

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Tarkvarateaduse instituut

Mark Grabovski

155546IAPB

**Alavalgustatud piltide valgustamine generatiivsete  
võistlusvõrkude abil**

Bakalaureusetöö

Juhendaja: Priit Järv  
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: Mark Grabovski

Kuupäev: 21.05.2019

## **Annotatsioon**

Käesoleva bakalaureusetöö eesmärgiks on hinnata kui hästi suudavad generatiivsed võistlusvõrgud alavalgustatud pilte valgustada ning võrrelda neid muude traditsiooniliste digitaalsete pildi taastamismeetoditega.

Töös kirjeldatakse ja võrreldakse erinevaid generatiivsete võistlusvõrkude arhitektuure ja nende tulemusi. Generatiivsete võistlusvõrkude treenimiseks ja testimiseks luuakse andmestikud, mis koosnevad alavalgustatud ja valgustatud piltidest. Saadud tulemusi võrreldakse omavahel. Töö tulemusi hinnatakse kvantitatiivselt ja kvalitatiivselt ning võrreldakse traditsiooniliste digitaalsete pildi taastamismeetoditega. Kvantitatiivses hinnangus mõõdeti piltide erinevust hästi valgustatud originaal piltidest. Kvalitatiivse hinnangu jaoks koostati küsimustik, kus küsitletavavad võrdlesid omavahel generatiivsete võistlusvõrkude poolt loodud pilte traditsiooniliste pildi taastamismeetodiga saadud piltidega.

Tulemuseks on generatiivsete võistlusvõrkude poolt valgustatud pildid ning nende võrdluse käigus tehtud hinnangud.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 45 leheküljel, 5 peatükki, 35 joonist.

## **Abstract**

The goal of this thesis is to evaluate how well will generative adversarial networks brighten low light images and compare them to other traditional digital image restoration methods.

This work describes and compares different architectures of generative adversarial networks and their results. A data set of combined low light and well lit images is created in order to train and test generative adversarial networks. The results are then evaluated quantitatively and qualitatively, and compared to traditional digital image restoration methods. Quantitative evaluation is done by calculation how different are results from original well lit images. Qualitative evaluation is done by questionnaire, where participants compare images produced by generative adversarial networks to traditional image restoration methods.

Result of this work are images that are produced by generative adversarial networks and their evaluation.

The thesis is in Estonian and contains 45 pages of text, 5 chapters, 35 figures.

## Lühendite ja mõistete sõnastik

|         |  |
|---------|--|
| CNN     | Konvolutsiooniline närvivõrk                               |
| GAN     | Generatiivne võistlusvõrk                                  |
| CGAN    | Tinglik generatiivne võistlusvõrk                          |
| DCGAN   | Sügav-konvolutsiooniline generatiivne võistlusvõrk         |
| C-DCGAN | Tinglik sügav-konvolutsiooniline generatiivne võistlusvõrk |
| SRGAN   | Super-resolutsiooni generatiivne võistlusvõrk              |

# Sisukord

|       |  |    |
|-------|--|----|
| 1.    | Sissejuhatus .....   | 10 |
| 2.    | Tehisnärvivõrgud .....   | 11 |
| 3.    | Sügavaõpe .....  | 12 |
| 3.1   | Sügavad närvivõrgud .....  | 12 |
| 3.2   | Konvolutsioonilised närvivõrgud .....                              | 12 |
| 3.3   | Generatiivsed võistlusvõrgud .....                                 | 14 |
| 3.3.1 | GAN'ide treenimine .....   | 15 |
| 3.3.2 | CGAN.....  | 16 |
| 3.3.3 | DCGAN.....   | 16 |
| 3.3.4 | SRGAN .....  | 17 |
| 4.    | Kasutatud tehnoloogiad .....                                       | 18 |
| 4.1   | Pix2pix.....   | 18 |
| 4.2   | TensorFlow .....   | 18 |
| 4.3   | GIMP .....   | 18 |
| 4.4   | Python.....  | 18 |
| 5.    | Katsetused.....  | 20 |
| 5.1   | Andmestik .....  | 20 |
| 5.2   | CGAN.....  | 21 |
| 5.2.1 | Mudeli treenimine ja testimine 1 000 pildiga ja 100 epohhiga ..... | 22 |
| 5.2.2 | Mudeli treenimine ja testimine 2 000 pildiga ja 200 epohhiga ..... | 25 |
| 5.2.3 | Mudeli treenimine ja testimine 4 000 pildiga ja 200 epohhiga ..... | 28 |
| 5.2.4 | Järeldused .....   | 30 |
| 5.3   | C-DCGAN.....   | 30 |
| 5.3.1 | Mudeli treenimine ja testimine 4 000 pildiga ja 50 epohhiga .....  | 31 |
| 5.3.2 | Järeldused .....   | 34 |
| 5.4   | SRGAN .....  | 34 |

|  |    |
|--|----|
| 5.4 Kvantitatiivne hindamine .....                                       | 35 |
| 5.4.2 CGAN'i 4 000 ja 2 000 pildiga treenitud mudelite võrdlus .....     | 36 |
| 5.4.3 C-DCGAN'i ja CGAN'i mudelite võrdlus.....                          | 36 |
| 5.4.4 C-DCGAN'i võrdlus traditsiooniliste pilditaastamise meetodiga..... | 38 |
| 5.4.5 Järeldused .....   | 38 |
| 5.5 Kvalitatiivne hindamine .....  | 39 |
| Kokkuvõte .....  | 41 |
| Kasutatud kirjandus .....  | 42 |
| Lisa 1. Kvalitatiivse küsimustiku näide.....                             | 45 |

## Jooniste loetelu

|  |    |
|--|----|
| Joonis 1. Tehisnärvivõrgu arhitektuur [3].   | 11 |
| Joonis 2. Konvolutsioonilise närvivõrgu arhitektuur [14].  | 13 |
| Joonis 3. Värvilise pildi kolmemõõtmeline maatriks [15].   | 13 |
| Joonis 4. Pildi maatriksi töötlemine filtri poolt, kus oranž maatriks on pildi pikslite maatriks, roheline on filtri maatriks ja kollane tulemused [17]. | 14 |
| Joonis 5. GAN'i arhitektuur [21].  | 15 |
| Joonis 6. CGAN arhitektuur.  | 16 |
| Joonis 7. DCGAN'i generaatori võrgudisain [24].  | 17 |
| Joonis 8. SRGAN'i generaatori ja diskriminaatori võrgu arhitektuurid [25].   | 17 |
| Joonis 9. Piltide võrdlus  | 20 |
| Joonis 10. Näide treening- või testandmestikus olevast pildist.  | 21 |
| Joonis 11. Generaatori kaod treenimise käigus.   | 22 |
| Joonis 12. Diskriminaatori kaod treenimise käigus.   | 23 |
| Joonis 13. Alavalgustatud sisendpildist (vasak) genereeritud väljundpilt (parem).  | 23 |
| Joonis 14. CGAN'i poolt genereeritud pildi (vasak) võrdlus originaalse hästi valgustatud pildiga (parem)   | 24 |
| Joonis 15. Alavalgustatud sisendpildist (vasak) genereeritud väljundpilt (parem).  | 24 |
| Joonis 16. CGAN'i poolt genereeritud pildi (vasak) võrdlus originaalse hästi valgustatud pildiga (parem).  | 25 |
| Joonis 17. Generaatori kaod treenimise käigus.   | 26 |
| Joonis 18. Diskriminaatori kaod treenimise käigus.   | 26 |
| Joonis 19. Alavalgustatud sisendpildist (vasak) genereeritud väljundpilt (parem).  | 27 |
| Joonis 20. Alavalgustatud sisendpildist (vasak) genereeritud väljundpilt (parem).  | 27 |
| Joonis 21. Generaatori kaod treenimise käigus.   | 28 |
| Joonis 22. Diskriminaatori kaod treenimise käigus.   | 28 |
| Joonis 23. Eelneva mudeli (vasak) ja selle mudeli (parem) väljund pildid.  | 29 |
| Joonis 24. Eelneva mudeli (vasak) ja selle mudeli (parem) väljund pildid.  | 29 |
| Joonis 25. Generaatori kaod treenimise käigus.   | 31 |
| Joonis 26. Diskriminaatori kaod treenimise käigus.   | 32 |
| Joonis 27. Alavalgustatud sisendpildist (vasak) genereeritud väljundpilt (parem).  | 32 |
| Joonis 28. Alavalgustatud sisendpildist (vasak) genereeritud väljundpilt (parem).  | 33 |



|  |    |
|--|----|
| Joonis 29. C-DCGAN'i poolt genereeritud pildi (vasak) võrdlus originaalse hästi valgustatud pildiga (parem). ..... | 33 |
| Joonis 30. CGAN'i pildi (vasak) töötlemine SRGAN'iga (parem). .....  | 34 |
| Joonis 31. C-DCGAN'i pildi (vasak) töötlemine SRGAN'iga (parem).....   | 35 |
| Joonis 32. CGAN'i tulemuse (vasak) võrdlus DCGAN'i omaga (parem). .....  | 37 |
| Joonis 33. CGAN'i tulemuse (vasak) võrdlus DCGAN'i omaga (parem). .....  | 37 |
| Joonis 34. DCGAN'i tulemuse (vasak) võrdlus Fotor'i omaga (parem).....   | 38 |
| Joonis 35. Kvalitatiivse hinnangu küsitluse tulemused. ....  | 39 |

# 1. Sissejuhatus

Paljudel fotokaameratel esineb probleem pildistamisega väheselt valgustatud keskkonnas. Sellistel alavalgustatud keskkonnas tehtud fotodel olevad objektid on müra ja piisava valguse puudumise tõttu ebaselged ja raskesti eristatavad, mistõttu esineb tehtud piltidel informatsiooni kadu.

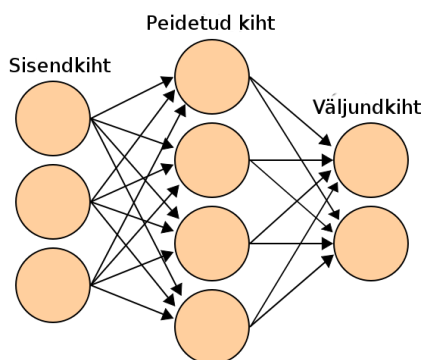
Antud bakalaureusetöö eesmärgiks on analüüsida ja hinnata, kui hästi suudavad generatiivsed võistlusvõrgud (lühidalt GAN'id) alavalgustatud fotosid valgustada ning võrrelda neid muude traditsiooniliste digitaalsete taastamismeetoditega. Selleks treenitakse generatiivseid võistlusvõrke nii, et need oleksid suutelised alavalgustatud sisendpiltidest genereerida sellised väljundpildid, mis oleksid võimalikult selged, hästi valgustatud ning minimaalse müra ja info kaoga. Valgustatud piltide genereerimiseks on töös kasutatud kahte erinevat GAN arhitektuuri – cGAN ja cDCGAN, mis on kirjutatud Pythonis kasutades TensorFlow tarkvarateeki.

Töö käigus analüüsitakse cGANi ja cDCGAN'i arhitektuuride ja tulemuste erinevusi. CGAN'i ja c-DCGAN'i poolt genereeritud tulemusi püütakse parandada SRGAN'i abil, et tõsta pildi kvaliteeti. Lõpptulemusi hinnatakse kvantitatiivselt ja kvalitatiivselt ning võrreldakse Fotor pilditöötlusprogrammis valgustatud piltidega. Kvantitatiivseks hindamiseks arvutatakse välja GAN'ide ja Fotori poolt valgustatud piltide maksimaalne signaali-müra suhe (PSNR), keskmine ruutviga (MSE) ja struktuurne sarnasus (SSIM) ning võrreldakse tulemusi omavahel. Kvalitatiivne hindamine toimub küsimustiku alusel, kus küsitletavad võrdlevad generatiivsete võistlusvõrkude poolt genereeritud lõpptulemusi Fotor'is kunstlikult valgustatud piltidega.

