksti kadu (*content loss*) ja teist, mis kirjeldaks piltide erinevust nende stiilide osas ehk stiilikadu (*style loss*) [26]. Närvivõrgu stiiliülekanne juures kasutatakse tihti eeltreenitud konvolutsioonilisi närvivõrgu mudeleid, et nendelt saadud info läbi kadusid optimeerida (joonis 6).



Joonis 6. Piltide esitlused konvolutsioonilises närvivõrgus [27].

Sisend pilti, esitletakse konvolutsioonilisele närvivõrgule kui pikslite komplekti [27]. Sisendpildiks on eraldi nii stiilipilt kui ka kontekstipilt. Igal töötlus etapil suureneb filtrite arv, aga piltide suurus väheneb (ahenduskihi pärast), kuigi kõrgemates kihtides läheb piksli informatsioon kaduma, on siiski suur osa pildi kontekstist alles [27]. Seda läheb vaja, et luua kolmas pilt. Stiilikao leidmiseks kasutatakse konvolutsioonilise närvivõrgu ruumilisi funktsioone, mis leiavad stiili pildilt tekstuuri info, sellest arvutatakse vastastikune sõltuvus erinevate omaduste vahel erinevates konvolutsiooni kihtides ja erinevate kihtide kao abil arvutatakse kolmanda pildi loomise juures stiilikadu. Nii saame kätte stiili pildi info väiksemal kujul, sest vähendatakse muud pildil olevat infot, nagu näiteks paigutus [27].

Närvivõrgu stiiliülekanne töötab optimeerimistehnikal, genereeritakse uusi pilte alustades valgest müra ja info põhjal, mis saadakse kontekstpildi konvolutsioonilise närvivõrgu kihtide aktivatsioonidest ja stiilipildi Gram maatriksitest (*Gram matrix*) [28]. Konvolutsioonilisest kihist saadud info põhjal optimeeritakse pilte, arvutades konteksti kadu ($L_{content}$) ja stiilikadu (L_{style}) [28]. Kogu kao saamiseks mõlemad kaod liidetakse valemi (2) abil.

$$\mathcal{L} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style} \quad (2)$$

Valemis (2) viitavad α ja β konteksti ja stiili kadude kaaludele (*weights*), mida vastavalt kadudele muudetakse. Kaalud esindavad filtrite suurusi, millega üle piltide käiakse. Konteksti valem (3) näitab, et kontekstikadu defineeritakse tunnustekaardi (*feature map*) ja kindla kontekstipildi kindla närvivõrgu kihi tunnuste kaardi vahelise ruudus [28].

$$\mathcal{L}_{content} = \frac{1}{2} \sum_{i=1}^{N_l} \sum_{j=1}^{M_l} (F_{ij}^l - P_{ij}^l)^2 \quad (3)$$

Stiilikadu tuleb erinevate stiili kihtide kadude summast, mida demonstreerib valem (4). Erinevate kihtide stiili kaod korrutatakse kaaluga (ω) ja iga kihi stiilikadu on defineeritud tunnuste kaardi ja eri kihtide gramm maatriksi vahe ruuduga [28].

$$\mathcal{L}_{style} = \sum_l \omega_l \mathcal{L}^l_{style} \quad (4)$$

Tänu konvolutsioonilistele närvivõrkudele arvutatakse stiili ja konteksti kaod ja neid tulemusi optimeerides genereeritakse uusi pilte kuni konteksti ja stiili kaod on piisavalt väikesed, et on jõutud meelepärase visuaalse tulemuseni.

2.6 Genereerivad vastandvõrgud

Genereerivad vastandvõrgud (*generative adversarial networks*) ehk GAN on võimas klass genereerivaid mudeleid, mida tutvustati aastal 2014 Ian Goodfellow poolt [29]. Genereerivad vastandvõrgud olid omal alal uuenduslik lähenemine, sest enam ei olnud tarvis kasutada sildistatud andmestikke nagu on näiteks vaja konvolutsiooniliste närvivõrkude treenimiseks. Genereerivad vastandvõrgud on laialdaselt uuritud, kuna meetod võimaldab jõuda tulemusteni puudulike andmetega, näiteks saab genereerivate vastandvõrkude abil genereerida halva kvaliteediga pildist hea kvaliteediga pildi [30]. Genereerivaid vastandvõrke kasutatakse ka kunsti loomiseks ja katsetamiseks, näiteks piltide genereerimiseks inimestest, keda tegelikkuses ei eksisteeri. Genereerivad vastandvõrgud on võimelised looma lihtsast kriipsujukust inimese silueti [31]. Genereerivad vastandvõrgud on muutunud tehisintellekti arendamise juures populaarseteks, kuna on avastatud, et ühendades tehisintellekti ja genereerivad vastandvõrgud oleks võimalik luua masinad, kes suudab reaalselt ise asju luua [32]. Olgu siis selleks tekst, kõne või pilt.

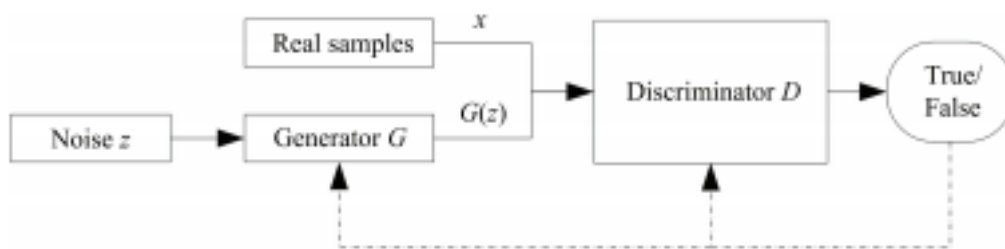
Genereerivate vastandvõrkudega katsetamise teeb kohati raskeks asjaolu, et selliste mudelite jooksutamise nõuab väga palju võimsust eriti, graafikaprotsessori seisukohalt.

Genereerivad vastandvõrgud on omal alal veel suhtelist uued ja nõuavad veel palju uurimist ja avastamist, et jõuda nende täis potentsiaalini [31].

2.6.1 Genereerivate vastandvõrkude teooria

Genereerivate vastandvõrkude idee seisneb kahe mängija omavahelisel mängul. Mängijateks on genereerival funktsioonil põhinev genereeriv mudel (generaator) ja diskrimineerival funktsioonil põhinev elimineeriv mudel (detektiiv). Generaator loob pilte ja detektiivi ülesanne on otsustada, kas need on päris või ei, generaatori ülesanne on aga genereerida pilt, mis petaks detektiivi ära, et see on päris [31]. Generaator ja detektiiv on tööpõhimõtelt teineteisest erinevad- kui detektiivi huvitab ainult, kas pildil on kass või ei, siis generaatorit huvitab, kuidas kassi pildile saada.

Genereerivate vastandvõrkude põhimõte on inspireeritud kahe mängija nullsumma mängust, milles kahe mängija võidu summa kokku on 0, ning ühe mängija võit või kaotus on täpselt tasakaalus teise mängija kaotuse ja võiduga [32]. Genereerivate vastandvõrkude korral on lihtsalt mängijateks generaator ja detektiiv. Generaator genereerib suvalisest müra-st pildi, mille annab edasi detektiivile, detektiiv on treenitud piltide andmestikuga ja otsustab, kas pilt on õige või ei, hinnates pilti nulliks ehk valeks (*false*) või üheks ehk õigeks (*true*) vahemikus (joonis 7).

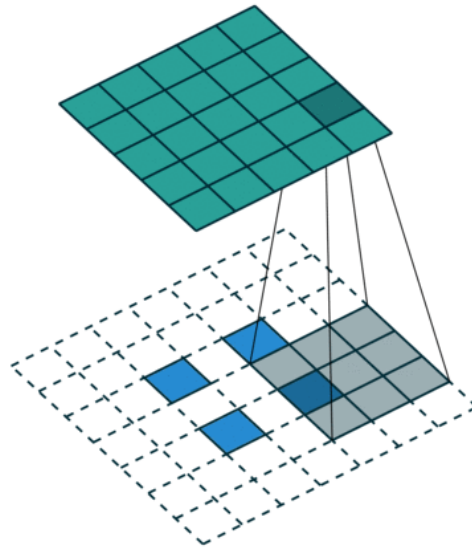


Joonis 7. GAN mudeli joonis [32].

Generaator alustab uuesti genereerimist kohandatud kaaludega. Generaatori eesmärk on suurendada tõenäosust, et detektiiv teeb vea [29]. Generaator muudab oma kaale eesmärgiga jõuda detektiivi hinnang ühele võimalikult lähedale.

