

Aalto-yliopisto  
Sähkötekniikan korkeakoulu  
Bioinformaatioteknologia

**Emma Rauhala**

**Koneoppiminen ja generatiiviset kilpailevat verkostot magneettikuvien  
painotuksien syntetisoinnissa**

Kandidaatintyö

30.11.2020

Työn ohjaaja:

Apulaisprofessori, TkT Ilkka Laakso

Tekijä: Emma Rauhala	
Työn nimi: Koneoppiminen ja generatiiviset kilpailevat verkostot magneettikuvien painotuksien syntetisoinnissa	
Koulutusohjelma: Bioinformaatioteknologia	
Päiväys: 30.11.2020	Sivumäärä: 44+4
Vastuunopettaja: TkT Markus Turunen	
Ohjaaja(t): Apulaisprofessori, TkT Ilkka Laakso	
Kieli: suomi	
<p>Tässä kandidaatintyössä esitetään kuvaus siitä, miten koneoppimista ja erityisesti generatiivisia kilpailevia verkostoja voi soveltaa magneettikuvien painotuksien syntetisointiin. Tutkielmassa kuvataan seikkaperäisesti konvoluutioneuroverkkojen toiminta kuvankäsittelysovelluksissa, ja tämä ymmärrys kytketään edelleen kilpailevien verkkojen arkkitehtuuriin. Kirjallisuuskatsaukseen pohjautuen työhön on koottu yhteenveto tavoista, joilla kilpailevien verkkojen oppimista voidaan ohjata kuvankäsittelyssä ja erityisesti magneettikuvasynteesin kontekstissa. Esitettyjen tulosten perusteella kilpailevien verkkojen avulla saadaan tuotettua eritoten visuaalisesti laadukkaita synteettisiä kuvia. Erityisesti huomattavaa on se, että verkkojen opetuksessa voidaan hyödyntää myös paritonta opetusaineistoa. Magneettikuvien korkean tason rakenteellisen informaation säilymistä tukeva ohjaus vaikuttaisi olevan avainasemassa synteesin onnistumisessa.</p>	
Avainsanat: generatiivinen kilpaileva verkosto, koneoppiminen, konvoluutioneuroverkko, kuvankäsittely, magneettikuvaus	

# Sisällysluettelo

<b>Sisällysluettelo .....</b>	<b>iii</b>
<b>Käsitteet ja lyhenteet .....</b>	<b>iv</b>
<b>1. Johdanto.....</b>	<b>1</b>
<b>2. Neuroverkot.....</b>	<b>3</b>
2.1. Perusrakenne: perseptroniverkot .....	3
2.2. Kuvankäsittely: konvoluutioneuroverkot .....	4
2.2.1. Kuvapiirteiden erottelu: kokoava konvoluutio .....	5
2.2.2. Piirteiden uudelleenyhdistäminen: transponoitu konvoluutio ja täyttö ...	7
2.3. Kuva-aineisto neuroverkon syötteenä: tensorirakenteesta .....	8
2.4. Oppiminen.....	9
2.4.1. Arvofunktio.....	9
2.4.2. Optimointi.....	10
2.5. Haasteita .....	12
2.5.1. Laskennallinen alivuoto ja katoavat gradientit .....	12
2.5.2. Ylisovittaminen.....	13
2.5.3. Arvofunktion valinta.....	13
<b>3. Kilpailevat neuroverkot eli GAN:it.....</b>	<b>14</b>
3.1. Rakenne ja toiminta.....	14
3.2. Ehdollinen generointi: GAN:ista cGAN:iin.....	15
3.3. Soveltaminen kuvankäsittelyyn.....	16
3.3.1. Pikselikohtainen häviö: pGAN eli Pix2pix.....	16
3.3.2. Rekonstruktiohäviö: CycleGAN.....	17
3.3.3. Erilliset enkooderit: UNIT .....	18
3.3.4. Bayesilainen lähestymistapa: KL-divergenssihäviö .....	19
3.3.5. Muita arvofunktion lisätermejä.....	20
3.4. Generaattorin arkkitehtuuri .....	20
3.5. Diskriminaattorin arkkitehtuuri.....	21
3.6. Haasteita .....	22
<b>4. GAN:ien soveltaminen magneettikuvapainotusten syntetisoinnissa.....</b>	<b>23</b>
4.1. Magneettikuvien painotukset .....	23
4.1. Lääketieteellisten kuvien synteisistä .....	24
4.1.1. Tarpeet .....	24
4.1.2. Kilpailevien verkkojen vahvuuksia.....	25
4.2. Tulosten arviointi .....	26
4.2.1. Kvantitatiivinen arviointi: tunnuslukuja .....	26
4.2.2. Kvalitatiivinen arviointi .....	27
4.2.3. Mallin luotettavuus .....	27
4.3. Magneettikuvien T1↔T2 -synteesi: kirjallisuudessa esitettyjä sovelluksia ..	28
Taulukko 1: GAN-sovelluksia magneettikuvien T1 ja T2 -painotusten	
syntetisoinnissa.....	36
<b>5. Yhteenveto .....</b>	<b>37</b>
<b>Lähdeluettelo .....</b>	<b>40</b>

## Käsitteet ja lyhenteet

<b>CNN</b>	<i>Convolutional Neural Network</i> konvoluutioneuroverkko
<b>FCN</b>	<i>Fully Convolutional Neural Network</i> täysin konvolutiivinen neuroverkko
<b>GAN</b>	<i>Generative Adversarial Net</i> generatiivinen kilpaileva verkosto
<b>cGAN</b>	<i>conditional Generative Adversarial Net</i> ehdollinen generatiivinen kilpaileva verkosto
<b>CycleGAN</b>	<i>Cycle Generative Adversarial Network</i> syklinen generatiivinen kilpaileva verkosto
<b>pGAN</b>	<i>pixelwise Generative Adversarial Network = Pix2pix</i> generatiivinen kilpaileva verkosto pikselikohtaisella häviöllä
<b>MRI</b>	<i>Magnetic Resonance Imaging</i> magneettikuvaus
<b>T1</b>	magneettikuvapainotus; mittauspulssissa lyhyt toisto aika ja lyhyt kaiku aika
<b>T2</b>	magneettikuvapainotus; mittauspulssissa pitkä toisto aika ja pitkä kaiku aika
<b>T2-Flair</b>	<i>T2-Fluid Attenuated Inversion Recovery</i> magneettikuvapainotus; mittauspulssissa pitkä toisto aika ja pitkä kaiku aika, aivo-selkäydinnestesuppressoitu
<b>PD</b>	<i>Proton Density</i> magneettikuvapainotus; mittauspulssissa pitkä toisto aika ja lyhyt kaiku aika
<b>Arvofunktio</b>	<i>Value function</i> Optimoinnissa maksimoitava tai minimoitava funktio