## 2. Tehisnärvivõrgud

Tehisnärvivõrk või lihtsalt närvivõrk (*artificial neural network, neural network*) on algoritmide kogum, mis on loodud mustrite ära tundmiseks. Mustriteks võivad olla pildid, helid, tekstid ja muu andmed, mis on teisendatud numbriteks. Saadud numbrid hoitakse vektorites. Tehisnärvivõrgud on inspiratsiooni saanud bioloogilisest inimajust, kuna nad püüavad imiteerida seda kuidas inimese aju õpib ja informatsiooni töötleb. [1, 2]

Närvivõrgud koosnevad mitmest närvist ehk neuronist, millest igaüks saab eelnevalt neuronilt sisendid, seejärel viib läbi arvutuse ning annab väljundi. Neuronite vahel on ühendused. Igal ühendusel on kaal, millest sõltub kui palju ühe neuroni väljund teise neuroni väljundit mõjutada saab. Neuronid on jagatud kihtidesse. Kõige esimene kiht on sisendkiht, kuna siia sisestatakse andmed. Järgmised kihid on peidetud kihid, mis on enamasti täielikult ühendatud võrku. Viimane kiht on väljundkiht, see annab välja informatsiooni, mida sisendandmete kohta sooviti teada.[2, 3] Närvivõrgu arhitektuuri ehitus on kujutatud joonisel 1.



Joonis 1. Tehisnärvivõrgu arhitektuur [3].

Kuigi tehisnärvivõrgud on kasutusel alates 1943ndast aastast, on nad alles peale 1980ndaid aastaid muutunud tehisintellekti põhiliseks osaks. Selle põhjuseks on tagasilevi kasutuselevõtt.[2] Tagasilevi (*back-propagation*) on optimeerimise algoritm, mis aitab närvivõrkudel õppida. See täidab närvivõrgus sõnumitooja rolli, aidates võrgul reguleerida neuronite varjatuid kihte siis kui tulemus ei ole sobilik, nt närvivõrk, mis on õpetatud tuvastama piltidelt koeri, tuvastab valesti pildid oleva kassi koeraks. [4, 5]

### 3. Sügavaõpe

Sügavõpe või hierarhiline õpe (*deep learning, hierarchical learning*) on algoritmide kogumist koosnev masinõppe alamliik, mis püüab modelleerida kõrge-tasemelisi abstraktsioone andmetes [6]. Üheks sügavõppe aspektiks on tunnusehõive, kus algoritmid konstrueerivad sisendandmete põhjal kasulikke tunnuseid selleks, et nende põhjal õppida ja treenida [7]. Sügavõppe võrgud suudavad õppida ilma juhendamiseta andmetelt, mis on struktureerimata ja märgendamata [8]. Juhendamata õpe (*unsupervised learning*) tähendab seda, et inimesel pole vaja masina treenimisel või õpetamisel andmeid märgendada ega klassifitseerida, seda teeb masin ise. Sügavõppega on seotud mitmeid arhitektuure nagu sügavad närvivõrgud ja rekurrentsed närvivõrgud. Sügavõpet kasutatakse loomuliku keele töötlustes, arvutinägemises, kõnetuvastamises, bioinformaatikas ja muudes valdkondades. [9]

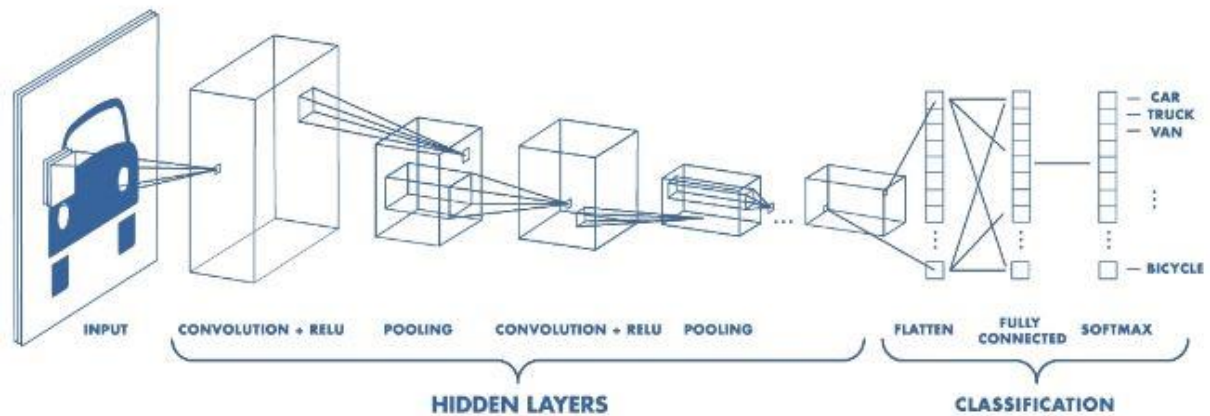
#### 3.1 Sügavad närvivõrgud

Sügavad närvivõrgud (*deep neural networks*) nagu ka muud närvivõrgud on kogum algoritme, mis on loodud mustrite ära tundmiseks. Võrreldes teiste ühekihiliste närvivõrkudega on sügavad närvivõrgud keerulisema ehitusega, sest koosnevad mitmest peidetud kihist [1]. Sõna „sügav“ on tehniline termin ning viitab kihtide arvule närvivõrgus ning põhimõtteliselt ei erine see muudest närvivõrkudest. Suurema kihtide arvuga võrke on intensiivsem treenida kui väikese arvu kihtidega võrke, kuna need sisaldavad rohkem matemaatilisi operatsioone, mida sisendandmed läbima peavad. Selline andmete töötlemise intensiivsus on sügavõppe iseloomulik joon, mistõttu on sügavõppe mudelite treenimistel tekkinud nõudlus videokaartidele järele. [10]

#### 3.2 Konvolutsioonilised närvivõrgud

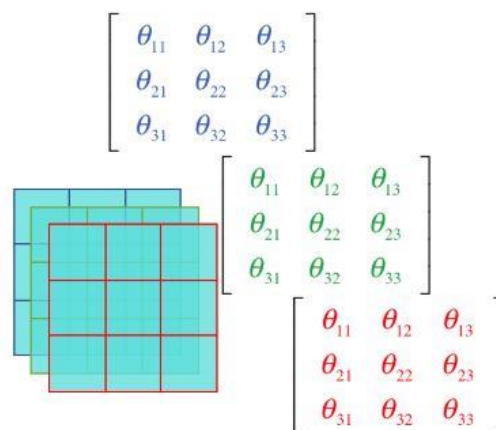
Konvolutsioonilised närvivõrgud (*convolutional neural networks, CNNs*) on sügavad närvivõrgud, mida kasutatakse peamiselt piltide klassifitseerimiseks, klasterdamiseks ja piltidelt objektide ära tundmiseks [11, 12]. Nagu ka muud närvivõrgud koosnevad ka konvolutsioonilised närvivõrgud sisend kihist, peidetud kihtidest ja väljundkihist. Konvolutsioonilise närvivõrgu peidetud kihid sisaldavad konvolutsioonilisi kihte (*convolutional layers*), kust närvivõrk saabki oma nimetuse. Lisaks võib närvivõrk sisaldada ahenduskihte (*pooling layers*) ja mittenegatiivsed lineaarfunktsioone (*rectified linear units, ReLu*). Üheks CNN'i eripäraks on see, et selle neuronid pole täielikult ühendatud st seda, et

neuron, mis asub ühes kihis pole ühendatud kõikide järgmises kihis olevate neuronitega. [12] Iga konvolutsiooniline kiht sisaldab mingi arv filtreid, mis suudavad sisendpiltidelt mustreid ära tunda. Mida sügavamal võrgus asuvad kihid, seda keerulisemaid mustreid suudavad filtrid ära tunda, alustades kergete geomeetriliste kujudega ja lõpetades keerulistema objektidega nagu auto, inimene, loom jne. [13] Järgneval joonisel on illustreeritud konvolutsioonilise närvivõrgu arhitektuur.



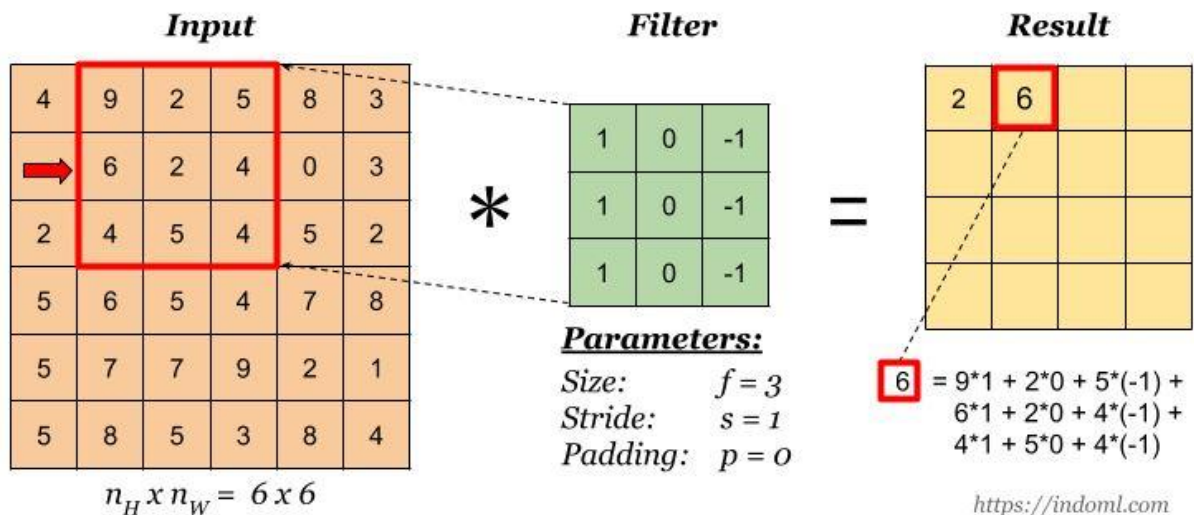
Joonis 2. Konvolutsioonilise närvivõrgu arhitektuur [14].

Konvolutsioonilised närvivõrgud tajuvad pilte kahe- või kolmemõõtmeliste maatriksitena. Reeglina on halltoon pildid kahemõõtmelised maatriksid ja värvilised pildid kolmemõõtmelised maatriksid. Esimesed kaks dimensiooni tähistavad pildi pikkust ja laiust pikslites, ja kolmas dimensioon tähistab pildi värvi. Kolmas dimensioon on sügavusega 3, sest koosneb kolmest värvi kanalist – punane, roheline, sinine (*red-green-blue, RGB*). [15] Järgneval joonisel on kujutatud värvilise pildi kolmemõõtmeline maatriks.



Joonis 3. Värvilise pildi kolmemõõtmeline maatriks [15].

Selle asemel, et sisendpilti ühe piksli kaupa töödelda, nagu teevad seda mõned närvivõrgud, töötleb konvolutsiooniline närvivõrk piksleid maatriksite kaupa, lastes nad läbi närvivõrgu kihis oleva filtri. Filter koosneb maatriksist, mis on väiksem kui pilt ise. Sellist filtrit nimetatakse ka kerneliks või tunnuste detektoriks. Kernel töötleb pildi piksleid liikudes paremalt vasakule ja ülevalt alla korrutades omavahel vastavate pikslite väärtused ja liites tulemused kokku (joonis 4). [11, 16]



Joonis 4. Pildi maatriksi töötlemine filtri poolt, kus oranž maatriks on pildi pikslite maatriks, roheline on filtri maatriks ja kollane tulemused [17].