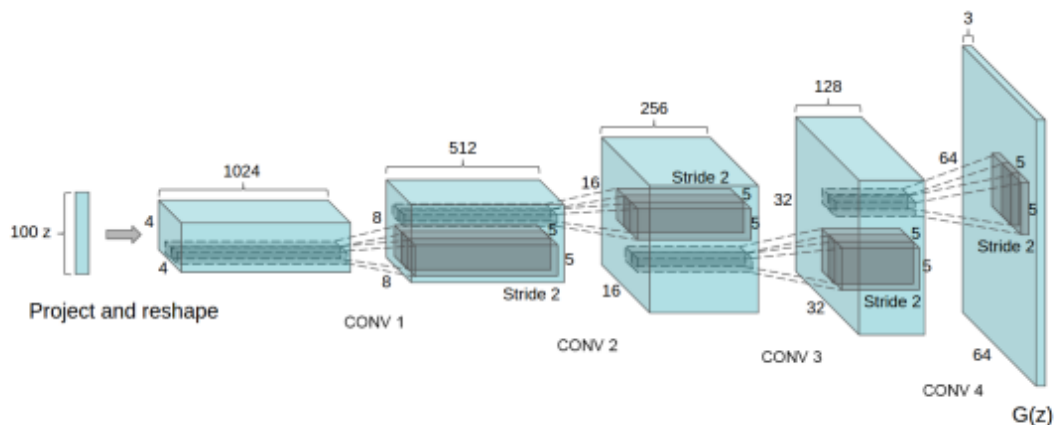
Genereerivad vastandvõrgud kasutavad konvolutsioonilisi närvivõrke. Konvolutsiooniliste närvivõrkude ja genereerivate vastandvõrkude sobitamiseks generaatoris, eemaldatakse konvolutsioonilisest närvivõrgust ahenduskihid ning asendatakse need vastand konvolutsiooniga (*deconvolution*) [33]. Erinevalt

konvolutsioonilisest närvivõrgust suurendab ta info põhjal pilti, mitte ei ahenda seda (joonis 8).



Joonis 8. Näide vastand konvolutsiooni ahendustuuma idee [33].

Vastandkonvolutsioonide abil käiakse üle suurem pilt ja arvutatakse see väiksema filtri abil kokku suuremaks pildiks [34]. Generaatoris kasutatakse vastandkonvolutsiooni kihti, et genereeriv mudel õpiks iseenda ruumilisest suurenemisest, mürast suuremat pilti luues (joonis 9).



Joonis 9. Generaatori mudeli konvolutsioonilise närvivõrgu joonis [35].

Generaator võtab sisendmüra ja hakkab seda läbi konvolutsioonilise kihi rohkete tunnuste kaartite abil suurendama kuni jõuab 64x64 suuruse pildini. See sama pilt saadetakse edasi detektiiv mudelisse, mis harutab selle jälle lahti väiksemateks kihtideks ja otsustab, kas pilt on tõene või väär [36]. Detektiiv töötab sarnaselt tavalisele konvolutsioonilise

närvivõrgule. Detektiivimudeli viimases konvolutsioonilises kihis tulemus lamendatakse (*flattening*) ehk viiakse ühte. Näiteks kui enne oli tulemuseks 3x3 maatriks siis pärast lamendamist on 1x9, kus esimesed kolm väärtust on maatriksi esimese rea väärtused ja nii edasi. Lamendatud maatriks suunatakse sigmoidfunktsiooni valem (5), mille abil viiakse väärtused 0 ja 1 vahele [35].

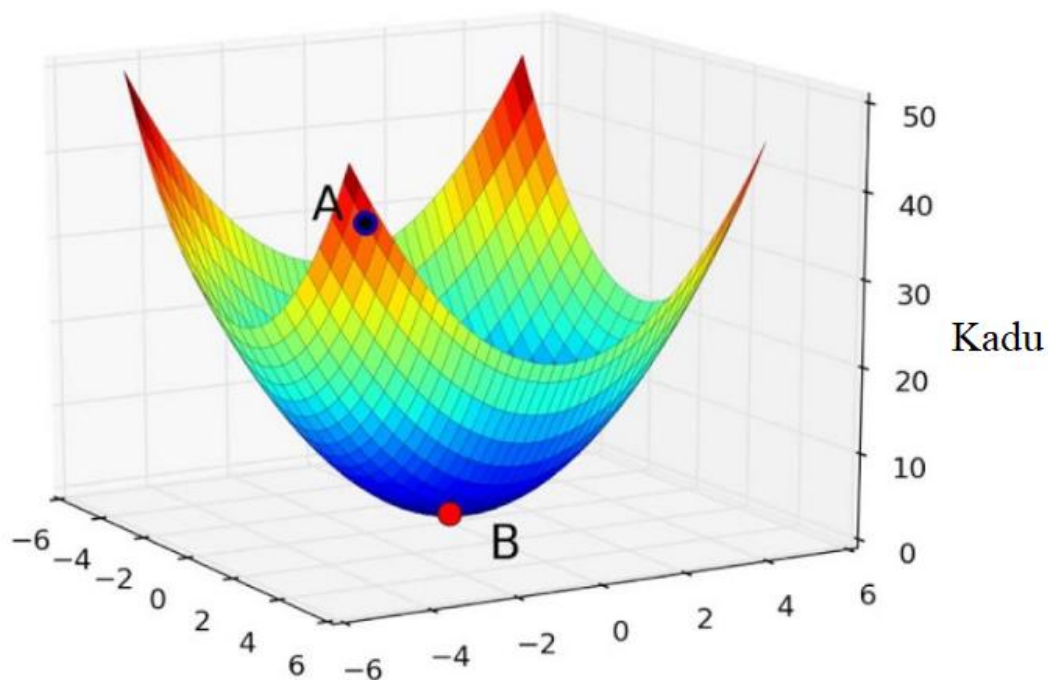
$$f(x) = \frac{1}{1 + e^{-z}} \quad (5)$$

Otsustanud, kas pilt on tõene või väär ja arvestades kas tulemus kaldub rohkem nulli või ühe poole saadab detektiiv oma vastuse generaatorile ja generaator hakkab saadud pildi põhjal uut pilti genereerima.

2.6.2 Genereerivate vastandvõrkude treenimine

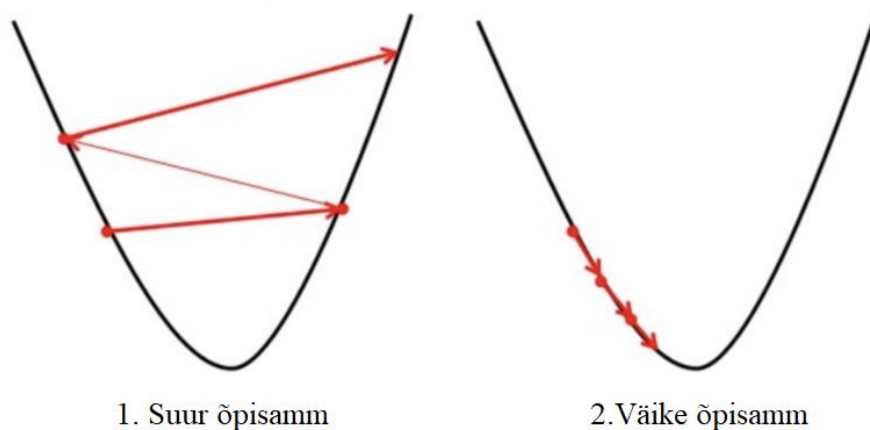
Genereerivaid vastandvõrke treenitakse kasutades erinevaid optimeerimismeetodeid, mis aitavad kadude ja gradientide abil genereerimisel kasutatavaid kaale ja vabaliikmed kohandada.

Genereeriva vastandvõrgu tulemuste parandamiseks kasutatakse erinevaid optimeerimistehnikaid, millega leitakse kaod ja gradiendid, mille abil kohandatakse kaale ja vabaliikmeid. Kõige populaarsem neist on gradientlaskumise optimeerimine (Gradient Descent Optimizer) [37]. Eesmärk on lihtne - jõuda läbi treenimise minimaalse kaotuse kohani (joonis 10).



Joonis 10. Kao miinimumi visuaalne näide [37], autori markeeringuga.

Treenimist alustatakse punktis A ning läbi kadude ja gradientide abil arvutusi tehakse kohandatakse kaale ja vabaliikmeid ning iga treeningsammuga jõutakse lähemale punktile B (joonis 10). Treeningsammudel gradientide leidmisel mängib suurt rolli õpisamm (*learning rate*) (joonis 11), mille abil liigutakse miinimumini. Õpisamm mängib väga suurt rolli genereerivate vastandvõrkude treenimisel, kuna valides liiga suure võib juhtuda, et pörgatakse edasi tagasi ja ei jõutagi miinimumini [37].



Joonis 11. Õpisammu olulisus [37], autori markeeringuga.

Kuna gradientlaskumise optimeerimise toimimiseks peab olema õpisamm väga väike on välja mõeldud teisi optimeerimis tehnikaid, et saaks õpisammu suurendada ja kiiremini

miinimumini jõuda. Populaarseteks peetakse näiteks Geoff Hintoni poolt loodud ruutkeskmise laiendusega optimeerimist (*root mean square propagation*) ehk lühendatult RMSprop [38].

Ruutkeskmise laiendusega optimeerimine on oma nime saanud, kuna kasutatakse ruut tehet, et liikuda miinimumile kiiremini lähemale ilma suuri kõrvalepõikeid tegemata. Erinevalt gradientide laskumis optimeerimise viisist, millel on terve treenimise aja kindel õpisamm, ruutkeskmise laiendamisega optimeerimine kohandab oma õpisammu kasutades selleks gradientide ruutude vahet. Ruutkeskmise laiendamisega optimeerimine arvestab iga gradiendi vahelise suurusega, tagades, et jõudes miinimumkohale lähemale, aeglustatakse õppesammu ning ei hüpata miinimumkohast üle [37]. Ruutkeskmise kasutamisel on soovitatav esialge õpisamm 0.001 [39].