# 1. Johdanto

Eri kudostyyppien erottuvuuden parantamiseksi magneettikuvia otetaan tyypillisesti usealla eri painotuksella. Kuvaussyklin toistaminen useaan kertaan on kuitenkin aikaa vievää sekä kallista, ja usein osa kuvista on niissä esiintyvien häiriöiden vuoksi käyttökelvottomia. Niinpä on pyritty kehittämään laskennallisia menetelmiä, joiden avulla yhdestä kuvaussekvenssistä voitaisiin synteettisesti tuottaa muita painotuksia vastaavia kuvia. Synteettisillä kuvilla on erityisen suuri arvo esimerkiksi koneoppimiseen perustuvan automaattisen diagnostiikan kehittämisessä, jossa opetusaineiston rajallinen saatavuus aiheuttaa haasteita.

Koneoppimisen hyödyntäminen kuvien syntetisoinnissa on jo arkipäivää usealla alalla, niin tieteessä kuin taiteessakin. Koneoppimiseen perustuvien mallien etuja niin kuvankäsittelyssä kuin muissakin sovelluskohteissa ovat automaattisuus, adaptoituvuus ja mahdollisuus mallien jatkuvaan kehittämiseen. Generoivan eli luovan koneoppimisen lupaavin menetelmä on kilpailevien neuroverkkojen arkkitehtuuri eli nk. generatiivinen kilpaileva verkosto (*Generative Adversarial Network, GAN*), jonka esittelivät Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville & Bengio vuonna 2014. Kilpailevien verkkojen arkkitehtuuri mahdollistaa kuvien syntetisoinnin ainutlaatuisella tavalla, jopa täysin ilman parillisia opetusesimerkkejä.

Tämän kandidaatintyön päämääränä on luoda katsaus siihen, miten koneoppimista ja erityisesti kilpailevia neuroverkkoja voidaan hyödyntää synteettisten magneettikuvakontrastien luomisessa. Erityisesti työssä keskitytään selvittämään, miten kuvien kontrastia voidaan vaihtaa kahden tyypillisesti käytetyn magneettikuvapainotuksen, T1- ja T2-painotusten, välillä.

Työn varsinaisena tavoitteena ei ole syventyä magneettikuvauksen – tai sen eri painotusten – taustalla olevaan fysiikkaan ja niinpä painotuksien taustaa kuvataankin vain siltä osin, kuin niiden välisten erojen ja diagnostisen tarkoituksen ymmärtämisen kannalta on välttämätöntä. Koneoppimispuolella työssä rajaudutaan niin ikään tarkastelemaan neuroverkkojen toimintaa melko käytännöllisellä abstraktiotasolla,

menemättä syvemmin esimerkiksi menetelmien taustalla oleviin matemaattisiin oletuksiin.

Tutkielman kaksi seuraavaa lukua painottuvat koneoppimiseen: luvussa kaksi esitellään neuroverkkojen – ja luvussa kolme erityisesti kilpailevien neuroverkkojen rakenne ja toimintaperiaatteet. Samassa yhteydessä esitellään myös kilpailevien verkkojen erityisesti kuvankäsittelyyn soveltuvia variaatioita. Neljännessä luvussa kuvataan aluksi lyhyesti magneettikuvakontrastit ja tarkastellaan sitten niiden syntetisointiin kirjallisuudessa esitettyjä koneoppimissovelluksia. Viidennessä yhteenvetoluvussa kootaan kirjallisuuskatsauksen anti sekä tuodaan esille työn tarjoamia oivalluksia.

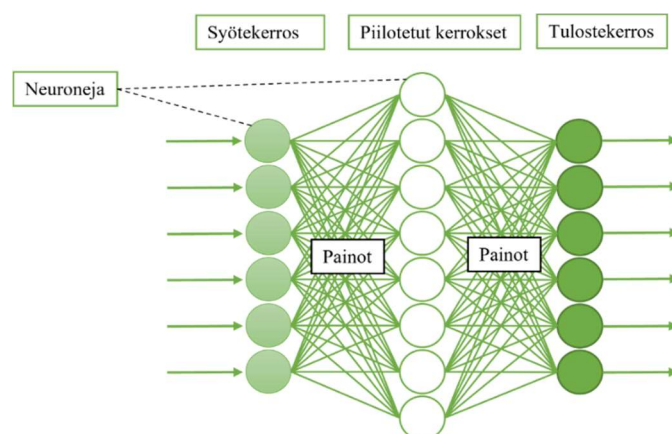
## 2. Neuroverkot

### 2.1. Perusrakenne: perseptroniverkot

Keinotekkoisten neuroverkkojen rakenteen esitteli jo vuonna 1957, tiettävästi ensimmäisenä, psykologi Frank Rosenblatt yrittäessään mallintaa biologisten hermosolujen toimintaa (Nagy, 1991; Rosenblatt, 1957). Rakenteen oletettiin jäljittelevän tapaa, jolla ihmisen aivosolut prosessoivat näköaistimuksia. Rosenblattin esittämä rakenne on adoptoitu lähes sellaisenaan modernin koneoppimisen käyttöön.

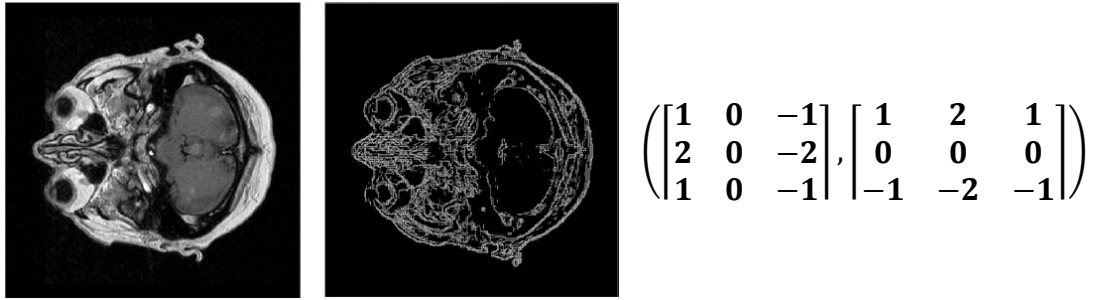
Koneoppimisessa käytettävät neuroverkot koostuvat *kerroksista*, jotka kohdistavat syötedataan  $x$  erilaisia tensorioperaatioita tuottaakseen ennusteen  $y$ . Kerrokset voivat olla rakenteeltaan erilaisia, mutta yhteistä niille on, että kerroksessa suoritettujen toimenpiteiden voimakkuutta kuvataan vastaavilla *painoilla*. Opetettaessa verkkoon siis tallentuvat sellaiset painot, jotka tuottavat halutun kuvauksen  $f: X \rightarrow Y$ . Kerroksen muodostavia laskennallisia yksiköitä kutsutaan biologista analogiaa seuraten *neuroneiksi*.