### 3.3 Generatiivsed võistlusvõrgud

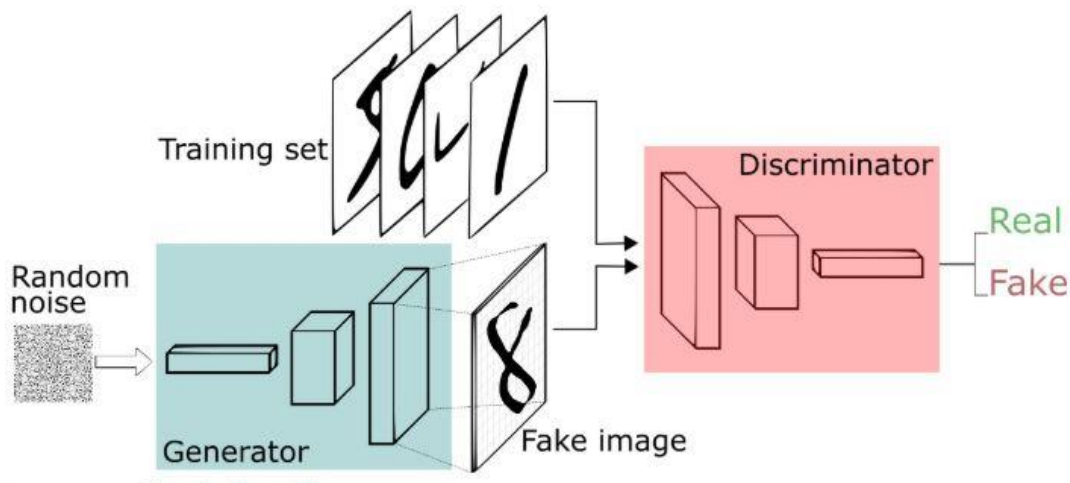
Generatiivsed võistlusvõrgud (*generative adversarial networks, GANs*) on sügava närvivõrgu arhitektuurid, mis koosnevad kahest omavahel võistlevast närvivõrgust [18]. Tegemist on masinõpe tehnikaga, mis oli väljatöötatud Ian Goodfellow ja teiste uurijate – Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio – poolt aastal 2014 Montreal ülikoolis. GAN põhineb minimax mängul, kus üks võrk püüab teist võita. Neid kahte omavahel võistlevaid närvivõrke nimetatakse diskriminaatoriks (*discriminator, discriminator network*) ja generaatoriks (*generator, generator network*). Nii diskriminaator kui ka generaator on mitmekihilised tsükliteta närvivõrgud (*multilayer perceptron*). [19] Selleks, et GAN'e nagu ka muid närvivõrke saaks treenida on vaja treeningandmestiku. Treeningandmestik on kogus digitaalseid andmeid, mida kasutatakse algoritmide treenimiseks, et algoritmid oskaksid rakendada selliseid kontsepte nagu

närvivõrgud nii, et need oleksid suuteliselt õppima ja tekitama tulemusi [20]. Sellisteks andmeteks võivad olla pildid, tekstid, videod, helid ja muud andmed.

Eksisteerib mitmeid erinevaid GAN'i tüüpe nagu tavaline GAN (*vanilla GAN*), cGAN, DCGAN, SRGAN jt.

### 3.3.1 GAN'ide treenimine

Generatiivsete võistlusvõrkude treenimise protsess toimub järgmiselt: generaator loob uusi väljundandmeid võttes sisendiks juhusliku müra ja diskriminaator otsustab, kas generaatori väljundid kuuluvad treeningandmestikku või on generaatori poolt tekitatud, seejärel klassifitseerib diskriminaator generaatori väljundid kas autentseks või võltsiks. Generaatori ülesandeks on petta diskriminaatorit ja genereerida sellised pildid, mida diskriminaator klassifitseeriks autentseks. Diskriminaatori ülesandeks on tuvastada generaatori poolt genereeritud pildid ja klassifitseerida nad võltsiks. Diskriminaatori väljundiks on arvud vahemikus 0 kuni 1, mis vastavad tõenäosustele. Nende tõenäosuste abil saab diskriminaator generaatori tulemusi liigitada kas autentseks, kui tõenäosus on lähemal 1-le või võltsiks, kui tulemus on lähemal 0-le. [18] Joonisel 5 on kujutatud GAN'i arhitektuur.



Joonis 5. GAN'i arhitektuur [21].

Mõlemad võrgud on dünaamilised st treenimise käigus püüab iga võrk oma ülesannet paremini täita. Selleks, et üks võrk teist alla ei suruks, peavad generaatori ja diskriminaatori võrgud tasakaalus olema. Juhul kui võrgud pole tasakaalus siis on raske GAN'i treenida, sest kumbi võrk ei õpi efektiivselt. [18]

### 3.3.2 CGAN

CGAN (*conditional-GAN*) on tinglik generatiivne võistlusvõrk. Kuna tavalises GAN'is pole kontrolli selle üle missuguseid andmeid luuakse siis erineb tinglik GAN selle poolest, et selle generaatori ja diskriminaatori sisendile on lisatud parameeter  $y$ , mis võib tähistada märgendit (*label*) või mõnda muud abistavat infot (joonis 6). Selline meetod aitab juhtida andmete genereerimise protsessi, et saavutada soovitava väljundi. [22]



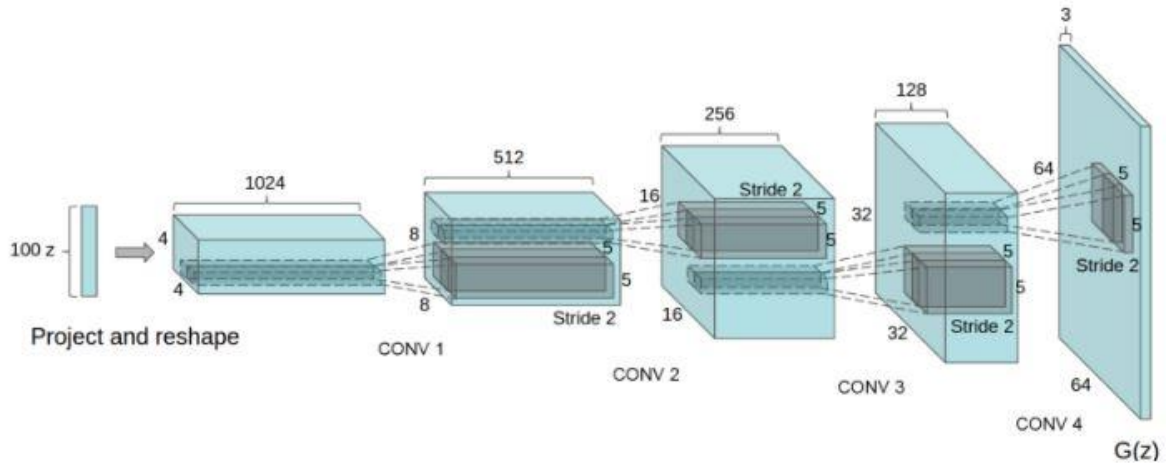
Joonis 6. CGAN arhitektuur.

CGAN leiab peamiselt rakendust andmete muutmises (nt genereerides sisendpildist soovitud väljundpilt: talvest suve, fotost maali, must-valge pilt värviliseks jne) või teisendamises ühest liigist teise (nt genereerides tekstist pildi). Tingliku GAN'i saab rakendada DCGAN'le ja ka teiste generatiivsete võistlusvõrkude tüüpidele. [23]

### 3.3.3 DCGAN

Sügav konvolutsiooniline generatiivne võistlusvõrk ehk DCGAN (*deep convolutional GAN*) on konvolutsiooniliste närvivõrkude klass. Võrreldes tavaliste GAN'idega on DCGAN'id stabiilsemad ning loovad paremaid tulemusi. DCGAN'i arhitektuur koosneb peamiselt konvolutsioonilistest närvivõrkudest. Erinevalt tavalisest GAN'ist ei ole DCGAN'i kihid täielikult ühendatud. Sügava konvolutsioonilise GAN'i generaatoris ja diskriminaatoris kasutatakse plokk normaliseerimist (*batch normalization*). Selle generaator sisaldab mittenegatiivseid lineaarfunktsioone (*rectified linear unit, ReLU*) kõikides oma kihtides, välja arvatud väljundkihis. Diskriminaator sisaldab lekkivaid mittenegatiivseid lineaar-funktsioone kõikides oma kihtides. [24] Joonisel 7 on kujutatud DCGAN'i generaatori võrgudisaini.

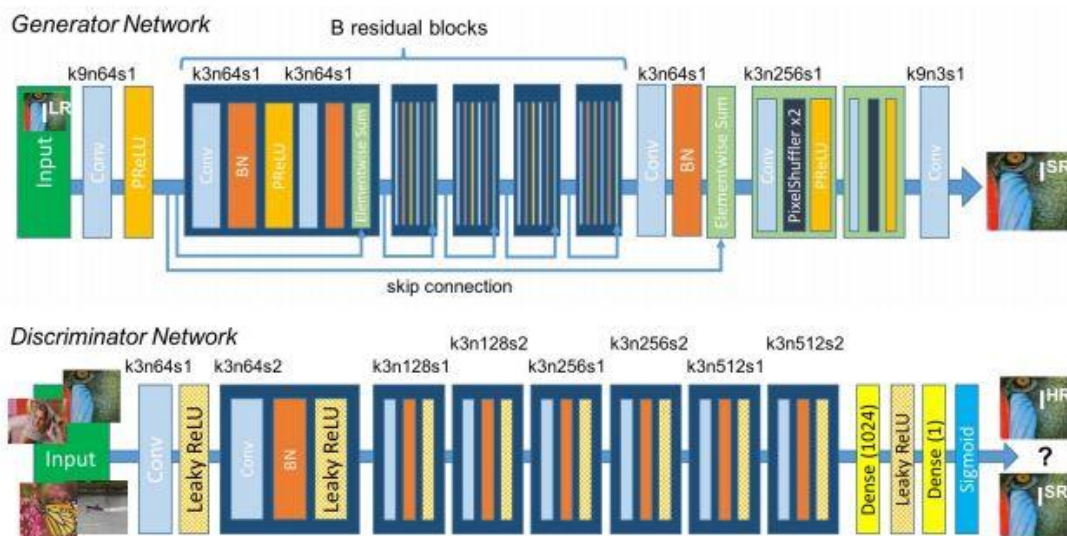




Joonis 7. DCGAN'i generaatori võrgudisain [24].

### 3.3.4 SRGAN

SRGAN'iks nimetatakse super-resolutsiooni generatiivset võistlusvõrku (*super-resolution GAN*). SRGAN'i ülesandeks on genereerida madala resolutsiooniga piltidest kõrge resolutsiooniga pildid. SRGAN'i arhitektuur põhineb VGG'il (*visual geometry group*) ja kasutab sügavat jääkvõrku (ResNet). Treenimise protsessis muudetakse kõrge resolutsiooniga pildid madala resolutsiooniga piltideks. Generaator genereerib madala resolutsiooniga piltidest super-resolutsioon pildid ning diskriminaator eristab kõrge-resolutsiooniga pilte. Mõlemad võrgud (generaator ja diskriminaator) koosnevad peamiselt konvolutsioonilistest kihtidest. [25] Joonisel 8 on kujutatud SRGAN'i generaatori ja diskriminaatori võrgu arhitektuurid.



Joonis 8. SRGAN'i generaatori ja diskriminaatori võrgu arhitektuurid [25].

## 4. Kasutatud tehnoloogiad

Antud bakalaureusetöös kasutati piltide valgustamiseks tinglike generatiivseid võistlusvõrke, mis põhinevad pix2pix projektil [26]. Mõlemad selles töös kasutatavad GAN'id (c-DCGAN ja cGAN) kasutavad TensorFlow'is implementeeritud pix2pix koodi, mis on kirjutatud Python programmeerimiskeeles [27, 28]. Generatiivsete võistlusvõrkude treenimiseks ja testimiseks vajaminev andmestik on koostatud autori poolt alla laetud piltidest ja tehtud fotodest. CGAN'i ja c-DCGAN'i tulemustele rakendati lisaks SRGAN'i, et proovida parandada genereeritud piltide kvaliteeti.

### 4.1 Pix2pix

Pix2pix on Pythonis kirjutatud tarkvara, mis muudab sisendpildi vastavaks väljundpildiks. Selleks kasutatakse pix2pix koodis tinglike generatiivseid võistlusvõrke, mis võimaldavad ühest pildist luua teise. Originaalses pix2pix projektis olid GAN'id implementeeritud kasutades Torch ja PyTorch tarkvarateeke, mis on ümber porditud ka järgnevasse teekidesse: TensorFlow, Keras ja Chainer. [26, 29]

### 4.2 TensorFlow

TensorFlow on avatud lähtekoodiga masinõppe tarkvarateek, mis teeb numbrilisi arvutusi kasutades andmevoogude graafe. Tarkvarateek oli välja töötatud Google'i poolt aastal 2015 ja seda kasutatakse peamiselt sügavõppes ja närvivõrkudes. TensorFlow töötab nii arvutiprotsessorite (CPU) kui ka videokaartide (GPU) peal [30]. TensorFlow abil on võimalik visualiseerida GAN'i diskriminaatori ja generaatori kadusid graafikutena.

### 4.3 GIMP

GIMP (*GNU image manipulation program*) on tasuta avatud lähtekoodiga pilditöötlusprogramm. Antud töös kasutasin GIMP'i piltide töötlemiseks treeningandmestiku jaoks. Selleks tuli kirjutada pilditöötlusprogrammile GIMP-Python plugin.