Ruutkeskmise laiendamisega optimeerimisest on veidi populaarsem hetkeväärtustega kohanduv optimeerimine (*Adaptive Moment Estimation*) ehk lühendatult Adam [38]. Nimi tuleneb sellest, et optimeerimiseks jälgitakse erinevate gradientide olekuid erinevates hetkedes. Sarnaselt ruutkeskmise laiendusega optimeerimisele erineb ka hetkeväärtustega kohanduv optimeerimine gradientlaskumise optimeerimisest, kuna kohandatakse treenimise käigus õpisammu [39]. Lisaks kasutatakse ära ka eelmiste gradientide vahede keskmisi. Hetke väärtustega kohanduv optimeerimis tehnikaks soovatakse esimeseks õpisammuks valida 0,001 [40].

2.7 Tehnoloogia

Antud lõputöös on kasutatud koodi keeleks Pythonit ning selle laiendusi ja teeke. Python töötab hästi koos paljude erinevate rakenduste ja mudelitega [41]. Pythonit on ka palju pilditöötlus teeke nagu näiteks PIL ja CV2 ja masinõppe teeke, mis aitavad erinevaid piltidega seotuid funktsioone läbi viia.

Genereerivate vastandvõrkude mudeli jooksumiseks kasutati antud töös Google Pilve (*Google Cloud*) pakutatavat võimalust jooksumata programme võimsama graafikaprotsessoriga läbi brauseri konsooli.

2.7.1 TensorFlow ja Keras

TensorFlow on võimas teek masinõppe jaoks, mida saab rakendada ka närvivõrkudele [42]. TensorFlow pakub rakendusliideseid ehk API-liideseid paljudele programmeerimiskeeltele, mille hulka kuulub ka Python. Samuti on TensorFlow avatud lähtekoodiga, mis võimaldab tal olla ajaga kaasas käiv. TensorFlow on tuntud oma andmevoo graafide (*dataflow graphs*) poolest (mis võimaldavad erinevate seadmete peal, nagu näiteks graafikaprotsessorid, jooksutada ahendustuumasi) ning tensorite, mis lubavad läbi viia keerulisi maatriksarvutusi [42]. Antud töös on kasutatud TensorFlow teeki mõlemas mudelis nii närvivõrgu stiilisülekandel kui ka genereeriva vastandvõrgu mudeli puhul.

Keras on kõrge tasemega närvivõrgu rakendusliides, mis võimaldab eksperimenteerida erinevate masinõppe teekidega nagu näiteks TensorFlow [43]. Keras sobib hästi närvivõrkude genereerimiseks, sest ta on lihtne ja tema kasutamise kohta leidub palju juhendeid. Antud töös on Kerast kasutatud närvivõrgu stiilisülekande mudelis konvolutsioonilist närvivõrku luues. Töös on kasutatud ka Kerase eeltreenitud närvivõrke VGG16 ja VGG19.

2.7.2 Google graafikaprotsessor

Genereerivate vastandvõrkude mudelite töötamine nõuab palju võimsust graafikaprotsessorilt. Mudeli tööprotsessi kiirendamiseks on antud töös kasutatud Google teenust, mis võimaldab taotleda võimalust kasutada läbi brauseri Google pilve graafikaprotsessoreid, mis on tunduvalt võimsamad ja võimaldavad kiiremini tulemusi kätte saada.

3 Mudelid

Antud lõputöö käigus loodi kaks mudelit, üks närvivõrgu stiiliülekanne optimeerimistehnikaga, mille sisendiks on üks portree ja karikatuur ja väljundiks kolmas pilt karikatuurina. Teine mudel on loodud genereerivate vastandvõrkude teooria põhjal, millega üritatud jõuda genereeritud tulemusteni, mis meenutaks karikatuure.

3.1 Andmestikud

Antud lõputöö jaoks loodi kaks andmestiku, üks närvivõrgu stiiliülekanne mudelile ning teine genereeriva vastandvõrgu mudeli jaoks. Kuna närvivõrgu stiilimudeli puhul sisestatakse korraga üks portree ja üks karikatuur, koguti selle tarvis 20 portreed ja 20 karikatuuri internetist otsides kuulsate inimeste portreesid ja karikatuure läbi Google otsingu. Genereeriva vastandvõrgu mudeli tarbeks loodi esimeste katsetuse jaoks andmestik 700 karikatuuriga. Andmestikeks olevad karikatuurid on enamus võetud erinevatest Pintresti kollektsioonidest, kus leidub palju erinevates stiilides karikatuure. Karikatuuride kogumist 200 karikatuuri on võetud Caroly Huobolaineni lõputöö „Masinõppe abil huumori tuvastamine karikatuuride ja portreede näitel“ karikatuuride andmestiku kogumist [21]. Kogutuid karikatuure oli erinevates stiilides (joonis 12 ja 13). Kõik karikatuurid keskenduvad ühe inimese näole.



Joonis 12. Karikatuur Pintresti kogumist [44].



Joonis 13. Karikatuur Caroly Huobolaineni kogumist [21].

Igat pilti leidub andmestikes üks kord ning pildid ei kordu. Ühest ja samast inimesest erinevaid karikatuure andmestikus esineb.

3.2 Närvivõrgu stiili ülekande mudel

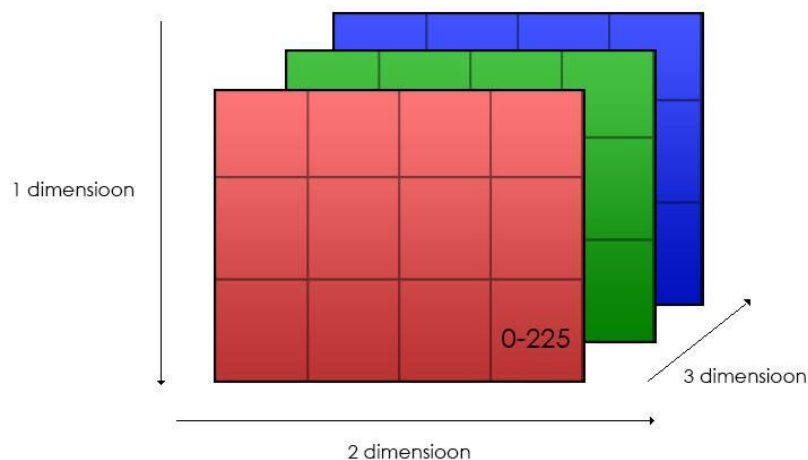
Närvivõrgu stiiliülekande meetodi katsetamiseke loodi kaks erinevat mudelit. Mõlema mudelite katsetamiseks kasutati sama andmestikke. Mudelid erinevad teineteisest, kuna esimesel juhul on kasutatud Kerase mudelit VGG 16 (*Very Deep Convolutional Networks for Large-Scale Image Recognition*) ehk sügav konvolutsioonilised närvivõrkude suurte

mõõtmeliste piltide tundmiseks [45]. Teise mudeli puhul on kasutatud Kerase mudelit VGG 19, mis kujutab samuti sügavate närvivõrkude kogu suurte mõõtmeliste piltide eristamiseks [46]. VGG 16 ja VGG 19 vahe seisneb, et ühe puhul sisaldab mudel kuusteist konvolutsioonilist kihti ja teine üheksateist konvolutsioonilist kihti.

3.2.1 VGG 16 mudel

Mudeli loomiseks on kasutatud koodi Hunter Heiderneichi loodud dokumendist, mis demonstreerib närvivõrgu stiiliülekannet [47]. Sisendandemteks antakse stiili ja konteksti pilt, iteratsiooni arv ja piltide suurused. Paremate tulemuste jaoks on konteksti- ja stiilpildid eelnevalt töödeldud ühe suurusteks ja ruudukujuliseks, et tulemused oleksid võimalikult head. Seda on tehes kasutades rakendust nimega Photoshop. See on ka vajalik, et pildid ei läheks propotsioonidest välja kui kood nad ruudukujuliseks muudab.