Tyypillisesti neuroverkon kerrokset jaetaan kolmeen tyyppiin: verkossa on yksi sensorinen kerros tai syötekerros, joka vastaanottaa datan, yksi vaste- tai tulostekerros, joka antaa verkon tuottaman ennusteen, ja näiden välissä yksi tai useita piilotettuja kerroksia (engl. *hidden layers*), joissa varsinainen oppiminen tapahtuu. Yksinkertaisimmillaan toimiva neuroverkko voi siis olla kolmen kerroksen syvyinen. Alla oleva kuva 1 on esimerkki yksinkertaisesta täysin kytketystä perseptroniverkosta, jossa jokainen alemman kerroksen neuroni on yhdistetty jokaiseen seuraavan kerroksen neuroniin.



Kuva 1: Neuroverkon kerrosrakenne

## 2.2. Kuvankäsittely: konvoluutioneuroverkot

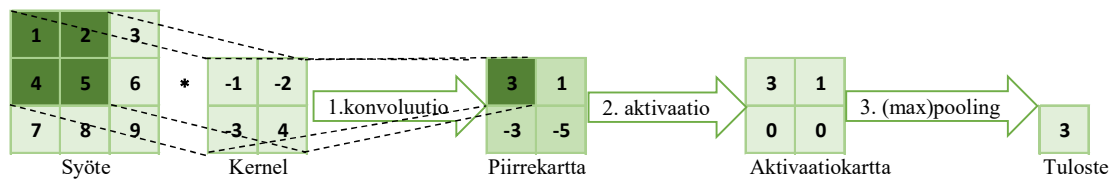


Kuva 2: Kuvapiirteiden eristäminen konvoluution avulla: Sobel-reunantunnistusfilterin tuotos ja sitä vastaavat kernel-matriisit. MRI-kuva: (Karras & Mertzios, 2019).

Konvoluutioneuroverkot ovat säädellympi versio perseptroniverkoista; ainakin osassa piilotettuja kerroksia suoritetaan konvoluutio, jolloin yksi ylemmän kerroksen neuronin vastaanottaa syötteen kaikkien alemman kerrosten neuronien sijaan vain pienestä osasta: alueesta, jota kutsutaan kyseisen neuronin reseptiiviseksi kentäksi (engl. *receptive field*). Reseptiivisestä kentästä käytetään kuvien yhteydessä tyypillisesti myös termiä *tilkku* (engl. *patch*). Konvoluutioneuroverkkoja käyttämällä voidaan säilyttää syötedatan spatiaalinen informaatio ilman, että optimoitavien parametrien, eli neuroverkon painojen, määrä kasvaa kohtuuttomasti. Tästä syystä ne soveltuvat erinomaisesti kuvankäsittelyyn. Parametrien rajallisempi määrä auttaa niin ikään välttämään ylisovittamista. Lisäksi pikselimäärältään edes kohtuullisen kuva-aineiston prosessointi ilman konvoluutioverkkojen mahdollistamaa dimensionaalisuuden vähennystä olisi ylipäätään laskennallisesti jopa mahdotonta: jo pienikin, esimerkiksi 100×100-pikselikoon mustavalkokuvasyöte, vaatisi täysin kytketyssä perseptroniverkossa 10 000:n painon optimointia kutakin neuronaa kohti.



### 2.2.1. Kuvapiirteiden erottelu: kokoava konvoluutio



Kuva 3: Tyypillisen konvoluutiokerroksen toiminta: konvoluutio: 2x2 kernel, yksikköaskel, aktivaatio: ReLU, pooling: max

Yllä olevassa kuvassa 3 on esitetty konvoluutiokerroksen kolme laskennallista vaihetta, joita voidaan yhdistellä eri tavoin. Tyypillinen järjestys on seuraava:

1. Varsinainen konvoluutio
2. Aktivaatiofunktion soveltaminen eli epälineaarisuustekijä lisäys
3. Mahdollinen koonti/yhdistäminen (engl. *pooling*)

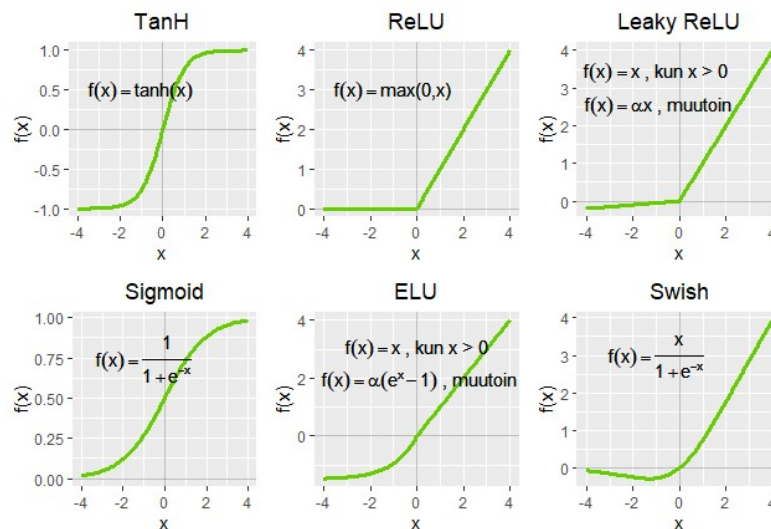
Ensimmäisessä, eli varsinaisessa konvoluutiiovaiheessa syötedata konvoloidaan niin kutsutun kernel-matriisin avulla. Tämä matriisi sisältää konvoluutioneuroverkkoa opetettaessa optimoitavat painot. Ensimmäisessä konvoluutiokerroksessa kernel-matriisi on kunkin neuronin reseptiivisen kentän kokoinen: se kerää syötteestä reseptiivisen kentän kattamien matriisialkiot, kertoo ne vastaavilla painoillaan ja summaa yhteen. Asetelma on esitetty yllä kuvassa 3. Kaksiulotteisessa tapauksessa kernel-matriisin  $A$  ja syöteen  $B$  välinen diskreetti konvoluutio voidaan esittää seuraavasti:

$$C(x, y) = \sum_{i=0}^{(h-1)} \sum_{j=0}^{(w-1)} A(i, j) B(x - i, y - j) \quad (1)$$

missä  $C(x, y)$  on syntyvä piirrekartta ja  $h$  ja  $w$  ovat kernel-matriisin dimensiot. Konvoluutiiovaiheen toimintaa määritteleviä parametreja ovat kernel-matriisin koon lisäksi askelpituus (engl. *stride*) ja reunojen täyttötapa (engl. *padding*). Askelpituus kuvaa sitä, montako alkiota kerrallaan kernel-ikkuna liukuu syötteen yli. Täyttötavalla puolestaan määritellään se, miten konvoluution halutaan käyttäytyvän syötteen reunoilla: jos kerneliä halutaan kierrättää tasapuolisesti koko varsinaisen syötteen yli,

on sen reunoihin lisättävä täytealkioita. Reunojen täyttö peilaamalla on esitetty kuvassa 5.

Toisessa vaiheessa edellä syntyneeseen matriisiin sovelletaan *aktivaatiofunktiota*, eli siihen lisätään jokin epälineaarisuus. Juuri aktivaatiofunktiot mahdollistavat neuroverkkojen kyvyn oppia epälineaarisia riippuvuuksia: ilman niitä verkon kerrokset olisivat vain sarja lineaarisia operaatioita. Tyypillisesti käytettyjä aktivaatioita ovat ReLU (*Rectified Linear Unit*)- ja Leaky ReLU-funktiot. ReLU vie syötematriisinsa negatiiviset soluarvot nollaan säilyttäen positiiviset muuttumattomana. Leaky ReLU toimii muutoin samoin, mutta se sallii myös negatiivisten arvojen vuodon pienellä kulmakertoimella. Nämä ja muita tyypillisiä aktivaatiofunktioita on esitetty alla kuvassa 4.



Kuva 4: Tyypillisiä aktivaatiofunktioita

Kolmannessa vaiheessa matriisin arvot yhdistetään eli kootaan jollakin tavoin. Tyypillisimmin käytetään niin kutsuttua *max pooling* -menetelmää, joka yksinkertaisesti poimii syötematriisin maksimiarvon. Yhdistämiseen voidaan käyttää myös keskiarvoistusta. Täysin konvolutiivisissa verkoissa tämä *pooling*-vaihe jätetään kokonaan pois ja korvataan perättäisillä konvoluutioilla.

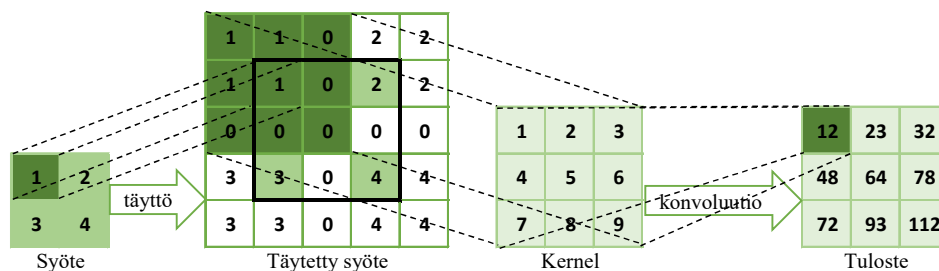
Käyttämällä erilaisia ja eri kokoisia kernel-matriiseja kuvasyötteestä saadaan eristettyä eri piirteitä. Yhden piirrekartan sijaan konvoluutiokerroksen tuotteena onkin piirrekarttojen sarja. Neuroverkkoa opetettaessa tavoitteena on löytää sellainen kernel-matriisien yhdistelmä, joka löytää syötedatasta halutut piirteet. Näiden piirteiden

perusteella kuvia voidaan esimerkiksi jakaa luokkiin, mitä käytetään hyväksi hahmontunnistuksessa.

### 2.2.2. Piirteiden uudelleenyhdistäminen: transponoitu konvoluutio ja täyttö

Hahmontunnistuksessa konvoluutioverkon tavoitteena on ennustaa kokonaiselle kuvalle yksi luokka. Kuvankäsittelyssä tarkoituksena kuitenkin on tuottaa alkuperäisen syötteen dimensioita vastaava muokattu esitys: "luokka" on siis ennustettava jokaiselle kuvan pikselille erikseen. Tähän tarvitaan transponoituja konvoluutiokerroksia, joita kutsutaan myös dekonvoluutiokerroksiksi. Erityisesti viimeksi mainittu nimi on kuitenkin harhaanjohtava, sillä kerroksessa suoritettava operaatio on sama konvoluutio kuin varsinaisessa konvoluutiokerroksessakin; vain sen suunta muuttuu. Siinä missä yllä perättäisiä konvoluutioita käytettiin erottelemaan kuvapiirteitä toisistaan, transponoiduissa kerroksissa näitä piirteitä yhdistellään uudelleen konvoluution avulla.

Koska tarkoituksena on niin ikään palata alkuperäiseen syötteen resoluutioon, yllä kokoavassa vaiheessa menetettyjen pikselien täyttämiseksi tarvitaan interpolointia. Käytännössä tämä tarkoittaa *täyttöä* ja sopivaa konvoluutiota. Alla olevassa kuvassa 5 on käytetty yksinkertaista nollatäyttöä ja reunoilla syötteen pikselien peilausta.



Kuva 5: Nollatäyttö, peilaus ja konvoluutio.