### 4.4 Python

Python on interpreteeritud, objektorienteeritud, kõrgetasemeline programmeerimiskeel koos dünaamiliste semantikatega. Pythoni süntaks on lihtne ja kergesti õpitav, mis teeb koodi

loetavaks ja seega vähendab ka programmi hoolduskulusid [31]. Python toetab ka paljusid tarkvarateeke, mis koos programmeerimiskeele lihtsusega teeb selle populaarseks valikuks masinõppes.

## 5. Katsetused

Antud peatükis võrreldakse erinevaid treenitud GAN'i mudelite tulemusi. Seejärel rakendatakse cGAN'i ja c-DCGAN'i tulemustele SRGAN, et proovida parandada piltide kvaliteeti. Lõpuks hinnatakse pilte kvantitatiivselt ja kvalitatiivselt võrreldes GAN'ide tulemusi traditsiooniliste digitaalsete piltide taastamismeetodi tulemustega.

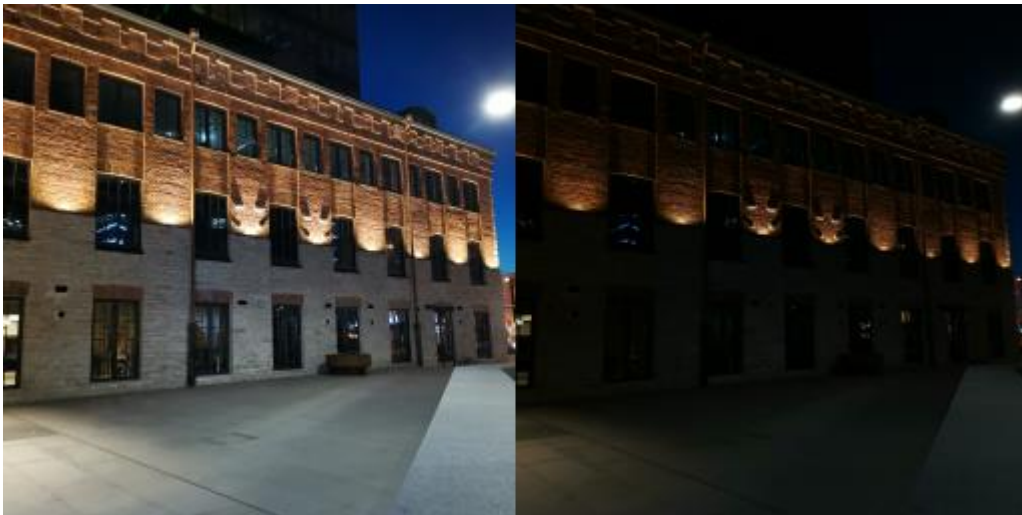
### 5.1 Andmestik

Generatiivsete võistlusvõrkude treenimiseks ja testimiseks on koostatud andmestik öisel ajal tehtud piltidest. Lõplik treeningandmestik koosneb 4 000 alla laetud ja töödeldud pildist. Testimiseks on fotokaameraga tehtud igast konkreetsest asukohast kaks fotot – hästi valgustatud ja alavalgustatud. Treeningandmestikus olevad pildid koosnevad kahest 256x256 resolutsiooniga kõrvuti olevast pildist. Esimene pilt on originaalne ehk pildistatud ja teine GIMP pilditöötlusprogrammiga kunstlikult alavalgustatud. Piltide töötlemiseks on kirjutatud GIMP'ile plugin Pythonis. Plugini koodis püütakse jäljendada seda, kuidas peaks alavalgustatud pilt välja nägema. Selleks võrreldakse fotokaameraga tehtud kaht fotot ühest ja samast asukohast kuid erineva valgustusega. Järgmistel joonistel on näha fotode erinevusi.



Joonis 9. Piltide võrdlus

Treeningandmestikku kasutatakse GAN'ide treenimiseks ja testandmestiku GAN'ide testimiseks peale treenimist. Sarnaselt treeningandmestikule koosneb ka testandmestik kahest 256x256 formaadiga kõrvuti olevast pildist. Mõlemad pildid on originaalsed fotod, aga erinevalt valgustatud. Järgneval joonisel on toodud näide andmestikudes olevast pildist, kus vasakul olev pilt on valgustatud (pilt, mida soovitakse saavutada ehk märgend (*label*)) ja paremal alavalgustatud (sisendpilt, mida GAN'ide generaator püüab valgustada).



Joonis 10. Näide treening- või testandmestikus olevast pildist.

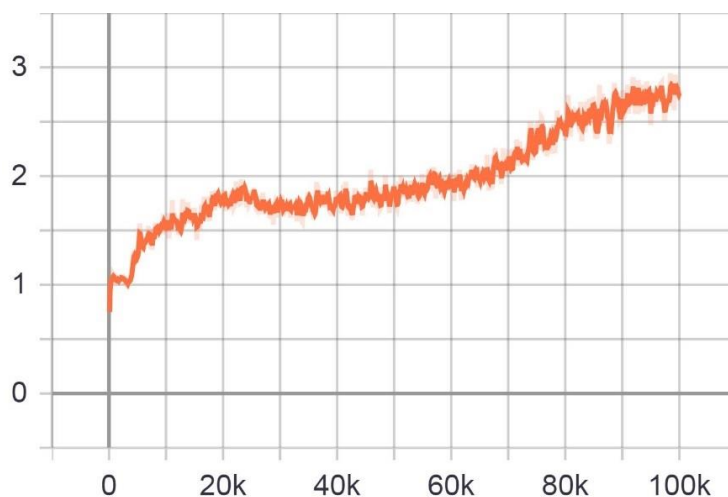
## 5.2 CGAN

Antud töös kasutatud pix2pix CGAN õpib teisendama sisendpildi väljundpildiks. Selle generaator rakendab muudatusi sisendpildile, et saada väljundpildi ja diskriminaator võrdleb sisendpilti tundmatu pildiga, mis on kas pilt andmestikust või väljundpilt generaatorilt. Seejärel oletab diskriminaator kas pilt on generaatori poolt genereeritud või mitte [32]. Generaatori struktuuri nimetatakse *encoder-decoder* (kooder-dekooder). Generaator muudab 256x256x3 sisendpildi maatriksi, mis on kõrgusega ja laiusega 256 pikslit ning koosneb kolmest värvi kanalist (punane-roheline-sinine), 1x1x512 maatriksiks. Sellisele kujule kokku surutud pilt saadakse mitme kooderi abil. Pildi kokku surumise protsessi otstarve on selles, et saada kõrgetasemelisema andmete esituse peale kooderi viimast kihti. Dekooderi kihtide protsess on vastupidine kooderi omadele st dekooder teisendab andmed tagasi esialgsele kujule.

CGAN'i treenimiseks ja testimiseks ettenähtu andmestikud on jagatud vastavatesse kaustadesse train ja test. Andmestikkude jaoks oli vaja kõik pildid teisendada 256x256 formaati, mida saab teha pix2pix programmis oleva lisa koodiga. Pildid teisendati 256x256 formaati ja pandi kokku kasutades TensorFlow tfimage moodulit ja NumPy teeki. Alguses teeb cGAN läbi treenimise protsessi, mille ajaline kulu sõltub piltide resolutsioonist, nende arvust treeningandmestikus (*batch size*) ja epochide arvust. Epoch on kasutaja poolt määratud arv, mis näitab mitu korda treenitakse antud andmestiku peal närvivõrke. Treenimise käigus salvestatakse arvutisse treenimise tulemused ja logi koos TensorFlow graafikuga, kus on kujutatud generaatori ja diskriminaatori treenimise protsessi käigus muutunud kadusid. Pärast cGAN'i treenimist on võimalik selle tulemusi testida. Generatiivse võistlusvõrgu testimiseks kasutatakse testandmestikku ja määratakse ära treeningu tulemuste kausta (*checkpoint*), mille põhjal saab GAN'i katsetada. Testimiseks kasutati antud töös autori poolt tehtud fotosid. CGAN'i treeniti maksimaalselt 4 000 pildiga ja 200 epochiga, mille jaoks kulus ligikaudu 30 tundi.

### 5.2.1 Mudeli treenimine ja testimine 1 000 pildiga ja 100 epochiga

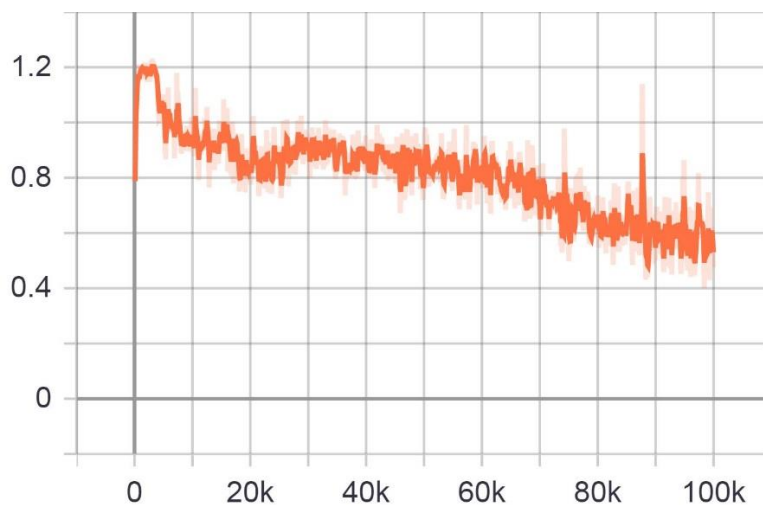
Esialgu testiti CGAN'i mudelit, mida treeniti 1 000 pildi ja 100 epochiga. Protsess võttis aega umbes 3-4 tundi. Mudeli generaatori treenimise kadu on kujutatud TensorFlow abil saadud graafikul (Joonis 11).



Joonis 11. Generaatori kaod treenimise käigus.

Graafikul kujutatud joon tähistab generaatori kadu iga sammu käigus (samm on piltide arv treeningandmestikus korrutatud epochide arvuga). Kao kasv näitab, et generaatoril õnnestub luua pilte, mis diskriminaatorit petavad. Graafikul kujutatud joon kasvab stabiilselt ning

100 000-nda sammu juures pole märgata olulist kasvu aeglustust, mis näitab, et generaatorit saab veel edasi treenida. Edasi vaadeldakse diskriminaatori kadu (Joonis 12).



Joonis 12. Diskriminaatori kaod treenimise käigus.

Graafikul kujutatud joon tähistab diskriminaatori kadu iga sammu käigus. Diskriminaatori kao langus näitab, et diskriminaatoril õnnestub generaatori pilte tuvastada. Graafikul kujutatud joon on stabiilselt langemas ka 100 000ndat sammu juures, mis näitab, et ka diskriminaatorit saab veel edasi treenida.

Mudeli testimiseks kasutati 10 autori poolt tehtud fotot. Järgnevatel joonisel on kujutatud cGAN'i mudeli sisendpilti ja sisendpildist saadud väljundpilti.



Joonis 13. Alavalgustatud sisendpildist (vasak) genereeritud väljundpilt (parem).



Joonisel on näha kuidas treenitud mudel valgustab pilti, et teha sellel olevad objektid nähtavamaks. Märkata võib ka kuidas GAN on genereerinud pildi paremale alla nurka objekti, mis meenutab lilli ning mida pole originaalsel hästi valgustatud fotol. Pilt on tuhm ning sellel esineb musti laiike. Erinevusi genereeritud väljundi ja originaalse hästi valgustatud foto vahel on näha joonisel 14.



Joonis 14. CGAN'i poolt genereeritud pildi (vasak) võrdlus originaalse hästi valgustatud pildiga (parem)



Joonis 15. Alavalgustatud sisendpildist (vasak) genereeritud väljundpilt (parem).

Joonisel 15 on näha teist cGAN'i mudeli poolt genereeritud pildi näidet. Võrreldes joonisel 13 olnud pildiga tundub see tulemus parem. Sarnaselt joonisel 13 väljundpildiga esineb ka joonisel 15 oleval väljundpildil musti laiike (pildi üleval pool).





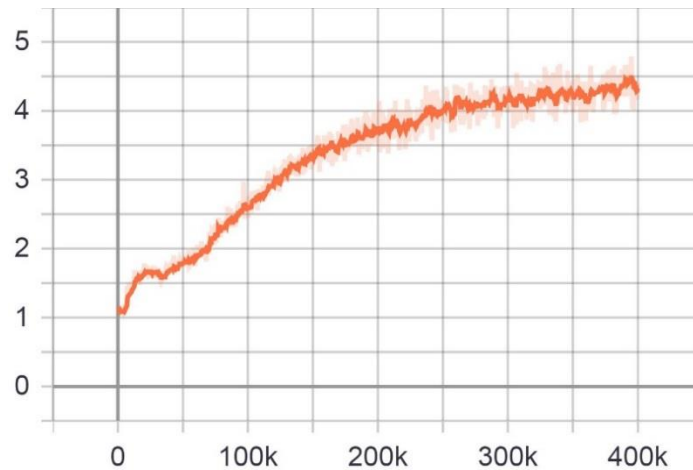
Joonis 16. CGAN'i poolt genereeritud pildi (vasak) võrdlus originaalse hästi valgustatud pildiga (parem).