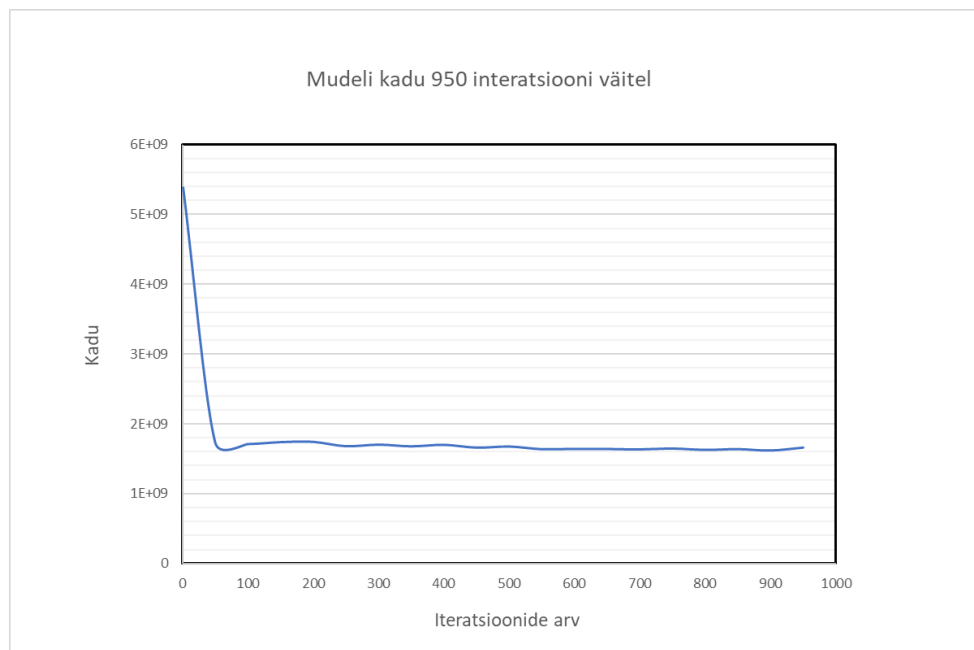
Kuna sisestatud pildid on RGB skaalas ehk koosnevad punasest, rohelisest ja sinisest kihist (joonis 14). VGG 16 mudeli kasutamiseks peab pildid viima BGR skaalale ehk esimesena esitatud sinine siis roheline ja siis punane.



Joonis 14. Pildi esitus (autori joonis).

Selle tõttu tõstetakse pildi andmed ümber ja muudetakse dimensioonidega massiiviks, mis sobitaks VGG16 konvolutsioonilisse närvivõrku. Konvolutsioonilisest närvivõrgust võetakse valitud kihid ja nende abil toimub erinevate kadude arvutamine. Lisaks stiili ja kadudele arvutatakse mudelis ka erinevuse kadu (*variation loss*), mis peaks tagama selle, et pildid ei muutuks liiga pikslilisteks. Koodi jooksutati kasutades Anaconda keskkonda, millele installiti vaja minevad TensorFlow ja Keras.

Koodi katsetades jooksutati pilte erineva iteratsioonide arvuga. Selgus, et peale esimesi iteratsioone kadu enam oluliselt ei parane (joonis 15).



Joonis 15. Närvivõrgu VGG16 mudeli kadu (autori joonis).

Sellest võib järeldada, et olenevate iteratsioonide suurendamisest tulemused olulised ei erine. Seda oli näha ka tulemustest, 20 iteratsiooni korral oli tulemus sarnane, mis 1000 iteratsiooni korral (joonis 16).



Joonis 16. Tulemuste võrdlus iteratsiooni arvu põhjal (autori joonis).

Katsete põhjal võib samuti järeldada, et mudel oma ülesandega karikatuure luua hakkama ei saanud. Kuigi algsel pildil muutusid värvid ja jooned (joonis 17), siis proportsioonid,

mis eristavad karikatuuri portreest, ei muutunud ja kuna proportsioonid on näol paigas, jätab ta mulje portreest, mitte karikatuurist.



Joonis 17. Mudeli tulemus, konteksti pilt [21], stiilipilt [48], autori markeeringuga.

Tulemuste põhjal võib järeldada, et mudel ei olnud edukas karikatuuride loomisel. Mudel küll suutis muuta konteksti pildi värve, aga karikatuuri efekti ta ei loonud.

3.2.2 VGG 19 mudel

Mudeli loomiseks on kasutatud Prezmek Pobrtynyni koodi [49], mis mõeldud närvivõrgu stiiliülekanne tegemiseks kasutades VGG 19 eeltreenitud versiooni [46]. Idee on sama, mis VGG 16 mudeli puhul. Muudetakse pildid dimensiooniliste maatriksitena konvolutsioonilisse närvivõrku ja seal olevate kindlate kihtide tulemuste põhjal tehakse arvutusi, et leida kogu kadu. Mudelit katsetati erinevate piltidega, erinevate pildi suuruste ja kvaliteediga, katsetades erinevaid iteratsioonide arve.

VGG 19 mudeli puhul oli näha, et iteratsiooni arvude kasvades, suurenes ka genereeritud piltide erinevus esialgsest kontekstpildist (joonis 18). Sellest võib järeldada, et kao arvutamise valem oli selles mudelis suutlikum.



Joonis 18. Iteratsioonide võrdlus (autori joonis), konteksti pilt [50] , stiilipilt [48].

Katsed näitasid, et tulemuse mõjutas kaks tegurit. Nimelt oli genereeritava karikatuur oluliselt kvaliteetsem kui pilt oli suurem ja kvaliteetsem. Mida suurem ja kõrgema kvaliteediga oli pilt, seda paremini olid näha erinevad detailid genereeritud piltidel (joonis 19 ja 20).



Tulemuse pilt



1.Konteksti pilt



2.Stiilipilt

Joonis 19. 512 x 512 suuruse pildi tulemuse kvaliteet (autori joonis), konteksti pilt [51], stiilipilt [52].



Tulemuse pilt



1.Konteksti pilt

2.Stiilipilt

Joonis 20. 1000 x 1440 tulemus pildi kvaliteet (autori joonis), konteksti pilt [53], stiilipilt [54].

Suurema konteksti ja stiili pildiga võttis mudel ka oluliselt rohkem aega iteratsioonide läbimiseks. See on ka loogiline, kuna kvaliteetsemad ja suuremad pildid koosnevad rohkemast arvust pikslitest, millega mudel arvutusi teeb. Käivitades mudeli stiili ja konteksti piltidega, mis mõlemad suuruses 512 x 512, läks mudelil 950 iteratsiooni läbimiseks 4 tundi ja 37 minutit. Piltidega suuruses 1000 X 1140 läks sama arvu iteratsioonide läbimiseks 10 tundi ja 49 minutit.

Tulemuspiltidel on näha, et mudel viib stiili jooned üle teisele pildile (joonis 21).



Joonis 21. Tulemus 500 iteratsiooniga 680x780 suuruses (autori joonis) konteksti pilt [55], stiilipilt [56] Genereeritud pildile tekitatakse sarnased värvitoonid, mis konteksti pildil. Parimad tulemused (joonis 22) saadi samasuuruste piltide ja sarnaste värvidega. Genereeritavate piltide näo propotsioonid jäid siiski suhteliselt lähedale algsele kontekstpildile ehk portreedele.

3.3 Genereeriva vastandvõrgu mudel

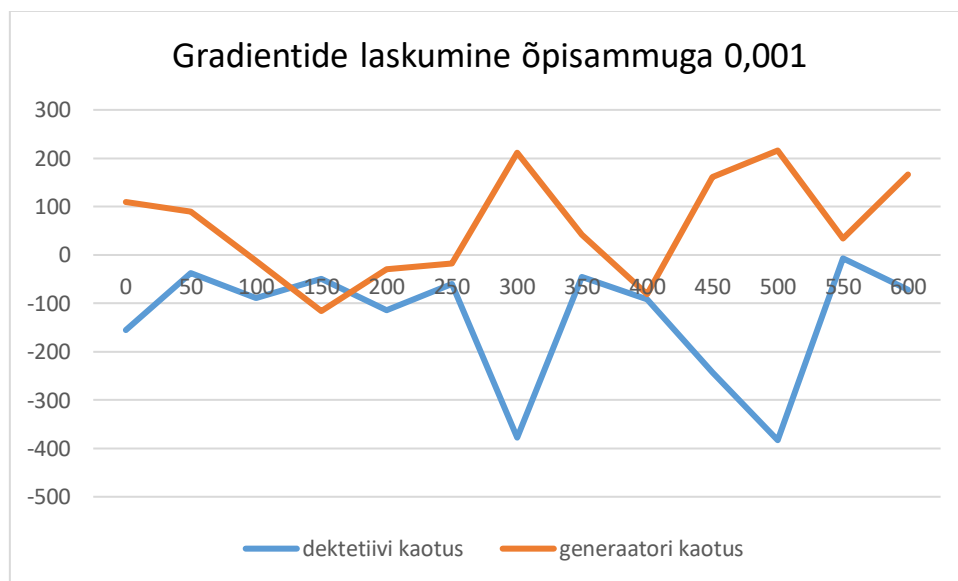
Genereeriva vastandvõrguga mudeliga loomiseks kasutati Siray Ravali koodi [57]. Genereeriva vastandvõrgu mudeli puhul tuli ilmsiks, et selle töötamiseks on vaja võimast graafikaprotsessorit, sest sülearvuti oma ei ole piisav. Mudeleid käivitati seetõttu Google pilves. Mudeli jooksumine kõrgete iteratsiooni arvudega võtab mitu päeva aega, otsustati kõigepealt leida parim optimeerimismeetod. Parimate tulemuste saamiseks viidi läbi katseid kolme erineva optimeerimismeetodiga. Optimeerimismeetoditega viidi läbi katseid testides mudelit erinevate optimeerimismeetoditega 600 epohhiga.