### 2.3. Kuva-aineisto neuroverkon syötteenä: tensorirakenteesta

Edellä on matriisiesityksen yksinkertaisuuden vuoksi käsitelty esimerkkejä, joissa neuroverkon syötedata on 2-dimensionaalista, esimerkiksi yksi mustavalkokuva. Todellisuudessa kuvasyöte on useimmiten vähintään 3- ja usein 4-ulotteinen tensori muotoa:

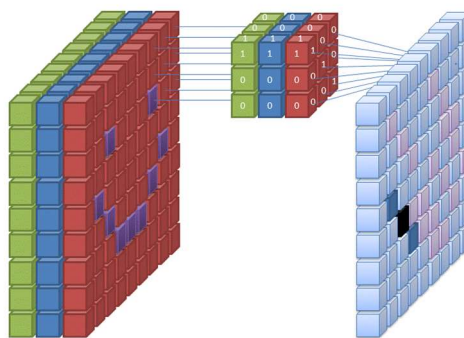
*(kuvan korkeus  $h$ )  $\times$  (kuvan leveys  $w$ )  $\times$  (kuvan syvyys  $d$ )  $\times$  (näytteiden määrä eli eräkkö  $n$ )*

Tai dekonvoluutiovaiheessa vastaavasti:

*(piirrekartan korkeus  $h$ )  $\times$  (piirrekartan leveys  $w$ )  $\times$  (piirteiden määrä  $d$ )  $\times$  (näytteiden määrä eli eräkkö  $n$ )*

Edellä esiteltyt periaatteet kuitenkin yleistyvät helposti useampaan dimensioon: esimerkki kahden kolmeulotteisen tensorin diskreetistä 2D-konvoluutiosta on visualisoitu kuvassa 6. Matemaattisesti se voidaan määritellä seuraavasti:

$$C(x, y) = \sum_{i=0}^{(h-1)} \sum_{j=0}^{(w-1)} \sum_{k=0}^{(d-1)} A(i, j, k) B(x - i, y - j, k) \quad (2)$$

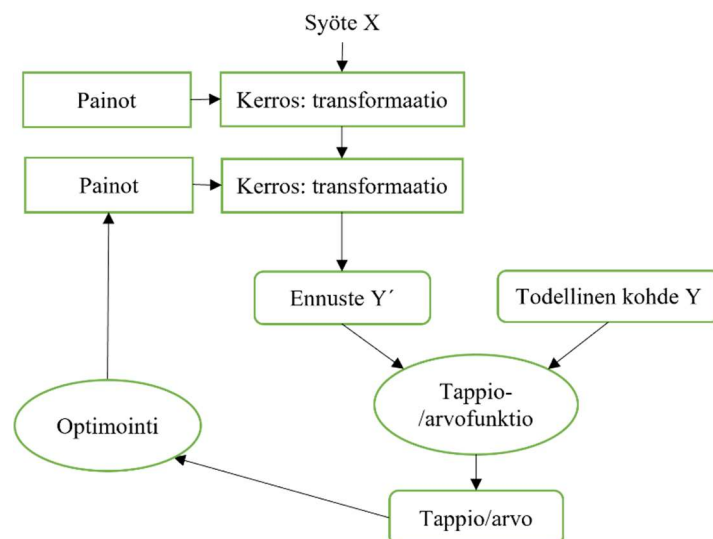


Kuva 6: 3D-tensorien 2D-konvoluutio: esimerkki RGB-värikanavien konvoluutiosta (Wikimedia-käyttäjä Cecbur, 2019).

Verkon optimointikaan ei yleensä perustu yksittäisiin kuviin vaan  $n$ -kokoisiin kuvaeriin (engl. *batch*). Erän koolla on merkitystä neuroverkon optimoinnissa erityisesti laskennallisen tehokkuuden kannalta ja se voi vaikuttaa myös oppimisen onnistumiseen. Eräkoon merkitykseen optimoinnissa palataan seuraavassa alaluvussa.

## 2.4. Oppiminen

Neuroverkkojen oppiminen perustuu malliin tallennettujen painojen optimointiin takaisinkytkennän avulla. Alla olevassa kaaviossa on esitetty neuroverkon optimoinnissa käytettävä takaisinkytkentärakenne. Seuraavissa alaluvuissa esitellään rakenteen olennaiset osat.



Kuva 7: Neuroverkon takaisinkytkentärakenne (mukaillen Chollet, 2018, s. 11).

### 2.4.1. Arvofunktio

Optimointia varten neuroverkko tarvitsee mekanismin, jonka avulla se arvioi suoriutumistaan. Tätä mekanismia kutsutaan arvofunktioksi tai tappiofunktioiksi, riippuen siitä, pyrkiikö verkko maksimoimaan vai minimoimaan funktion arvoa. Arvofunktion muotoilu riippuu käytetystä neuroverkon rakenteesta ja datan ominaisuuksista: kuvien tapauksessa se voi olla esimerkiksi todellisen kohteen ja verkon tuottaman ennusteen pikseliarvojen erotus.

## 2.4.2. Optimointi

### *Gradientti ja ketjusääntö*

Tietääkseen mihin suuntaan painoja on muutettava lopputuloksen optimoimiseksi, neuroverkko käyttää hyväkseen arvofunktion gradienttia. Kunkin verkon kerroksen painoja voidaan kuvata tensorina ja eri kerrosten painojen vaikutus lopputulokseen voidaan päätellä gradientin ketjusäännön avulla. Tensorimuotoisten muuttujien avulla ketjusääntö voidaan esittää seuraavasti:

$$\nabla_X z = \sum_j (\nabla_X Y_j) \frac{\partial z}{\partial Y_j} \quad (3)$$

missä  $z = f(Y)$ ,  $Y = g(X)$ ,  $Y$  ja  $X$  ovat peräkkäisten kerrosten painotensoreita ja summausindeksiin  $j$  on yhdistetty kaikki  $Y$ :n mahdolliset dimensiot (Goodfellow, Bengio & Courville, 2016). Gradientin laskentaan käytetään niin kutsuttua vastavirta-algoritmia (engl. *backpropagation*) (Rumelhart, Hinton & Williams, 1986): gradientti lasketaan ensin tulostekerroksen painojen suhteen, ja siitä edelleen kerros kerrokselta kohti syötekerrosta. Laskennassa samat osittaisderivaatat toistuvat useaan kertaan ja niiden evaluointi yhä uudelleen olisi laskennallisesti tehotonta. Vastavirta-algoritmissa tämä toisteisuus vältetään hyödyntämällä dynaamista laskentaa: kertaalleen lasketut derivaatat tallennetaan, jotta niiden arvoja voidaan käyttää suoraan myöhemmissä vaiheissa. (Goodfellow ym., 2016.)