Joonisel 16 toodus näites on mudeli väljundpilt heledam kui originaal pilt seega võib väita, et cGAN'il õnnestus pilti hästi valgustada kuid pildil esineb rohke müra, mis teeb pildi ähmaseks.

Generaatori ja diskriminaatori graafikul kujutatud kaod näitavad, et mõlemad võrgud suudavad oma ülesannetega hästi toime tulla. Kadude pidev kasv/ langus ütleb seda, et antud generatiivsed võistlusvõrgud õpivad ja neid võib veel edasi treenida, et saavutada paremad tulemused. CGAN'i mudelil õnnestus pilte valgustada ning nendele midagi genereerida. Antud mudelil ei õnnestunud aga vabaneda müra ja osadel piltidel ei ole värvid päris korrektsed.

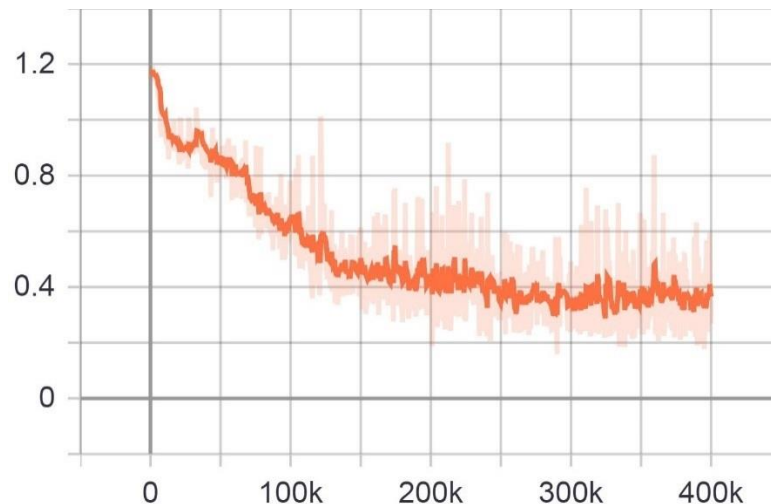
### **5.2.2 Mudeli treenimine ja testimine 2 000 pildiga ja 200 epohhiga**

Paremate tulemuste saavutamise eesmärgiks suurendati järgmiseks katseks piltide kogust 2 000ni ja epohhide arvu 200ni. Mudeli treenimiseks kulus ligikaudu 15 tundi. Mudeli generaatori treenimise kadu on kujutatud graafikul (Joonis 17).



Joonis 17. Generaatori kaod treenimise käigus.

Graafikul on näha, et generaatori kadu on stabiilselt kasvanud kuni 250 000nda sammuni ja peale seda tunduvalt aeglustunud. See näitab, et generaator õppimise protsess edaspidi on väike kuid samas pole kao kasv seisma jäänud. Edasi vaadeldakse diskriminaatori kadusid (Joonis 18).



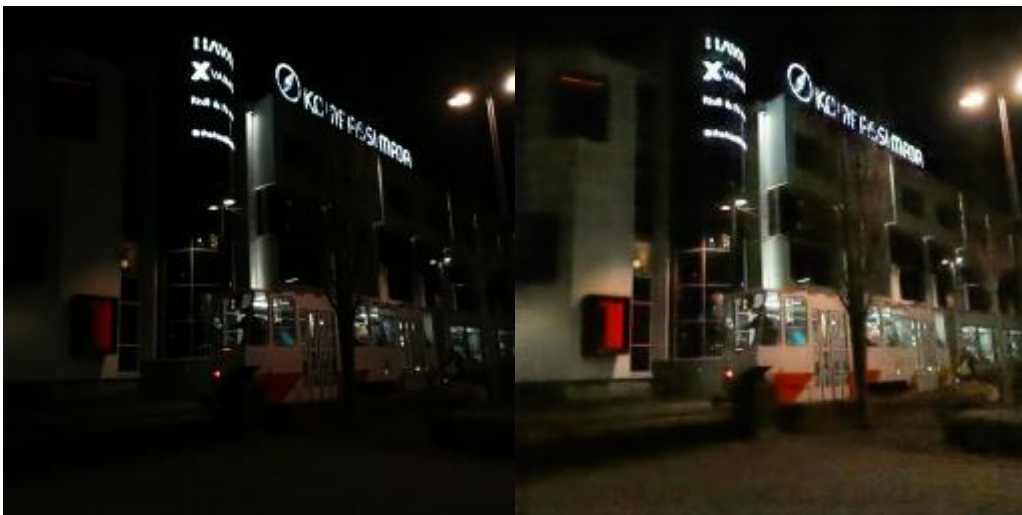
Joonis 18. Diskriminaatori kaod treenimise käigus.

Graafikul kujutatud joon tähistab diskriminaatori kadu iga sammu käigus. Jooniselt 18 on näha, et diskriminaatori kao langust pole enam peale 250 000ndat sammu märgata. Sellest võib järeldada, et diskriminaator õppimise protsess on edaspidi väike.

Mudeli testimiseks kasutati 10 autori poolt tehtud fotot. Järgnevatel joonistel on kujutatud nii sisendpildid kui ka cGAN'i testimise käigus genereeritud väljundpilte.



Joonis 19. Alavalgustatud sisendpildist (vasak) genereeritud väljundpilt (parem).



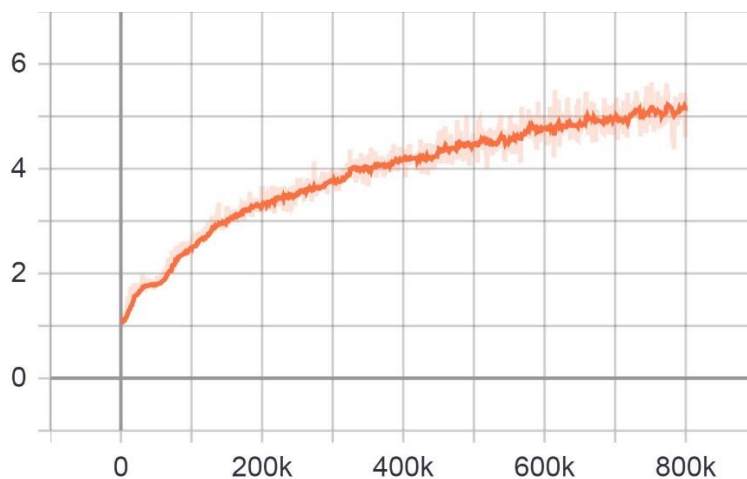
Joonis 20. Alavalgustatud sisendpildist (vasak) genereeritud väljundpilt (parem).

Joonistel 19 ja 20 kujutatud mudelite väljundpildidel võib märgata, et enam ei esine nendel musti plekke nagu esinesid need eelneva mudeli väljundpildidel (joonis 13 ja joonis 15). Joonise 19 väljundpilt on võrreldes joonise 13 omaga tumedam aga selgem. CGAN'i mudelil ei õnnestunud ka seekord vabaneda müra nagu see ilmneb joonisel 20.

Generaatori ja diskriminaatori graafikul kujutatud kaod näitavad, et võrkude õppimise protsess on aeglane ning edaspidine treenimine ei anna sellise andme kogusega enam märgatavaid tulemusi. Antud mudelil õnnestus genereerida selgemad pildid kui eelmisel mudelil.

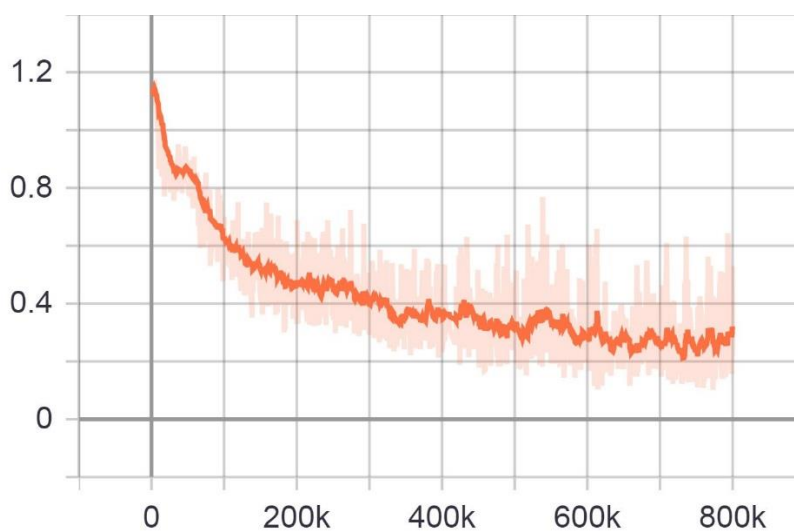
### 5.2.3 Mudeli treenimine ja testimine 4 000 pildiga ja 200 epohhiga

Eesmärgiga saavutada veelgi parem tulemus, treeniti mudelit 4 000 pildiga ja 200 epohhiga. Antud mudeli treenimiseks kulus ligikaudu 30 tundi. Mudeli generaatori treenimise kaod on kujutatud graafikul (Joonis 21).



Joonis 21. Generaatori kaod treenimise käigus.

Joonisel 22 kujutatud graafikul on näha, et generaatori kadu on treenimise käigus kasvanud. Kuigi generaatori kao kasv aeglustub peale 600 000ndat sammu, saab generaatorit edasi treenida. Edasi vaadeldakse diskriminaatori kadu (Joonis 22).



Joonis 22. Diskriminaatori kaod treenimise käigus.

Graafikul kujutatud joon tähistab diskriminaatori kadu iga sammu käigus. Jooniselt 23 on näha, et diskriminaatori kao langust on peale 450 000ndat sammu muutunud väga aeglaseks.

Mudeli testimiseks kasutati 10 autori poolt tehtud fotot. Järgneval joonisel on võrreldud eelneva mudeli väljundpilti selle mudeli omaga



Joonis 23. Eelneva mudeli (vasak) ja selle mudeli (parem) väljund pildid.



Joonis 24. Eelneva mudeli (vasak) ja selle mudeli (parem) väljund pildid.

Vaadates joonistes 23 ja 24 toodud erinevate mudelite tulemuste peale, pole nende vahel märgata olulisi erinevusi. Joonisel 23 toodud piltidel tundub, et eelneva mudeli väljundpilt selgem, joonise 24 puhul aga vastupidi. Sama kehtib ka ülejäänud testandmete puhul.

Generaatori ja diskriminaatori kadudelt selgus, et võrgud ei õpi enam efektiivselt ning edasine treenimine ei tohiks enam anda paremaid tulemusi. Võrreldes omavahel eelneva ja selle mudeli väljundpilte, pole nende vahel märgata olulisi erinevusi.

## 5.2.4 Järeldused

Üldiselt õnnestus cGAN'i mudelitel pilte üsnagi hästi valgustada ning mõnele pildile on proovitud mingit objekti juurde genereerida. Negatiivseteks külgedeks oli see, et pildid olid ähmased, sest ei suudetud vabaneda piltidel olevast müra, mida isegi piltidele juurde tekkis. Peale viimase mudeliga treenimist (mudel 4 000 pildi ja 200 epochiga) ei esinenud piltide kvaliteedi paranemist võrreldes sellele eelneva mudeliga (mudel 2 000 pildi ja 200 epochiga), mistõttu pole mõtet mudelit antud andmestikuga edasi treenida.