3.3.1 Optimeerimismeetodi valimine

Katseid viidi läbi gradientlaskumise optimeerimisega [58], ruutkeskmise laiendusega optimeerimisega [59] ja hetkeväärtustega kohanduva optimeerimisega [60]. Optimeerimismeetodite juures kasutati erinevaid õpisamme. Läbiviidut katsete põhjal valiti optimeerimismeetod millega hakati karikatuure genereerima.

Sobivaima optimeerimismeetodi leidmiseks lasti neil joosta 700st karikatuurist koosnevas andmestikus 600 korda ja hinnati nende tööd nii detektiivide kaod, generaatori kaod ning genereeritud piltide järgi. Otsiti optimeerimistehnikat, mille kaod liiguks ühtlaselt miinimumkohale lähemale ja ka mille genereeritavad pildid seda kajastaks.

Kõige kehvemaks osutus antud mudeli puhul gradientlaskumisega optimeerimine õpisammuga 0,001 (joonis 22).



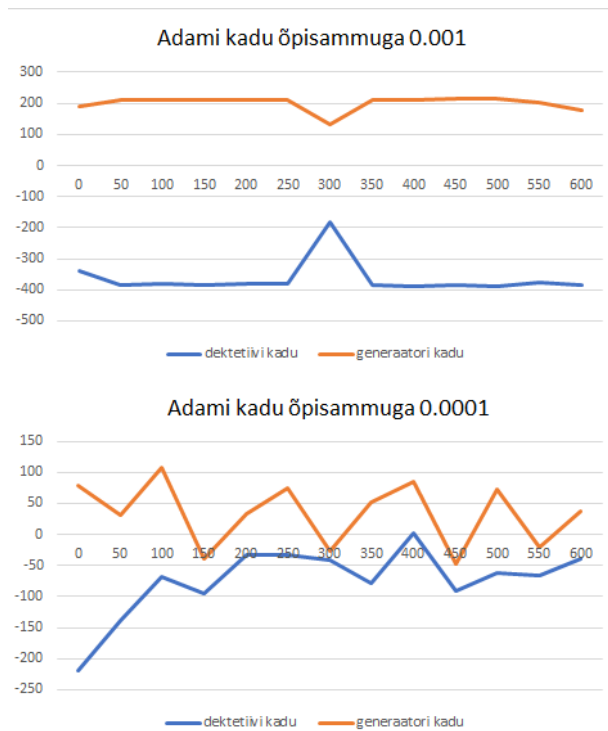
Joonis 22. Gradientlaskumine sammuga 0,001 (autori joonis).

Mudeli kaod erinesid igal treeningsammul teineteisest, suurim vahe tuli sisse detektiivide kaod puhul 500 epochi ja 550 epochi vahel, kus detektiivide kadu liikus -387 pealt -7 peale. Genereeritud pildid 600 epochi juures oli pildidel ebamäärased kujundid suurte valgete kohtadega (joonis 23).



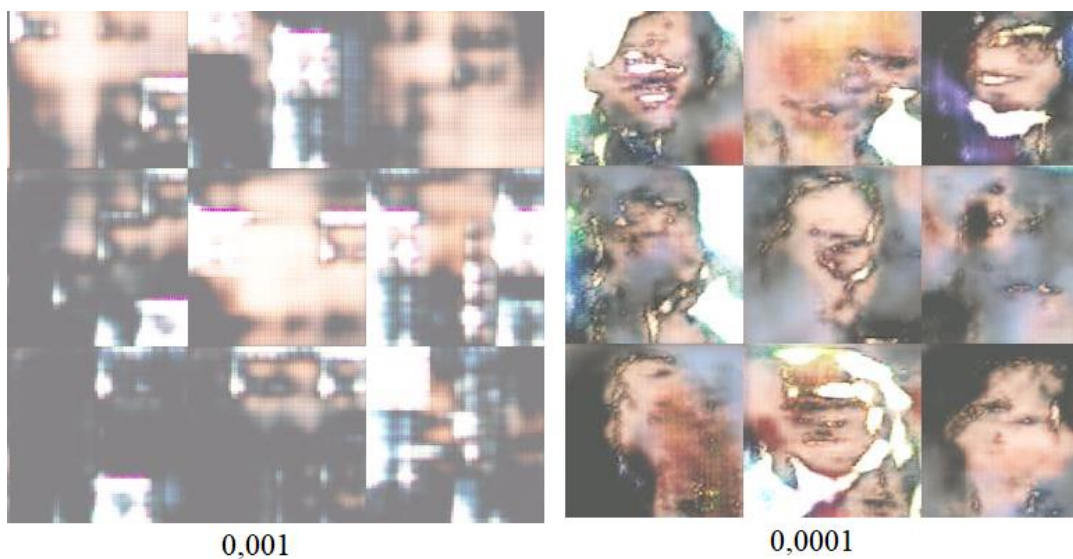
0,001

Joonis 23. Gradientlaskumise optimeerimisega tulemused 600 epohh korral (autori joonis). Hetkeväärtustega kohanduva optimeerimise katsetamisel viidi läbi kaks katset. Üks õpisammuga 0,001 ja teine 0,0001. Antud katsest tundus, et samm 0,001 on hetkeväärtustega kohanduva optimeerimise puhul parem, sest kaod ei erinenud epohhide vahel teineteisest suuresti, erinevalt tulemustest õpisammuga 0,0001, mille puhul varieerusid kaod mitmekümne ühiku võrra epohhide vahel (joonis 24).



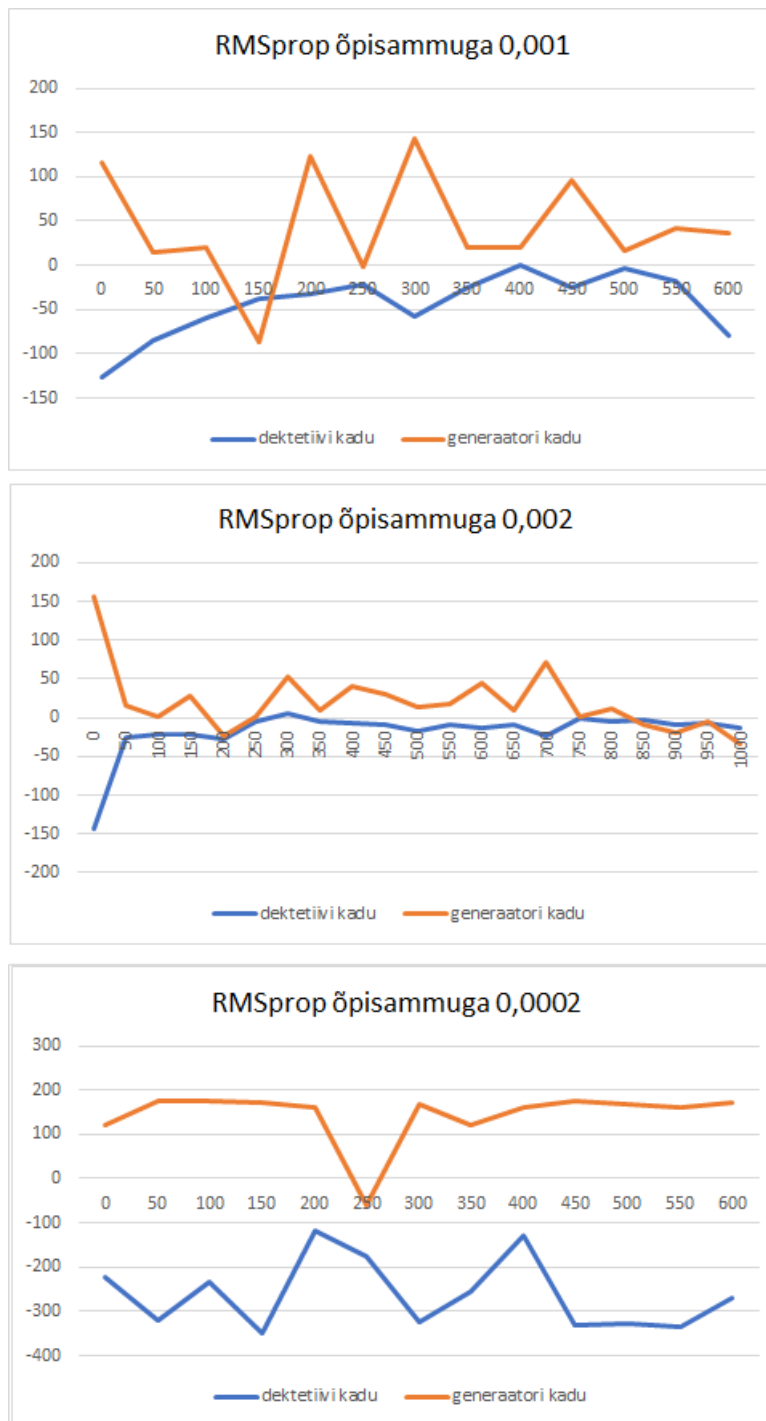
Joonis 24. Adam võrdlus 0.001 õpisammu ja 0.0001 õpisammu vahel (autori joonis).

Genereeritud pildid aga näitasid, et õpisamm 0,001 korral ei hakanud pildid 600 epohh korral inimese pea kuju võtma (joonis 25). Genereeritud piltide põhjal saab järeldada, et õpisammuga 0,0001 on 600. epohhi puhul tulemused hakanud rohkem lähenema inimese pea kujule.