### *Optimointialgoritmit ja hyperparametrit*

Neuroverkon optimointimenetelmät voidaan jaotella menetelmän hyödyntämän eräkoon (engl. *batch size*), ja toisaalta sen perusteella, kuinka paljon ja miten verkon painoja muutetaan kerralla. Eräkokoa ja oppimisnopeuteen liittyviä tunnuslukuja kutsutaan neuroverkon *hyperparametreiksi*. Erä koko kertoo, kuinka paljon opetus-esimerkkejä verkko käyttää yhdellä opetuskierroksella. Sen perusteella optimointimenetelmät jaetaan tyypillisesti kolmeen tyyppiin: *stokastiseen*-, *batch*- ja *mini-batch*-menetelmään. Stokastisessa menetelmässä verkolle syötetään yksi sattumanvarainen opetus-esimerkki kerrallaan, lasketaan vastaava arvo/tappio ja sitä vastaava gradientti, sekä päivitetään painot. Batch-menetelmässä puolestaan koko opetusdata ajetaan verkon läpi kerralla ja kokonaisarvo/tappio lasketaan yksittäisten

esimerkkien keskiarvona. Viimeisimpänä, mini-batch on yhdistelmä kahta edellistä: siinä opetusdata jaetaan pieniin eriin ja painojen päivitys tehdään aina yhden aineistoerän perusteella. Jako voidaan tehdä joko deterministisesti tai stokastisesti. Menetelmästä riippumatta koko opetusaineisto yleensä ajetaan verkon läpi useaan kertaan. Yhtä tällaista, koko opetusaineiston kattavaa kierrosta kutsutaan *epookiksi*.

Kun jokin ylläesitetty joukko opetusesimerkkejä on ajettu verkon läpi ja arvofunktion gradientti laskettu, tulee valita, kuinka voimakkaasti painoja muutetaan gradientin perusteella. Yksinkertaisimmassa, tavallisessa *laskevan gradientin menetelmässä* (engl. *gradient descent*) on vain yksi parametri, oppimismnopeus (engl. *learning rate*)  $\alpha$ , jonka mukaisessa suhteessa painoja muutetaan laskevan tai nousevan gradientin suuntaan (arvofunktion muotoilusta riippuen).

$$Y_t = Y_{t-1} - \alpha \nabla_Y \mathcal{L}(Y_{t-1}) \quad (4)$$

Kaikessa yksinkertaisuudessaan laskevan gradientin menetelmä on kuitenkin epäluotettava: liian pienillä painoilla algoritmi jää helposti jumiin lokaaliin minimiin ja liian suurilla se voi divergoida täysin. Edistyneemmissä menetelmissä hyödynnetäänkin yksityiskohtaisempaa informaatiota gradientista. Esimerkiksi niin kutsutussa *momenttimenetelmässä* käytetään hyväksi myös edeltävän optimointikierroksen  $t - 1$  tuloksia; momentti "hyväksi havaittuun" muutossuuntaan säilyy. Momentin voimakkuutta kuvaa parametri  $\gamma$ .

$$\begin{aligned} V(t) &= \gamma V(t-1) + \alpha \nabla_Y \mathcal{L}(Y_{t-1}) \\ Y_t &= Y_{t-1} - V(t) \end{aligned} \quad (5)$$

Momentti-idea soveltaa myös nykyisin käytössä olevista optimointimenetelmistä suosituin, Adam (*Adaptive moment estimation*) -algoritmi. Se hyödyntää sekä arvofunktion gradienttia että sen neliötä ja päivittää jokaisella optimointikierroksella näiden eksponentiaalisesti liukuvat keskiarvot  $\hat{m}$  ja  $\hat{v}$ . Nämä keskiarvot edustavat gradientin ensimmäistä momenttia ja toista raakamomenttia ja niiden hajoamisnopeutta kuvaavat menetelmät parametrit  $\beta_1$  ja  $\beta_2$ . (Kingma & Ba, 2014.)

$$\begin{aligned}
\hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, \text{ missä } m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_Y \mathcal{L}_t(Y_{t-1}) \\
\hat{v}_t &= \frac{v_t}{1 - \beta_2^t}, \text{ missä } v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_Y \mathcal{L}_t(Y_{t-1}))^2 \\
Y_t &= \frac{Y_{t-1} - \alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}
\end{aligned} \tag{6}$$

### Normalisointi

On helppoa ymmärtää, miksi neuroverkon alkuperäisen opetusaineiston normalisointi on tärkeää: jos esimerkiksi syötekuvien suhteellinen intensiteetti vaihtelee aineistossa, sen perusteella oppiminen on hidasta, ellei mahdotonta. Saman normalisointitarpeen voidaan katsoa koskevan jokaista neuroverkon kerrosta erikseen. Tätä varten neuroverkon kerroksissa käytetään niin kutsuttua eränormalisointia (engl. *batch normalization*). Sen tarkoituksena on muokata kunkin kerroksen tuottamaa *aktivaatiota* eli tulostetta niin, että se seuraavan kerroksen syötteenä toimiessaan noudattaa tiettyä jakaumaa, esimerkiksi standardoitua normaalijakaumaa (Ioffe & Szegedy, 2015).

## 2.5. Haasteita

### 2.5.1. Laskennallinen alivuoto ja katoavat gradientit

Syvien neuroverkkojen optimoinnissa ongelmana ovat katoavat gradientit. Ongelma korostuu erityisesti tiettyjä aktivaatiofunktioita käytettäessä: esimerkiksi tanh-funktion derivaatta saa arvoja vain väliltä  $[0,1]$ , ja tällaisten derivaattojen ketjutus vastavirta-algoritmissa saa aikaan sen, että gradientti pienenee lopulta niin, että sitä ei enää saada laskettua. Tämän vuoksi muiden aktivaatiofunktioiden kuten ReLU:n käyttö on yleisempää. Myös verkon painojen sopivalla alustuksella on roolinsa alivuodon välttämässä. Tarvittaessa voidaan käyttää logaritmista asteikkoa.



### 2.5.2. Ylisovittaminen

Kaiken datalähtöisen mallinnuksen ydinongelma on ylisovittaminen: malli, joka on optimoitu toimimaan täydellisesti opetusaineistossaan, ei välttämättä käyttäydy halutulla tavalla uuden opetusaineiston ulkopuolisen datan kanssa. Toisin sanottuna malli voi oppia liian tarkasti rajallisen opetusaineiston erityispiirteet. Neuroverkkojen tapauksessa ylisovittamisen ongelma tyypillisesti korostuu verkon kerrossyvyyden kasvaessa; neuroverkko pystyy tällöin oppimaan opetusdatansa yksityiskohdat liiankin tarkasti. Verkon syvyyttä ei siis ole tarkoituksenmukaista kasvattaa rajattomasti.