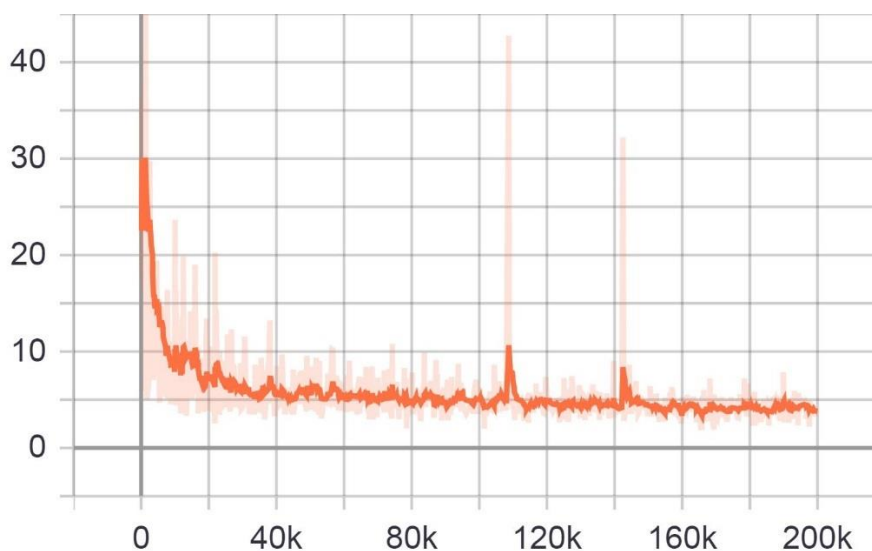
## 5.3 C-DCGAN

Sarnaselt antud töös kasutatavale cGAN'ile õpib tinglik DCGAN teisendama sisendpildi väljundpildiks. Nende kahe võrgu generaator ja diskriminaator täidavad sama ülesannet. Generaator teisendab  $256 \times 256 \times 3$  sisendpildi maatriksi, mis on kõrgusega ja laiusega 256 pikslit ning koosneb kolmest värvi kanalist (punane-roheline-sinine),  $1 \times 1 \times 512$  maatriksiks. Sellisele kujule kokku surutud pilt saadakse mitme kooderi abil. Pildi kokku surumise protsessi otstarve on selles, et saada kõrgetasemelisema andmete esituse peale kooderi viimast kihti. Dekooderi kihtide protsess on vastupidine kooderi omadele st dekodeer teisendab andmed tagasi esialgsele kujule.

C-DCGAN'i treenimiseks ja testimiseks ettenähtu andmestikud on jagatud vastavatesse kaustadesse train ja test. Lisaks treening- ja testandmestikule on vaja määrata c-DCGAN'i jaoks hinnangu andmestiku, mis asub kasutas val. Hinnangu andmestikku on vaja selleks, et testida mudeli tulemusi treening protsessi käigus. Andmestikude jaoks oli vaja kõik pildid teisendada  $256 \times 256$  formaati, mida saab teha `pix2pix` cGAN'i programmis oleva lisa koodiga. Alguses teeb cGAN läbi treenimise protsessi, mille ajaline kulu sõltub piltide resolutsioonist, nende arvust treeningandmestikus (*batch size*) ja epochide arvust. Epoch on kasutaja poolt määratud arv, mis näitab mitu korda treenitakse antud andmestiku peal närvivõrke. Peale treenimise käigus salvestatakse arvutisse treenimise tulemused ja logi koos TensorFlow graafiga, kus on kujutatud generaatori ja diskriminaatori treenimise protsessis käigus muutunud kadusid. Pärast cGAN'i treenimist on võimalik selle tulemusi testida. Generatiivse võistlusvõrgu testimiseks kasutatakse testandmestikku ja määratakse ära treeningu tulemuste kausta, mille põhjal saab GAN'i katsetada. Testimiseks kasutati antud töös autori poolt tehtud fotosid.

### 5.3.1 Mudeli treenimine ja testimine 4 000 pildiga ja 50 epohhiga

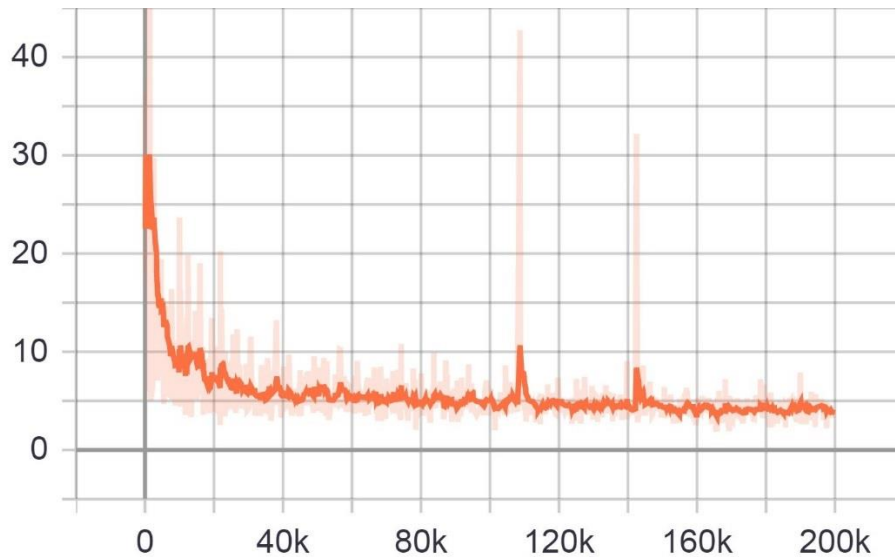
Tingliku DCGAN'i treeniti 4 000 pildiga ja 50 epohhiga, mille jaoks kulus ligikaudu 30 tundi. Kuna antud c-DCGAN'i implementatsiooni treenimise protsess on palju aeglasem cGAN'i omast, pidi epohhide arvu vähendama 50ni. Mudeli generaatori treenimise kadu on kujutatud graafikul (Joonis 25).



Joonis 25. Generaatori kaod treenimise käigus.

Graafikul kujutatud joon tähistab generaatori kadu iga sammu käigus (samm on piltide arv treeningandmestikus korrutatud epohhide arvuga). Sellel graafikul näitab generaatori kao langus, et generaatoril õnnestub luua pilte, mis diskriminaatorit petavad (generaatori kadu arutatakse teistmoodi võrreldes cGAN'is oleva implementatsiooniga). Graafikul kujutatud joon langeb alguses kiiresti kuid pärast 40 000ndat sammu püsib paigal, mis tähendab, et generaator enam ei õpi. Edasi vaadeldakse diskriminaatori kadu (Joonis 26).





Joonis 26. Diskriminaatori kaod treenimise käigus.

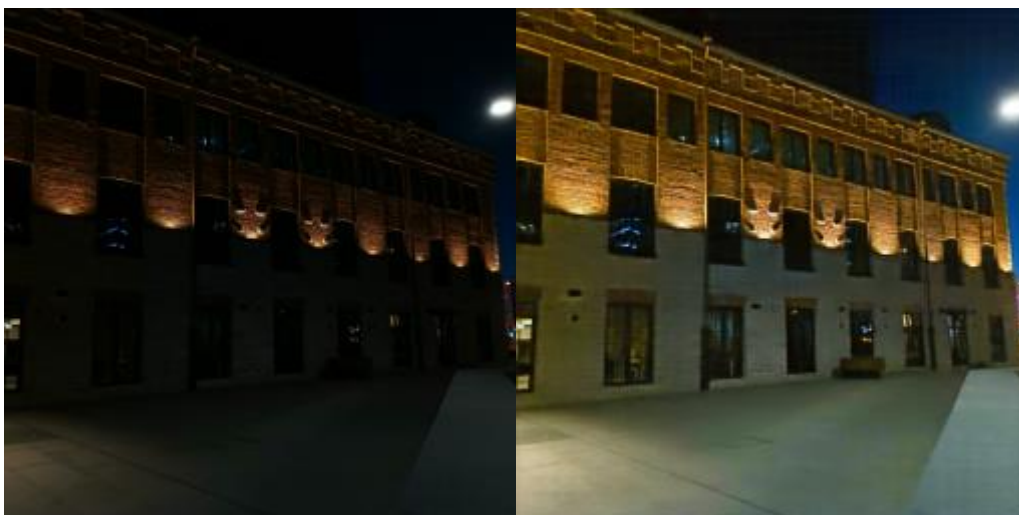
Joonisel 26 kujutatud graafikul näidatakse diskriminaatori kadu iga sammu käigus. Kao langus näitab, et diskriminaatoril õnnestub generaatori pilte tuvastada. Graafikul kujutatud joon langeb alguses kiiresti kuid peale 40 000ndal sammul püsib paigal, mis tähendab, et diskriminaator enam ei õpi.

Mudeli testimiseks on kasutatud 10 autori poolt tehtud fotot. Järgnevatel joonistel on kujutatud nii sisendpildid kui ka c-DCGAN'i testimise tulemusena genereeritud väljundpildid.



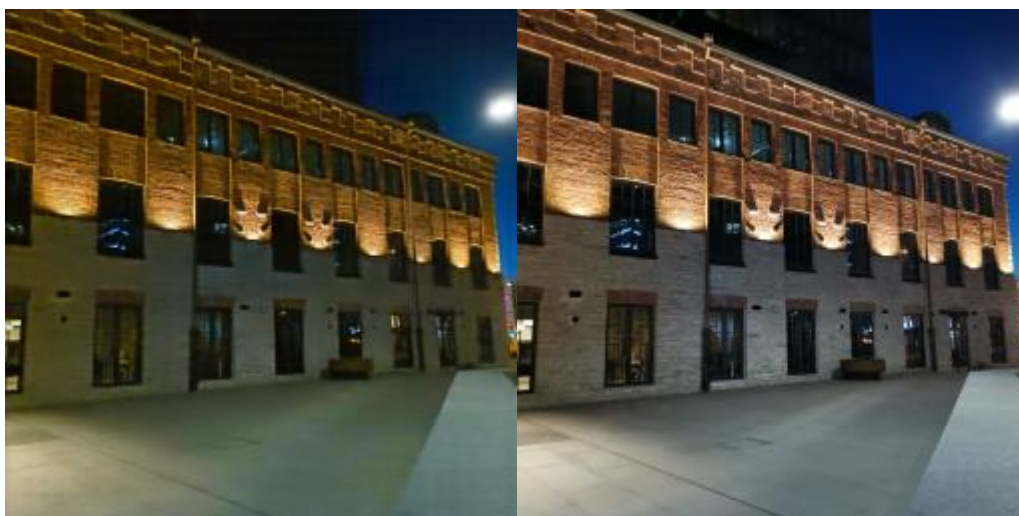
Joonis 27. Alavalgustatud sisendpildist (vasak) genereeritud väljundpilt (parem).





Joonis 28. Alavalgustatud sisendpildist (vasak) genereeritud väljundpilt (parem).

Joonistel 27 ja 28 on näha, kuidas DCGAN'i mudelil õnnestus alavalgustatud pilte valgustada. Joonise 27 väljundpildi üleval maja katuse kohal võib märgata tumedaid täppe, mida generaator on sinna genereerinud. Järgneval joonisel võrreldakse joonisel 29 esitatud väljundpilti selle hästi valgustatud originaal pildiga.



Joonis 29. C-DCGAN'i poolt genereeritud pildi (vasak) võrdlus originaalse hästi valgustatud pildiga (parem).

Võrreldes joonisel 29 esitatud pilte, õnnestus autori hinnangul DCGAN'il pilti küllaltki hästi valgustada. Negatiivsetest külgedest on pildil liigne värvide küllasus ning hoone taga oleva ehitise aknaid pole näha nagu on seda originaal pildil.

### 5.3.2 Järeldused

Mudeli õnnestus pilte küllaltki hästi valgustada ning mõnele pildile on proovitud midagi juurde genereerida, mis näitab, et generatiivne võistlusvõrk töötab, kuigi annab soovida paremat. Negatiivseteks külgedeks oli see, et mõned pildid olid ähmased, sest ei suudetud vabaneda pildidel olevast mürast, mida isegi osadele piltidele natuke juurde tekkis. Lisaks olid genereeritud piltide värvid liiga küllastatud, mis pigem on treeningandmestikku töödeldud GIMP'i plugini probleem. Antud DCGAN'i treenimisel selgus ka, et edasine võrkude treenimine on mõttetu, kuna generaatori ja diskriminaatori kaod püsivad ühel ja samal kohal ega lange.

### 5.4 SRGAN

Eesmärgiga parandada cGAN'i ja DCGAN'i tulemusi on proovitud rakendada nendele SRGAN'i. Töös kasutatud SRGAN on inspiratsiooni saanud pix2pix'ilt ning kasutab TensorFlow tarkvarateeki [25, 33]. SRGAN'i ülesandeks on teha pilt selgemaks parandades selle resolutsiooni. Antud generatiivne võistlusvõrk kasutab eelnevalt treenitud mudelit 8 156 pildiga. Testandmestikus on pildid jagatud kahte kausta ühed kõrge resolutsiooniga piltide kausta ja teised madala resolutsiooniga piltide kausta. Testandmestik koosneb 10 pildist. Joonisel 30 võrreldakse omavahel cGAN'i ja SRGAN'i tulemusi.