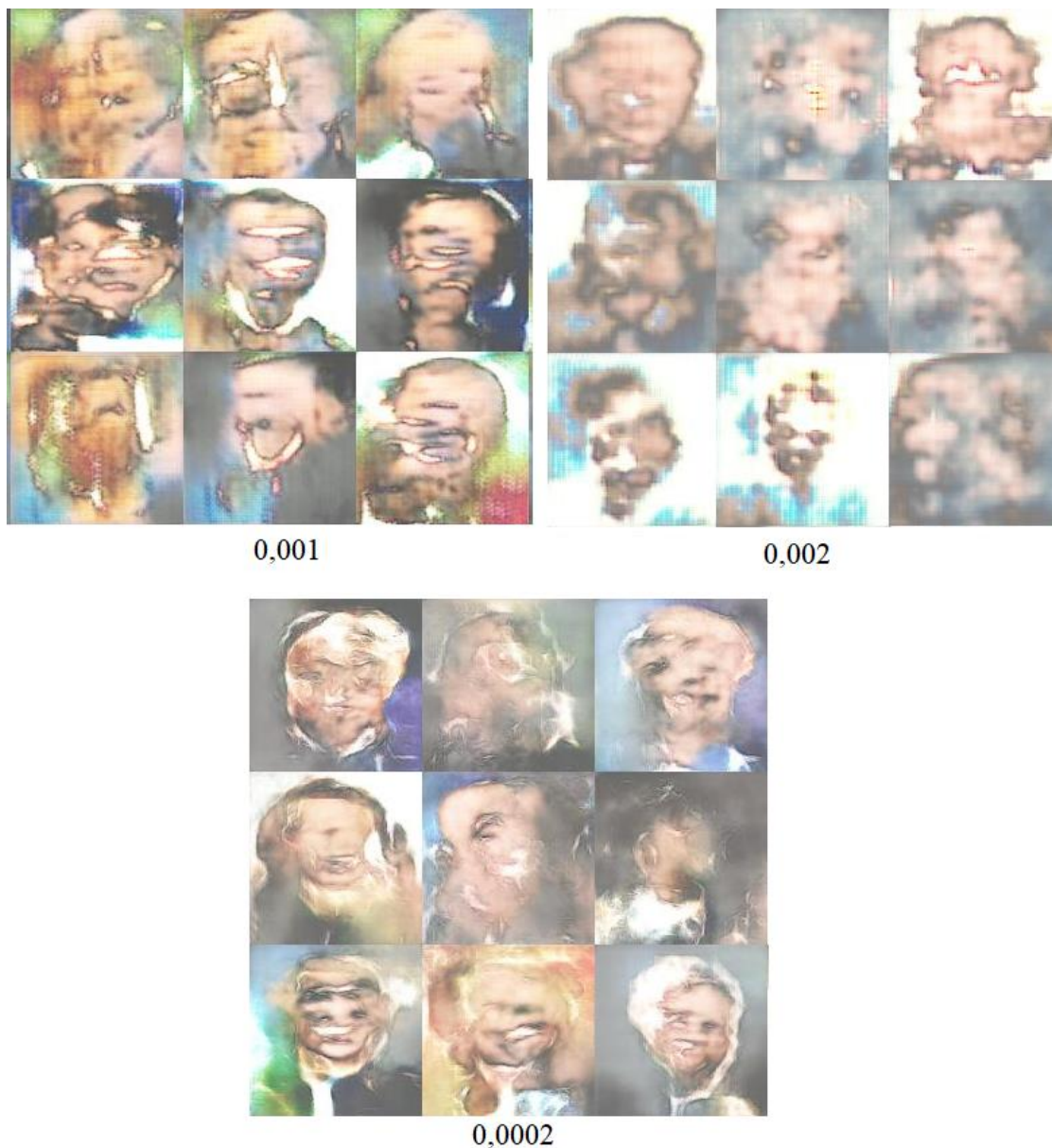
Joonis 25. Adam tulemused õpisamm 0,001 ja 0,0001 korral (autori joonis).

Ruutkeskmise laiendusega optimeerimisega katsetati õpisammu 0,001, 0,002 ja 0,001. Kõige kehvemaks antud katsetustest osutus õpisamm 0,002, sest tema kaod koondusid liialt (joonis 26). Mudel õpisammuga 0,001 muutis detektiivide kadu epohhide vahel tunduvalt rohkem kui generaatori kadu. Kadude järgi hinnates tunduks kõige efektiivsem õpisamm 0,002.



Joonis 26. RMSprop võrdlus 0,001 , 0,002 ja 0,0002 vahel (autori joonis).

Genereeritud piltide põhjal (joonis 27) tehti järeldus, et õpisammuga 0,0002 loodud pildid tunduvad enim sarnanevat karikatuuridele võrreldes teiste õpisammude ja optimeerimismeetodite tulemusi.



Joonis 27. 600. genereeritud piltide võrdlus erineva õpisammu vahel.

Katsete põhjal otsustati karikatuuride loomise katseid läbi viia ruutkeskmise laiendusega optimeerimisega ning õpisammuks kasutada 0,0002 ja 0,0001.

3.3.2 Mudeli tulemused

Mudeli tehti läbi kolm katsset. Esimesel katselt töötas mudel 5000 epohhiga, kasutades ruutkeskmise laiendamisega optimeerimist õpisammuga 0,0002. Mudel töötas 58 tundi ja 34 minutit. Tulemused ei meenuta karikatuure, aga osadelt on näha silmade, pea ja suu moodustumist (joonis 28).



Joonis 28. Genereeritu tulemused 5000 epohhiga 0,0002 õpisammu puhul.

Teise katse korral jooksutati mudelit õpisammuga 0.0001. Mudel jooksis 1500 korda ja genereeritud tulemused ei sarnane karikatuurile (joonis 30). Mudel töötas 20 tundi ja 17 minutit.



Joonis 29. Parimad tulemused õpisammuga 0.00012

Viimase katse korral suurendati epochide arvu 7000 peale. Erinevalt esimestest katsetest, kus iga epochi kohta töötas detektiiv viis ja generaator ühe korra, siis 7000 epochi korral töötas detektiivi kolm korda ühe epochi jooksul ja generaator ühe. Mudel jooksis 67 h ja 43 minutit.



Kõigi mudelite puhul on genereeritud pildid suhteliselt udused ja kindlaid näojooni pole lihtne eristada. Parimaks tulemuseks võiks lugeda esimese või kolmanda mudeli tulemusi.

Andmestikus, millega mudelid treeniti, leidis rohkem karikatuure meestest kui naistest ja sellepärast meenutavad enamik genereeritud pilte pigem meest.

3.4 Järeldused

VGG 16 mudeli tulemused muutsid konteksti pildi heledamatesse toonidesse. Näo propotsioonid jäid samaks, mis konteksti pildil ning sellepärast meenutas genereeritud pilt rohkem portreed kui karikatuuri. VGG 16 mudel lõi efekti nagu oleks portree joonistatud, aga karikatuuriks seda nimetada ei saa (joonis 30). Mudel ei ole värve genereeritud piltidele erksalt edasi teiseandanud.



Joonis 30. Närvivõrgu stiilimudeli VGG16 näite tulemused.

Närvivõrgu stiiliülekanne mudel VGG 19 suutis genereeritavatele piltidele genereerida erksamad värvid kui mudel VGG 16. Tulemustest (joonis 31) saab järeldada, et stiiliülekanne mudel oli edukas kuna erinevad stiili pildid tekitavad erinevaid efekte konteksti pildil, muutes sellega kohati ka inimese näo proportsioone.



Joonis 31. Närvivõrgu stiiliülekanne VGG 19 mudeli näite tulemused

Genereeriva vastandvõrgu mudel ei olnud piisavalt efektiivne karikatuuride loomisel. Genereeritud pilte uurides leiab küll inimese tunnusjooni, nagu nägu, suu, silm, juuksed või ära tuntava riietuse, aga peale vaadates ei meenuta genereeritud pilt karikatuuri (joonis 32).

Mudelite tulemustes saab kõige karikatuuri lähedasemaks pidada VGG 19 mudeliga genereeritud pilte. Kuigi genereerivate vastandvõrkude loodud tulemustel on näo proportsioonid kõige rohkem paigast ära, ei ole äratuntav, et tegu on inimesega. VGG 19 mudeli tulemuste nägude proportsioonid erinevad küll tavapärastest proportsioonidest vähe, aga piisavalt, et luua karikatuuri efekti.



Joonis 32. RMSprop õpisammuga 0,0002 epohh 6800.

Karikatuuride loomisel genereerivate vastandvõrkudega ei jõutud soovitud tulemusteni ilmselt mitmel põhjusel. Üheks mudeli miinuseks võib kindlasti pidada seda, et konvolutsioonilisi kihte oli suhteliselt vähe ja ei kasutatud ahenduskihte. Mudelit saaks efektiivsemaks luua kasutades rohkem konvolutsioonilisi kihte, ahenduskihte või hoopis eeltreenitud konvolutsioonilist närvivõrku. Mudelit aeglustas ka kindlasti asjaolu, et samal ajal treeniti nii detektiiv kui generaatorit, ette treenitud detektiiv kasutades jõuaks mudel kiirema ajaga läbida rohkem epohhe ning kiiremini liikuda minimaalse kaotuse lähedusse.