Tapauksissa, joissa käytettävissä olevaa opetusdataa on erityisen vähän, ylisovittamista voidaan välttää myös datan *augmentoinnilla*. Tällä tarkoitetaan sitä, että opetusesimerkkejä tuotetaan lisää alkuperäistä dataa hienovaraisesti muokkaamalla. Kuvien tapauksessa tämä voi tarkoittaa esimerkiksi skaalausta, rajausta, kääntöä, kiertoa sekä kirkkauden tai kontrastien säätöä.

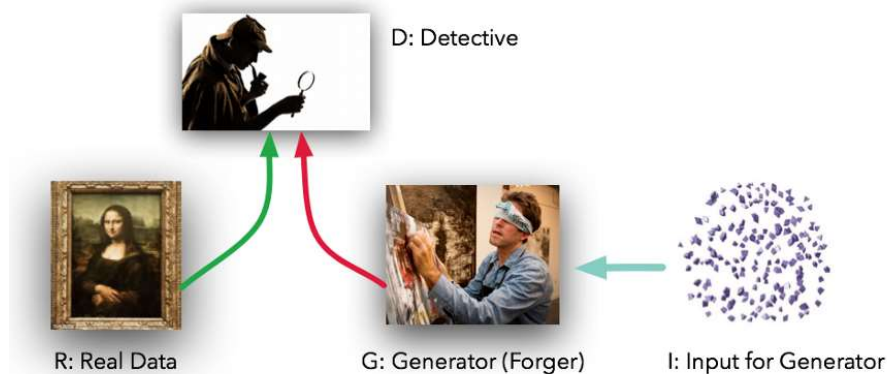
### 2.5.3. Arvofunktion valinta

Neuroverkkojen opetuksessa sopivan arvofunktion valinta ei ole aina yksiselitteistä. Arvofunktion valinnassa on huomioitava, että neuroverkko – optimaalisesti toimiessaan – löytää aina lyhimmän polun, joka maksimoi valitun funktion. Esimerkiksi kahden kuvan pikseliarvojen keskimääräisen erotuksen käyttö arvofunktiona johtaa todennäköisesti hyvin sumentuneeseen lopputulokseen. Joissakin tapauksissa voidaan näennäisesti käyttää useita arvofunktioita, mutta loppujen lopuksi niiden tulokset on joka tapauksessa yhdistettävä esimerkiksi keskiarvoistamalla yhdeksi ainoaksi arvoksi, johon optimointi perustetaan. Seuraavaksi esiteltävän kilpailevien neuroverkkojen arkkitehtuurin avulla "kovakoodatun" arvofunktion valinta voidaan kuitenkin välttää.

### 3. Kilpailevat neuroverkot eli GAN:it

#### 3.1. Rakenne ja toiminta

GAN (*Generative Adversarial Network*) eli generatiivinen kilpaileva verkosto on Goodfellow ym. vuonna 2014 esittelemä neuroverkkoarkkitehtuuri, jonka toiminta perustuu nollasummapeleihin. Generatiivinen kilpaileva verkosto koostuu kahdesta toisiaan vastaan kilpailevasta osasta: generatiivisesta ja diskriminoivasta neuroverkosta, joista jäljempänä käytetään vastaavasti termejä *generaattori* ja *diskriminaattori*. Generaattorin tarkoitus on oppia luomaan uutta, mahdollisimman aidonoloista dataa ja diskriminaattorin tehtävänä on oppia erottamaan aito data generaattorin tuottamasta. Nollasummapelellin hengen mukaisesti kilpailevat neuroverkot pakottavat toisiaan jatkuvasti parempiin suorituksiin: pelin edetessä generaattori oppii aidon datan todennäköisyysjakauman yhä tarkemmin ja diskriminaattori oppii erottamaan aidon jakauman tuottaman datan synteettisestä yhä paremmin. (Goodfellow ym., 2014.)



Kuva 8: GAN:in toiminta pähkinäkuoressa (Nag, 2017).

GAN:in arvofunktio – kuten se on esitetty alkuperäisessä julkaisussaan (Goodfellow ym., 2014) – on seuraava:

$$\begin{aligned} \min_G \max_D \mathcal{L}_{\text{GAN}}(D, G) \\ = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log (1 - D(G(z)))] \end{aligned} \quad (7)$$

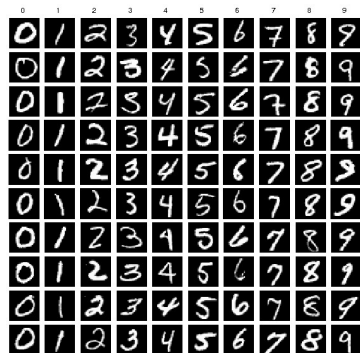
Arvofunktiossa  $D(x)$  edustaa todennäköisyyttä sille, että datapiste  $x$  on peräisin alkuperäisestä jakaumasta ja vastaavasti  $1 - D(G(z))$  todennäköisyyttä sille, että generaattorin  $G$  datapisteestä  $z$  tuottama versio  $G(z)$  ei ole peräisin alkuperäisestä jakaumasta. Generaattorin syötteenä toimiva  $z$  on esimerkiksi Gaussista kohinaa. Diskriminaattori pyrkii maksimoimaan oikeita luokituksiaan, eli näiden kahden odotusarvon summaa. Generaattori puolestaan pyrkii minimoimaan jälkimmäistä termiä harhauttamalla diskriminaattoria luulemaan tuotoksiaan aidoiksi. Todellisuudessa generaattorin arvofunktiolle joudutaan usein käyttämään muotoa  $\max_G \log D(G(z))$  olevaa lauseketta, sillä diskriminaattorin etumatka on alussa niin valtava, että  $\min_G \log 1 - D(G(z))$  saturoituu nolnaan, eikä sen gradienttia siten saa laskettua. (Goodfellow ym., 2014.)