Joonis 30. CGAN'i pildi (vasak) töötlemine SRGAN'iga (parem).

Nagu joonisel näha teeb SRGAN pildi teravamaks parandades selle resolutsiooni. Pildi teravuse suurenemisega muutuvad ka kompressiooni artefaktid (müra, ruudud jt) nähtavamaks. Joonisel 31 võrreldakse omavahel c-DCGAN'i ja SRGAN'i tulemusi.



Joonis 31. C-DCGAN'i pildi (vasak) töötlemine SRGAN'iga (parem).

Sarnaselt joonisele 30 teeb SRGAN pildi teravamaks parandades selle resolutsiooni kuid samuti muutuvad nähtavamaks ka pildi kompressiooni artefaktid.

SRGAN parandas teiste generatiivsete võrkude poolt genereeritud piltide resolutsiooni tehes pildid selgemaks. Kuna olulist piltide kvaliteedi paranemist pole antud töö tulemustes märgata ning pildidel olevad artefaktid on muutunud nähtavamaks, otsustas autor edasi töös SRGAN'i mitte kasutada.

## 5.4 Kvantitatiivne hindamine

Käesolevas peatükis võrreldakse töös kasutatud GAN'ide poolt loodud pilte originaalsete valgustatud piltidega. Selleks arvutatakse välja genereeritud piltide keskmine ruutviga (*mean squared error, MSE*), struktuurne sarnasus (*structural similarity, SSIM*) ja maksimaalne signaali müra suhe (*peak signal-to-noise ratio, PSNR*). Tulemuste saamiseks on kirjutatud Pythonis skript, mis võrdleb omavahel pilte, et arvutada MSE, SSIM'i ja PSNR'i väärtused ning salvestada need Excel faili. Saadud väärtused näitavad kui sarnased on kasutatud GAN'ide pildid originaal piltidega. Võrdlemiseks on kasutatud 380 erinevat pilti, mida iga generatiivne võistlusvõrk on genereerinud (kokku on iga GAN'i kohta 760 pilti kuna ühe ja

sama pildi kohta esineb 2 versiooni – genereeritud ja valgustatud originaal). Seejärel arvutatakse iga saadud tulemuse keskmise MSE, SSIM ja PSNR. Saadud tulemusi võrreldakse omavahel ning valitakse välja parima GAN'i, mida edaspidi võrreldakse traditsiooniliste pildi taastamismeetoditega. Leitud tulemused on kujutatud tabelis 1.

|           | MSE(avg) | SSIM(avg) | PSNR(avg) |
|-----------|----------|-----------|-----------|
| CGAN 2000 | 287.84   | 0.77      | 24.2      |
| CGAN 4000 | 337.53   | 0.73      | 23.52     |
| DCGAN     | 195.28   | 0.82      | 26.49     |
| Fotor     | 1031.43  | 0.65      | 18.42     |

Tabel 1. Keskmised MSE, SSIM ja PSNR väärtused.

Kõrgem MSE tähistab seda, et pildil esineb rohkem vigu pikslite kohta. Identse pildi puhul on MSE väärtus 0. SSIM leiab pildi pikslite vahel sarnasusi. SSIM'i väärtused asuvad vahemikus -1 kuni 1, kus väärtus 1 tähistab identseid pilte ja väärtus -1 erinevaid pilte. Võrreldes MSE'ga on SSIM'i väärtuseid paremini liigitada, kuna MSE väärtustel puudub kindel vahemik. PSNR on maksimaalne signaali müra suhe. Piltide puhul tähistab signaal originaal pildi andmeid ja müra pildi kompressiooni käigus tekkinud vigu. Mida suurem on PSNR seda lähemal on võrreldava pildi kvaliteet originaaline.

#### 5.4.2 CGAN'i 4 000 ja 2 000 pildiga treenitud mudelite võrdlus

Antud töö käigus selgus, et kahe cGAN'i mudelite tulemuste vahel ei ole olulisi erinevusi. Autori arvates esines 2 000 pildiga treenitud mudelil pilte, mis olid kvaliteedi poolt paremad kui 4 000 pildiga treenitud mudelil ja vastupidi. Tabelis 1 oleva 2 000 pildiga treenitud mudeli (CGAN 2000) keskmised tulemused on järgmised – MSE: 284,84; SSIM: 0,77; PSNR: 24,20. Tabelis 1 toodud 4 000 pildiga treenitud mudeli (CGAN 4000) keskmised tulemused on järgmised – MSE: 337,53; SSIM: 0,73; PSNR: 23,52. Tulemustes selgub, et mudel mida treeniti 2 000 pildiga suudab luua paremaid pilte kui mudel mida treeniti 4 000 pildiga.

#### 5.4.3 C-DCGAN'i ja CGAN'i mudelite võrdlus

Käesolevas peatükis võrreldakse tingliku DCGAN'i tulemusi CGAN'i omadega. CGAN'i mudeliks on selles võrdluses kasutatud 2 000 pildiga treenitud mudelit, kuna see suutis luua paremaid tulemusi. Antud töö käigus õnnestus DCGAN'il genereerida küllalt head

tulemused, mis autori arvates ületavad CGAN'i omi kvaliteedi poolest. Järgnevatel joonistel võrreldakse kahe generatiivse võistlusvõrgu tulemusi.



Joonis 32. CGAN'i tulemuse (vasak) võrdlus DCGAN'i omaga (parem).



Joonis 33. CGAN'i tulemuse (vasak) võrdlus DCGAN'i omaga (parem).

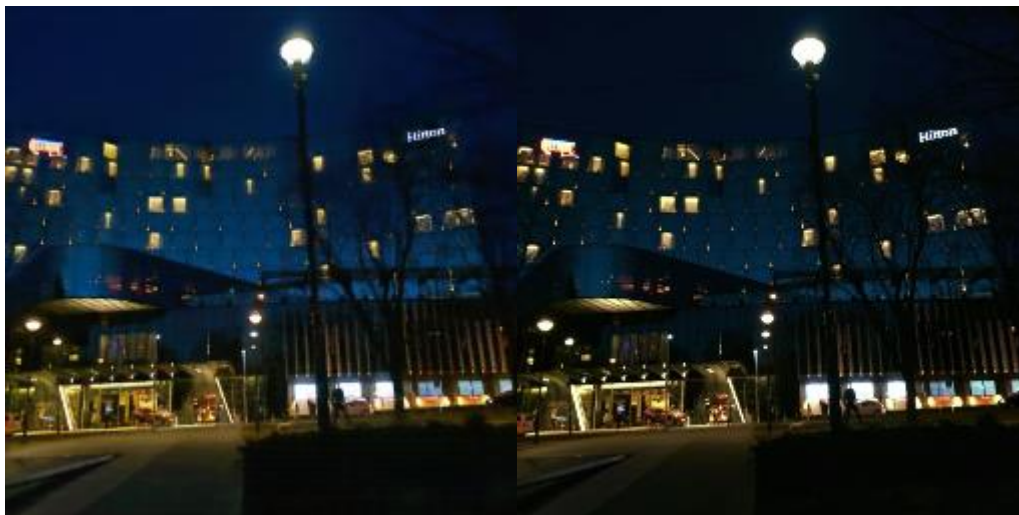
Ülaltoodud joonistel võib näha, et tinglikul DCGAN'il õnnestus luua selgemad pildid, kuna pildidel esineb vähem müra ja kompressiooni artefakte. Ka värvide ja valgustuse poolest tundub DCGAN'i tulemus veidi ületavat cGAN'i oma.

Vastavalt tabelile 1 on C-DCGAN'i kvantitatiivse hinnangu keskmised tulemused järgmised – MSE: 195,28; SSIM: 0,82; PSNR: 26,49. Tabelis 1 toodud cGAN'i kvantitatiivse hinnangu keskmised tulemused on järgmised – MSE: 284,84; SSIM: 0,77; PSNR: 24,20. Tulemustest selgub, et DCGAN'il saavutas paremad MSE, SSIM ka PSNR väärtused, mis kinnitab, et DCGAN'i poolt loodud pildid on sarnasemad originaalsetele valgustatud piltidele.



#### 5.4.4 C-DCGAN'i võrdlus traditsiooniliste pilditaastamise meetodiga

Eelnevates peatükkides selgus, et tinglikul DCGAN'il õnnestus luua parimad tulemused võrreldes cGAN'iga. Antud peatükis võrreldakse DCGAN'i tulemusi Fotor pilditöötlusprogrammis kunstlikult valgustatud piltidega. Piltide töötlemiseks pandi alavalgustatud piltide peale öö filter, et tõsta piltide heledust. DCGAN'i ja Fotor'i piltide erinevusi on näha järgneval joonisel.



Joonis 34. DCGAN'i tulemuse (vasak) võrdlus Fotor'i omaga (parem).

Jooniselt 34 võib näha, et DCGAN'il õnnestus pilti paremini valgustada kui Fotoril. Vaatamata sellele on pilditöötlusprogrammis saadud pilt selgem.

Vastavalt tabelile 1 on C-DCGAN'i kvantitatiivse hinnangu keskmised tulemused järgmised – MSE: 195,28; SSIM: 0,82; PSNR: 26,49. Fotor'is töödeldud kvantitatiivse hinnangu keskmised tulemused on järgmised – MSE: 1031.43; SSIM: 0.65; PSNR: 18.42. Kvantitatiivse hinnangu tulemustes selgus, et DCGAN'il tulemused on palju lähemal originaalsetele valgustatud fotodele kui Fotor'i tulemused. Kuigi Fotor'iga õnnestus säilitada pildi selgust, ei piisanud öö filtrist, et pilti piisavalt valgustada.

#### 5.4.5 Järeldused

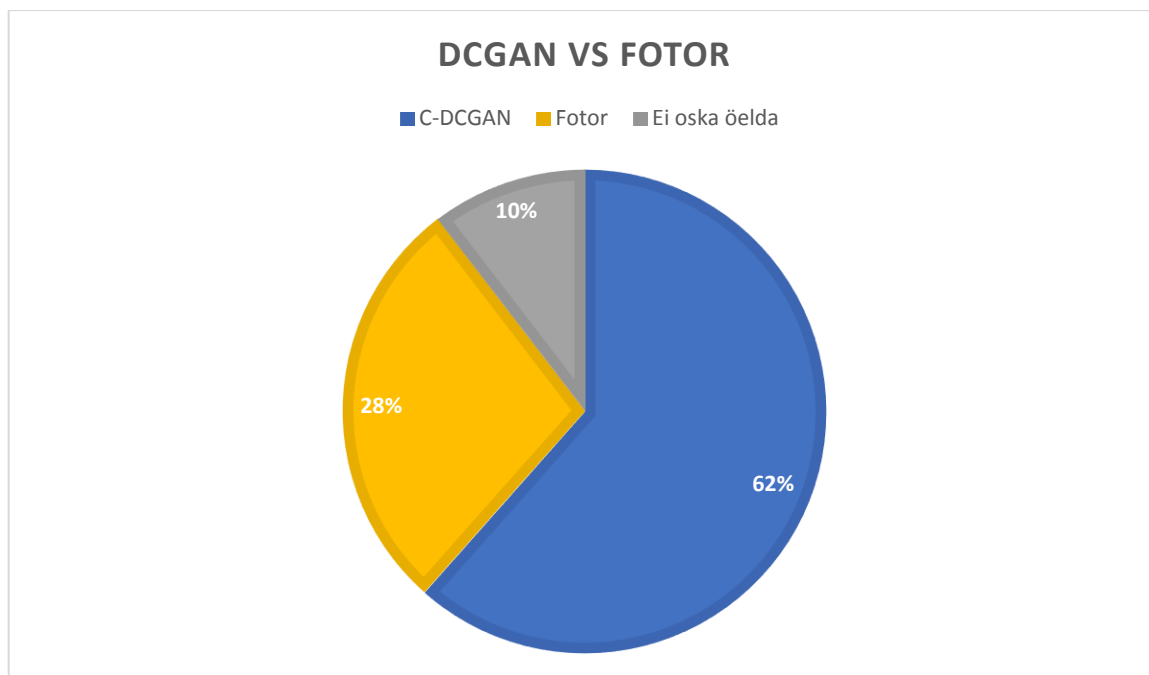
Kvantitatiivse hinnangu tulemustes selgus, et tinglikul DCGAN'il õnnestus saavutada paremad tulemused kui cGAN'il. Kõigi kolme arvutatud väärtuse suhtes ületab DCGAN cGAN'i omi. Vastavalt sellele võrreldi edasi tingliku generatiivse võistlusvõrgu tulemusi pilditöötlusprogrammis Fotor saadud tulemustega. Kuigi pilditöötlusprogrammiga saadud

pildid olid selgemad polnud nad piisavalt heledad. Mõlema võrreldava meetodi saadud keskmised MSE, SSIM ja PSNR väärtustest, selgus et DCGAN'i pildid olid kvantitatiivse hinnangu tulemusena paremad.