4 Kokkuvõte

Töö eesmärk oli luua mudelid karikatuuride loomiseks ja analüüsida nende ülesandega toimetulekut. Karikatuuride loomisel tuli välja, et karikatuuride puhul on kõige olulisem, et inimene oleks ära tuntav, aga näo proportsioonid peaksid siiski piisavalt tavapärasest erinema, et oleks aru saada, et tegemist on karikatuuriga. Proportsioonist väljas olemine tagab ka karikatuuri huumoriefekti.

Närvivõrgu stiiliülekanne mudeli puhul tuli esile, et eeltreenitud konvolutsiooniline närvivõrk üheksateistkümne konvolutsioonilise kihiga on närvivõrgu stiiliülekanne võimeline looma suuremat efekti genereeritud piltide puhul, kui mudel kuueteistkümne konvolutsioonilise kihiga. Piltidega katseid tehes tuli veel esile, et oluline on pildi suurus ja kvaliteet. Nendest sõltub, kui hea kvaliteediga on genereeritav pilt. Suurte piltide genereerimine võtab seevastu oluliselt kauem aega. Mudelite puhul tuli ka esile, et närvivõrgu stiiliülekanne puhul on oluline, et stiili ja konteksti pilt oleksid ühesuursed, sest see tagab parimad võimalikud tulemused.

Genereerivate vastandvõrkude puhul osutus oluliseks, milliseid optimeerimismeetodeid kasutada ja millise õpisammuga liikuda. Samuti tähendati antud ülesande juures, et genereerivate närvivõrkude puhul on oluline, et oleks võimalus programmil töötada seadme peal, millel on võimas graafikaprotsessor, sest tavaline sülearvutile ei pea genereerivate vastandvõrkude treenimisele vastu.

Karikatuuride loomisega sai kõige paremini hakkama närvivõrgu stiiliülekanne mudel VGG 19. See suutis luua uue pildi, mis sarnanes stiililt karikatuurile.

Edasimineku mõttes saaks kindlasti edasi arendada genereerivate vastandvõrkude mudelit, testides erinevaid treenimissamme ja lasta joosta rohkematel epochidel. Lisaks saaks veel lisada mudelisse rohkem konvolutsioonilisi kihte ja uurida ka võimalust alustada karikatuuri loomist portreest, mitte mürast.

Kasutatud kirjandus

- [1] D. Perkins, *A Definition of Caricature and Caricature and*, Pennsylvania: Penn Libraries, 1975.
- [2] E. Alpaydin, *Introduction to Machine Learning*, London, Cambridge: The MIT Press, 2010.
- [3] T. B. Lee, „Why Google believes machine learning is its future,“ *arstechnica*, 10 Mai 2019. [Võrgumaterjal]. Available: <https://arstechnica.com/gadgets/2019/05/googles-machine-learning-strategy-hardware-software-and-lots-of-data/>. [Kasutatud 8 Mai 2019].
- [4] T. Matiisen, „Närvivõrkude ja masinõppe sõnastik,“ *Data Science Estonia*, [Võrgumaterjal]. Available: <http://datasci.ee/masinoppe-sonastik/>. [Kasutatud 20 April 2019].
- [5] „IT terministandardi sõnastik,“ *Eesti Keele Instituut*, 1998-. [Võrgumaterjal]. Available: <http://www.eki.ee/dict/its/>. [Kasutatud 21 April 2019].
- [6] J. Parton, „*Caricature and Other Comic Art in All Times and Many lands*,“ Harper & Brothers, New York, 1877.
- [7] J. L. L. Y. Kaidi Cao, „CariGANs: Unpaired Photo-to-Caricature Translation,“ 22 November 2018. [Võrgumaterjal]. Available: <https://cari-gan.github.io/>. [Kasutatud 1 Mai 2019].
- [8] M. J. S. Rudwick, „Caricature as a Source for the History of Science: De la Beche's Anti-Lyellian Sketches of 1831,“ *Chicago Journals*, kd. 66, nr 4, pp. 534-560, 4 December 1975.
- [9] W. Ames, „britannica,“ *Encyclopædia britannica*, 20 September 2017. [Võrgumaterjal]. Available: <https://www.britannica.com/art/caricature-and-cartoon>. [Kasutatud 1 April 2019].
- [10] Lumen_V, „Caricature BOX01,“ *Behance*, 1 Detsember 2009. [Võrgumaterjal]. Available: <https://www.behance.net/gallery/358480/Caricature-BOX01>. [Kasutatud 24 April 2019].
- [11] R. Mauro ja M. Kubovy, „Caricature and face recognition,“ *Memory & Cognition*, kd. 4, nr 20, pp. 433-440, 1992.
- [12] D. Sahlin, *Digital Portrait Photography For Dummies*, Indianapolis: Wiley Publishing, 2010.
- [13] R. Miotto, F. Wang, S. Wang, X. Jiang ja J. T. Dudley, „Deep learning for healthcare: review, opportunities and challenges,“ *Briefings in bioinformatics*, kd. 19, nr 6, pp. 1236-1246, 2017.
- [14] J. Couto, „The major advancements in Deep Learning in 2018,“ *tryo labs*, 19 Detsember 2018. [Võrgumaterjal]. Available: <https://tryolabs.com/blog/2018/12/19/major-advancements-deep-learning-2018/>. [Kasutatud 17 April 2019].
- [15] B. Chandra ja R. K. Sharam, „Fast learning in Deep Neural Networks,“ *Neurocomputing*, kd. 171, nr 1, pp. 1205-1215, 2016.
- [16] I. Goodfellow, Y. Bengio ja A. Courville, *Deep Learning*, London, Cambridge: The MIT Press, 2016.

- [17] K. O'Shea ja R. Nash, „An Introduction to Convolutional Neural Networks,“ 26 November 2015. [Võrgumaterjal]. Available: <https://arxiv.org/pdf/1511.08458.pdf>. [Kasutatud 17 Aprill 2019].
- [18] E. A. Dominguez, „Estimación Precisa de la Orientación del Rostro Humano Utilizando Redes de Neuronas,“ Juuni 2017. [Võrgumaterjal]. Available: http://oa.upm.es/47220/1/TFG_ELIVIRA_AMADOR_DOMINGUEZ.pdf. [Kasutatud 20 April 2019].
- [19] J. Torres, „Convolutional Neural Networks for Beginners,“ Towards Data Science, 24 September 2018. [Võrgumaterjal]. Available: <https://towardsdatascience.com/convolutional-neural-networks-for-beginners-practical-guide-with-python-and-keras-dc688ea90dca>. [Kasutatud 21 Aprill 2019].
- [20] M. Nilsen, Neural networks and deep learning, 2015.
- [21] C. Huobolainen, „Masinõppe abil huumori tuvastamine karikatuuride ja portreede näitel,“ 07 Juuni 2018. [Võrgumaterjal]. Available: <https://digi.lib.ttu.ee/i/?10726>. [Kasutatud 17 Aprill 2019].
- [22] J. Wu, „Introduction to Convolutional Neural Networks,“ Mai 2017. [Võrgumaterjal]. Available: <https://pdfs.semanticscholar.org/450c/a19932fcef1ca6d0442cbf52fec38fb9d1e5.pdf>. [Kasutatud 22 Aprill 2019].
- [23] A. Deshpande, „A Beginner's Guide To Understanding Convolutional Neural Networks,“ 20 Juuli 2016. [Võrgumaterjal]. Available: <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>. [Kasutatud 26 Aprill 2019].
- [24] F. Luan, E. S. Sylvain Paris ja K. Bala, „Deep Photo Style Transfer,“ *CVPR*, pp. 4990-4998, 2017.
- [25] L. A. Gatys, A. S. Ecker ja M. Bethge, „A Neural Algorithm of Artistic Style,“ 26 August 2015. [Võrgumaterjal]. Available: <https://arxiv.org/pdf/1508.06576v1.pdf>. [Kasutatud 23 Aprill 2019].
- [26] R. Yuan, „Neural Style Transfer: Creating Art with Deep Learning using tf.keras and eager execution,“ medium, 3 August 2018. [Võrgumaterjal]. Available: <https://medium.com/tensorflow/neural-style-transfer-creating-art-with-deep-learning-using-tf-keras-and-eager-execution-7d541ac31398>. [Kasutatud 22 Aprill 2019].
- [27] L. A. Gatys, A. S. Ecker ja M. Bethge, „Image Style Transfer Using Convolutional Neural Networks,“ 2016. [Võrgumaterjal]. Available: http://openaccess.thecvf.com/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf. [Kasutatud 23 Aprill 2019].
- [28] Y. Li, N. Wang, J. Liu ja X. Hou, „Demystifying Neural Style Transfer,“ 1 Juuli 2017. [Võrgumaterjal]. Available: <https://arxiv.org/pdf/1701.01036.pdf>. [Kasutatud 23 Aprill 2019].
- [29] I. Goodfellow, J. Pouget-Abadie, B. X. D. W.-F. Mehdi Mirza, S. Ozair, A. Courville ja Y. Bengio, „Generative Adversarial Nets,“ 9 Juuni 2014. [Võrgumaterjal]. Available: <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>. [Kasutatud 25 April 2019].