## 5.5 Kvalitatiivne hindamine

Piltide kvaliteedi hindamiseks on koostatud küsimustik. Küsimustikus võrreldakse omavahel tingliku DCGAN'i tulemusi pilditöötlusprogrammi Fotor saadud tulemustega. Küsimustik koosneb 10 küsimusest, kus igas küsimuses on toodud 3 pilti. Küsitletaval tuleb võrrelda kahte kõrvuti olevat pilti (vasak pilt ja parem pilt) – kumb nendest on sarnasem kolmandale. Kaheks kõrvuti olevaks piltideks on tingliku DCGAN'iga ja pilditöötlusprogrammiga Fotor saadud tulemused, mis on juhuslikus järjekorras. Kolmandaks pildiks on hästi valgustatud originaal foto. Küsitletavale ei ole teada antud, mis meetodiga on pilti valgustatud. Iga küsimuse vastuseks on pakutud kolm varianti – vasak pilt, parem pilt ja ei oska öelda. Küsimustiku eesmärk on saada inimeste hinnangu, kumb kahest meetodist suudab saavutada tulemuse, mis on lähemal originaalsele pildile.

Küsimustikule vastas 26 inimest. Iga 10 küsimuse tulemused on kokku summeeritud ja kujutatud järgneval graafil.



Joonis 35. Kvalitatiivse hinnangu küsitluse tulemused.

Küsimustiku tulemusena selgus, et suurema osa küsitatavate inimeste (62%) hinnangul suutis c-DCGAN luua kvaliteedi poolest paremad pildid kui Fotor. See tähendab, et küsitatavate arvates olid tingliku generatiivse võistlusvõrku pildil sarnasemad originaal fotodele.



## Kokkuvõte

Antud bakalaureusetöö eesmärgiks oli analüüsida ja hinnata, kui hästi suudavad generatiivsed võistlusvõrgud alavalgustatud fotosid valgustada. Töökäigus võrreldi omavahel kahte erinevat generatiivset võistlusvõrku – cGAN ja c-DCGAN. GAN’ide maksimaalne treeningandmestik sisaldas 4 000 pilti. Mõlemaid GAN’i tulemusi prooviti parandada SRGAN’i abil. SRGAN suutis edukalt piltide resolutsiooni muuta, mis tegi piltidel olevad detailid teravaks. Teravuse suurenemisega piltidel muutusid ka piltidel olevad kompressiooni artefaktid nähtavamaks. Seoses sellega otsustas antud töö autor SRGAN’i tulemusi edaspidi töös mitte kasutada.

Kvantitatiivseks hinnanguks arvutati välja GAN’ide väljundpiltide keskmised MSE, SSIM ja PSNR. Tulemuste saamiseks võrreldi generatiivsete võistlusvõrkude pilte valgustatud originaal piltidega. Kvantitatiivse hinnanguga selgus, et tingliku DCGAN’i poolt saadud tulemused olid cGAN’i omadest paremad. Edasi võrreldi omavahel c-DCGAN’i ja pilditöötlus-programmi Fotor tulemusi. Ka sellest selgus, et c-DCGAN suutis saavutada paremad tulemused.

Kvalitatiivse hinnangu saamiseks oli koostatud küsimustik, kus inimesed said võrrelda omavahel c-DCGAN’i ja pilditöötlusprogrammi Fotor väljundpilte. Küsimustik koosnes 10 küsimusest, kus tuli otsustada kumb kahest meetodi poolt saadud piltidest oli sarnasem valgustatud originaal pildiga. Kokku oli vastanud 26 inimest. Küsimuste tulemused summeeriti ning selgus, et DCGAN suutis küsitletavate hinnangul (62%) saavutada paremad tulemused.

Üldiselt suutsid generatiivsed võistlusvõrgud edukalt pilte valgustada. Üheks puuduseks oli see, et ei suudetud piltidelt olevast müra vabaneda ning mida isegi tekkis juurde. Sellel põhjusel ei olnud genereeritud pildid päris selged. Lisaks ei õnnestunud GAN’idel pimeduse tõttu kaduma läinud informatsiooni edukalt taastada. Tulemusi saaks parandada veelgi kui kasutada palju suuremat treeningandmestikku ja katsetada teisi generatiivsete võistlusvõrkude implementatsioone.

## Kasutatud kirjandus

- [1] A Beginner's Guide to Neural Networks and Deep Learning [WWW] <https://skymind.ai/wiki/neural-network> (07.05.2019)
- [2] S. Russell, P. Norvig. Artificial Intelligence: A Modern Approach. 3rd edition. Prentice Hall, 2009.
- [3] J. Samuel. Neurovõrgud [WWW] <https://medium.com/@Tulevik/neurov%C3%B5rgud-c9b57b3f8673> (07.05.2019)
- [4] L. Dormehl. What is an artificial neural network? Here's everything you need to know [WWW] <https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/> (07.05.2019)
- [5] A Beginner's Guide to Backpropagation in Neural Networks [WWW] <https://skymind.ai/wiki/backpropagation> (07.05.2019)
- [6] L. Deng, D. Yu. Deep Learning: Methods and Applications. Now, 2014
- [7] K. D. Foote .A Brief History of Deep Learning [WWW] <https://www.dataversity.net/brief-history-deep-learning/> (08.05.2019)
- [8] M. Hargrave. Deep Learning [WWW] <https://www.investopedia.com/terms/d/deep-learning.asp> (08.05.2019)
- [9] Neural Networks vs Deep Learning – Useful Comparisons To Learn [WWW] <https://www.educba.com/neural-networks-vs-deep-learning/> (08.05.2019)
- [10] Artificial Intelligence (AI) vs. Machine Learning vs. Deep Learning [WWW] <https://skymind.ai/wiki/ai-vs-machine-learning-vs-deep-learning> (08.05.2019)
- [11] A Beginner's Guide to Convolutional Neural Networks (CNNs) [WWW] <https://skymind.ai/wiki/convolutional-network> (08.05.2019)
- [12] S. Indolia, A. K. Goswami, S. P. Mishrab, P. Asopa. Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. – *International Conference on Computational Intelligence and Data Science (ICCIDS 2018)*. 2018, 132, 679–688. [Online] ScienceDirect (10.05.2019)
- [13] Convolutional Neural Networks (CNNs) explained [WWW] [http://deeplizard.com/learn/video/YRhxdVk\\_sIs](http://deeplizard.com/learn/video/YRhxdVk_sIs) (08.05.2019)
- [14] S. Patel, J. Pingel. Introduction to Deep Learning: What Are Convolutional Neural Networks? [WWW] <https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html> (10.05.2019)

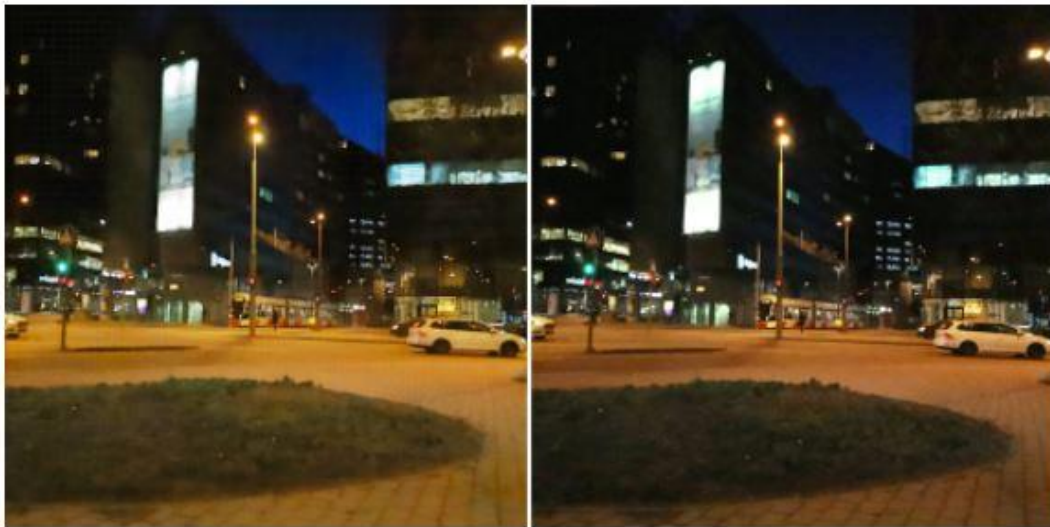
- [15] J. Jordan. Convolutional neural networks [WWW] <https://www.jeremyjordan.me/convolutional-neural-networks/> (10.05.2019)
- [16] T. L. Fonseca. What's happening inside the Convolutional Neural Network? The answer is Convolution. [WWW] <https://buzzrobot.com/whats-happening-inside-the-convolutional-neural-network-the-answer-is-convolution-2c22075dc68d> (10.05.2019)
- [17] Student Notes: Convolutional Neural Networks (CNN) Introduction [WWW] <https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/> (19.05.2019)
- [18] A Beginner's Guide to Generative Adversarial Networks (GANs) [WWW] <https://skymind.ai/wiki/generative-adversarial-network-gan>. (05.05.2019)
- [19] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, Generative Adversarial Nets. *arXiv:1406.2661 [stat.ML]*, 10 Jun 2014.
- [20] A. Gonfalonieri. How to Build A Data Set For Your Machine Learning Project [WWW] <https://towardsdatascience.com/how-to-build-a-data-set-for-your-machine-learning-project-5b3b871881ac> (07.05.2019)
- [21] T. Silva. An intuitive introduction to Generative Adversarial Networks (GANs) [WWW] <https://medium.freecodecamp.org/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394> (10.05.2019)
- [22] M. Mirza, S. Osindero. Conditional Generative Adversarial Nets. *arXiv:1411.1784 [cs.LG]*, 6 Nov 2014.
- [23] Conditional generative adversarial network (cGAN) [WWW] [https://golden.com/wiki/Conditional\\_generative\\_adversarial\\_network\\_\(cGAN\)](https://golden.com/wiki/Conditional_generative_adversarial_network_(cGAN)) (10.05.2019)
- [24] A. Radford, L. Metz, S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434 [cs.LG]*, 7 Jan 2016 (12.05.2019)
- [25] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *arXiv:1609.04802 [cs.CV]*, 25 May 2017.
- [26] Image-to-Image Translation with Conditional Adversarial Nets [Online] <https://phillipi.github.io/pix2pix/> (13.05.2019)
- [27] Tensorflow port of Image-to-Image Translation with Conditional Adversarial Nets [Online] <https://github.com/affinelayer/pix2pix-tensorflow> (13.05.2019)

- [28] TensorFlow implementation of "Image-to-Image Translation Using Conditional Adversarial Networks". [Online]  
<https://github.com/yenchenlin/pix2pix-tensorflow> (13.05.2019)
- [29] P. Isola, J. Zhu, T. Zhou, A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv:1611.07004 [cs.CV]*, 26 Nov 2018.
- [30] A. Unruh. What is the TensorFlow machine intelligence platform? [WWW]  
<https://opensource.com/article/17/11/intro-tensorflow> (13.05.2019)
- [31] What is Python? Executive Summary [WWW]  
<https://www.python.org/doc/essays/blurb/> (13.05.2019)
- [32] C. Hesse. Image-to-Image Translation in Tensorflow [WWW]  
<https://affinelayr.com/pix2pix/> (14.05.2019)
- [33] Tensorflow implementation of the SRGAN algorithm for single image super-resolution [Online] <https://github.com/brade31919/SRGAN-tensorflow> (16.05.2019)

## Lisa 1. Kvalitatiivse küsimustiku näide

7. Kumb pilt on sarnasem allolevale originaal fotole?

- Vasak pilt
- Parempilt
- Ei oska öelda



Originaal foto



Näide küsimustikus olevast küsimusest