- [30] C. Ledig, L. Theis, F. Huszar, J. Caballero ja A. Cunningham, „Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial,“ *CVPR*, pp. 4681-4690, 2017.
- [31] I. Goodfellow, „NIPS 2016 Tutorial: Generative Adversarial Networks,“ 3 Aprill 2017. [Võrgumaterjal]. Available: <https://arxiv.org/pdf/1701.00160.pdf>. [Kasutatud 24 Aprill 2019].
- [32] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng ja F.-Y. Wang, „Generative adversarial networks: introduction and outlook,“ *JOURNAL OF AUTOMATICA SINICA*, kd. IV, nr 4, pp. 588-598, 2017.
- [33] P.-L. Pröve, „An Introduction to different Types of Convolutions in Deep Learning,“ Towards Data Science, 22 Juuli 2017. [Võrgumaterjal]. Available: <https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d>. [Kasutatud 29 Aprill 2019].
- [34] A. Ng, „Strided Convolutions,“ coursera, [Võrgumaterjal]. Available: <https://www.coursera.org/lecture/convolutional-neural-networks/strided-convolutions-wfUhx>. [Kasutatud 25 aprill 2019].
- [35] A. Radford, „Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,“ 19 November 2015. [Võrgumaterjal]. Available: <https://arxiv.org/pdf/1511.06434.pdf>. [Kasutatud 25 Aprill 2019].
- [36] Y. Zhang, „Image Generation with DCGAN,“ timzhangyuxuan, 28 November 2016. [Võrgumaterjal]. Available: http://www.timzhangyuxuan.com/project_dcgan/. [Kasutatud 26 Aprill 2019].
- [37] R. Gandhi, „A Look at Gradient Descent and RMSprop Optimizers,“ Towards Data Science, 19 Juuni 2018. [Võrgumaterjal]. Available: <https://towardsdatascience.com/a-look-at-gradient-descent-and-rmsprop-optimizers-f77d483ef08b>. [Kasutatud 24 Aprill 2019].
- [38] V. Bushaev, „Understanding RMSprop— faster neural network learning,“ Towards Data Science, 2 September 2018. [Võrgumaterjal]. Available: <https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a>. [Kasutatud 2 Aprill 2019].
- [39] S. Ruder, „An overview of gradient descent optimization algorithms,“ 19 Jaanuar 2016. [Võrgumaterjal]. Available: <http://ruder.io/optimizing-gradient-descent/>. [Kasutatud 4 Mai 2019].
- [40] J. L. B. Diederik Kingman, „ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION,“ cORNE, 22 Detsember 2014. [Võrgumaterjal]. Available: <https://arxiv.org/pdf/1412.6980.pdf>. [Kasutatud 4 Mai 2019].
- [41] „About Phyton,“ phyton.org, [Võrgumaterjal]. Available: <https://www.python.org/about/>. [Kasutatud 26 Aprill 2019].
- [42] M. Abadi, M. Isard ja D. G. Murray, „A Computational Model for TensorFlow,“ 18 Juuni 2017. [Võrgumaterjal]. Available: http://delivery.acm.org/10.1145/3090000/3088527/pldiws17mapl-maplmainid2.pdf?ip=83.166.55.1&id=3088527&acc=OA&key=4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E5945DC2EABF3343C&__acm__=1557312131_ecdc962e4911b84428af6b893fd4b35b. [Kasutatud 26 April 2019].

- [43] G. Tanner, „Introduction to Deep Learning with Keras,“ Towards Data Science, [Võrgumaterjal]. Available: <https://towardsdatascience.com/introduction-to-deep-learning-with-keras-17c09e4f0eb2>. [Kasutatud 26 Aprill 2019].
- [44] „Caricatures,“ Pinterest, Aprill 2019. [Võrgumaterjal]. Available: <https://www.pinterest.co.uk/kerlyklee/caricatures/>. [Kasutatud 30 Aprill 2019].
- [45] A. Z. K. Simonyan, „Github Gist,“ Github, 2014. [Võrgumaterjal]. Available: <https://gist.github.com/baraldilorenzo/07d7802847aaad0a35d3>. [Kasutatud 1 Aprill 2019].
- [46] A. Z. Karen Simonyan, „Pretrained models,“ 10 April 2015. [Võrgumaterjal]. Available: <http://www.vlfeat.org/matconvnet/pretrained/>. [Kasutatud 2 Mai 2019].
- [47] H. Heidenreich, „Style Transfer Tutorial,“ 16 Aprill 2019. [Võrgumaterjal]. Available: <https://github.com/hunter-heidenreich/ML-Open-Source-Implementations/blob/master/Style-Transfer/Style%20Transfer.ipynb>. [Kasutatud 17 Aprill 2018].
- [48] „DonkeyHotey,“ flickr, 8 Jaanuar 2011. [Võrgumaterjal]. Available: <https://www.flickr.com/photos/donkeyhotey/5337333337/in/photostream/>. [Kasutatud 30 Aprill 2019].
- [49] P. Pobrotyn, „Neural style transfer in Tensorflow using a pretrained VGG-19,“ github, 14 November 2017. [Võrgumaterjal]. Available: <https://github.com/PrzemekPobrotyn/Neural-Style-Transfer>. [Kasutatud 29 April 2019].
- [50] Y. Steinbuch, „Kim Jong Un forgets to photoshop hist portrait,“ New York Post, 11 Mai 2016. [Võrgumaterjal]. Available: <https://nypost.com/2016/05/11/kim-jong-un-takes-a-nophotoshop-headshot/?fbclid=IwAR2blrgcDHwkgzwh54AQSuzULixfxazhyoi4pJ4RukiNxvKOgs7g-BDSzAU>. [Kasutatud 19 Aprill 2019].
- [51] S. Thompson, „Beyoncé's Netflix Documentary 'Homecoming' Provides A Blueprint For Reaching Multi-Cultural Customers,“ Forbes, 30 April 2019. [Võrgumaterjal]. Available: <https://www.forbes.com/sites/soniathompson/2019/04/30/beyonces-netflix-documentary-homecoming-provides-a-blueprint-for-reaching-multi-cultural-customers/#3aac1c941118>. [Kasutatud 2 Mai 2019].
- [52] R. S. Mas, „Caricatura de Beyoncé,“ Risasinmas, 1 Juuni 2014. [Võrgumaterjal]. Available: <http://www.risasinmas.com/caricatura-de-beyonce-6/>. [Kasutatud 25 April 2019].
- [53] L. M. Segarra, „'The Kind of Thing Dictators Do': Jimmy Kimmel Slams Trump for Firing James Comey,“ Time, 10 Mai 2017. [Võrgumaterjal]. Available: <http://time.com/4773602/jimmy-kimmel-comey-firing/>. [Kasutatud 16 Aprill 2019].
- [54] Y. Lori, „Jimmy Kimmel,“ Deviant art, Detsember 2014. [Võrgumaterjal]. Available: <https://www.deviantart.com/yoannlori/art/Jimmy-Kimmel-499608296>. [Kasutatud 16 Aprill 2019].
- [55] J. C. Ryan, „Jennifer Lawrence is engaged to boyfriend Cooke Maroney,“ The Press Democrat, 6 Veebruar 2019. [Võrgumaterjal]. Available: <https://www.pressdemocrat.com/entertainment/9253785-181/jennifer-lawrence-is-engaged-to?sba=AAS>. [Kasutatud 16 Aprill 2019].

- [56] N. Davies, „Caricatures, portraits and illustrations,“ Pinterest, [Võrgumaterjal]. Available: <https://www.pinterest.com/pin/528469337500029232/>. [Kasutatud 16 Aprill 2019].
- [57] S. Raval, „Pokemon_GAN,“ GitHub, 27 Juuni 2018. [Võrgumaterjal]. Available: https://github.com/lISourcell/Pokemon_GAN. [Kasutatud 24 Aprill 2019].
- [58] „tf.train.GradientDescentOptimizer,“ TensorFlow, [Võrgumaterjal]. Available: https://www.tensorflow.org/api_docs/python/tf/train/GradientDescentOptimizer. [Kasutatud 20 April 2019].
- [59] „tf.train.RMSPropOptimizer,“ TensorFlow, [Võrgumaterjal]. Available: https://www.tensorflow.org/api_docs/python/tf/train/RMSPropOptimizer. [Kasutatud 27 Aprill 2019].
- [60] „tf.train.AdamOptimizer,“ TensorFlow, [Võrgumaterjal]. Available: https://www.tensorflow.org/api_docs/python/tf/train/AdamOptimizer. [Kasutatud 1 Mai 2019].
- [61] C. Freeland, „Portraits in painting and photography,“ 24 May 2007. [Võrgumaterjal]. Available: <https://link.springer.com/content/pdf/10.1007%2Fs11098-007-9099-7.pdf>. [Kasutatud 2 April 2019].

Lisa 1- NÄRVIVÖRGU STÍLIÜLEKANDE ANDMESTIK



Lisa 2- Genereerivate vastandvõrkude andmestik