### 3.2. Ehdollinen generointi: GAN:ista cGAN:iin

Pian kilpailevien neuroverkkojen idean julkaisun jälkeen niistä alettiin kehitellä eri tarkoituksiin sopivia variaatioita. Jo alkuperäisjulkaisun yhteydessä ehdotettiin uutta versiota: niin kutsuttua ehdollista kilpailevaa verkostoa (*conditional Generative Adversarial Network*, *cGAN*). Siinä missä tavallisen GAN:in generaattorin tarkoitus on oppia luomaan uutta dataa silkasta kohinasta  $G: z \rightarrow x$ , cGAN generaattorin tehtävänä on oppia kuvaus  $G_c: (y, z) \rightarrow x$ , missä  $y$  on haluttu *luokkatunnus*. Arvofunktio saa tällöin muodon

$$\begin{aligned} \min_G \max_D \mathcal{L}_{\text{cGAN}}(D, G) \\ = \mathbb{E}_{x \sim p_{\text{data}}(x)} \log[D(x|y)] \\ + \mathbb{E}_{z \sim p_z(z)} [\log (1 - D(G(z|y)))] \end{aligned} \quad (8)$$

Mirzan ja Osinderon (2014) esittelemä malli pystyi esimerkiksi generoimaan "käsin kirjoitettuja" numeroita annettujen luokkatunnusten, eli syötteenä annettujen kokonaislukujen perusteella. Neuroverkon opettamiseen käytettiin koneoppimispiireissä tunnettua MNIST (*Modified National Institute of Standards and Technology*) -aineistoa (kuva 9), joka koostuu yhteensä 70 000:sta käsin kirjoitettua numeroa esittävästä kuvasta. (Mirza & Osindero, 2014.)



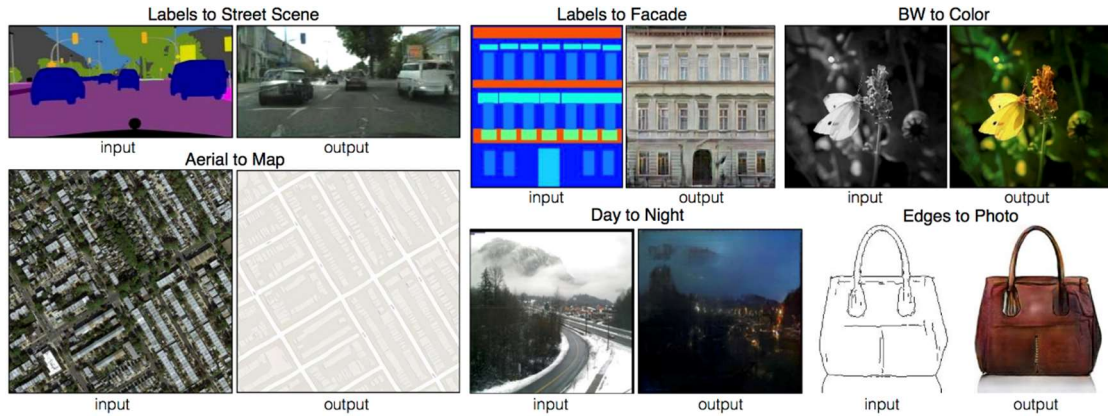
Kuva 9: MNIST-aineistoa (Lim, Young & Patton, 2016).

### 3.3. Soveltaminen kuvankäsittelyyn

Kattava yhteenvedon erilaisista kuvankäsittelyyn soveltuvista GAN-variaatioista ovat esittäneet esimerkiksi Huang, H., Yu ja Wang (2018). Seuraavaan on koottu magneettikuvapainotuksien syntetisointiin kirjallisuudessa usein sovellettuja menetelmiä.

#### 3.3.1. Pikselikohtainen häviö: pGAN eli Pix2pix

Yllä kuvatun ehdollisen cGAN:in luokkatunnusten ei välttämättä tarvitse olla atomisia, kuten kokonaislukuja, vaan ne voivat olla myös kuva-aineistoa. Sitten cGAN:ejä onkin sovellettu mitä moninaisimpiin kuvankäsittelytarkoituksiin. Esimerkiksi Isola, Zhu, Zhou ja Efros (2017) raportoivat menetelmän onnistuneesta käytöstä useissa erilaisissa kuvasta-kuvaksi-tyyppisissä sovelluskohteista, joita on esitetty alla kuvassa 10.



Kuva 10: Esimerkkejä cGAN:in kuvasta-kuvaksi -sovelluksista (Isola ym., 2017).

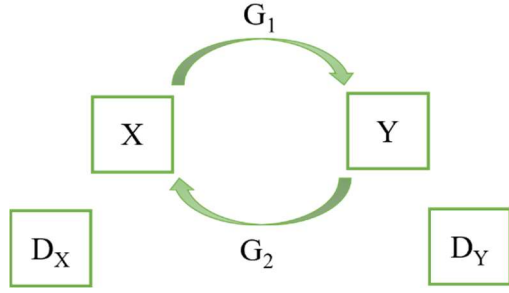
Näissä sovelluksissa generaattorin toimintaa ohjataan lisäämällä arvofunktiioon pikselikohtainen häviö, joka voidaan laskea joko L1- tai L2-normina. Arvofunktio on tällöin:

$$\min_G \max_D \mathcal{L}_{\text{pGAN}}(D, G) = \mathcal{L}_{\text{cGAN}}(D, G) + \lambda_{\text{L1/L2}} \mathcal{L}_{\text{L1/L2}}(G) \quad (9)$$

missä  $\lambda_{\text{L1/L2}}$  on pikselikohtaisen häviön painoarvoa kuvaava parametri. Mustavalkokuvien tapauksessa L1- ja L2-normit ovat yksinkertaisesti pikseliarvojen erotusten, tai niiden neliöiden summa. L1-normin käyttö tuottaa kuviin tyypillisesti vähemmän sumennusta (Isola ym., 2017). Pikselikohtaisen häviön hyödyntäminen vaatii kuitenkin parillisen ja täysin rekisteröidyn opetusaineiston käyttöä.

### 3.3.2. Rekonstruktiohäviö: CycleGAN

Syklinen kilpaileva verkosto eli CycleGAN vastaa tarpeeseen opettaa neuroverkko ilman parillista opetusaineistoa: siinä optimointi perustuu normaalin GAN-arvofunktion ohella syklisen johdonmukaisuuden (engl. *cycle consistency*) tavoitteluun. Käytännössä CycleGAN rakentuu kahdesta cGAN-verkostosta, joista ensimmäisen tarkoitus on oppia transformaatio datajoukosta X joukkoon Y ja toisen päinvastaiseen suuntaan.



Kuva 11: CycleGAN:in rakenne. G vastaa generaattoria ja D diskriminaattoria. X ja Y ovat datajoukkoja.

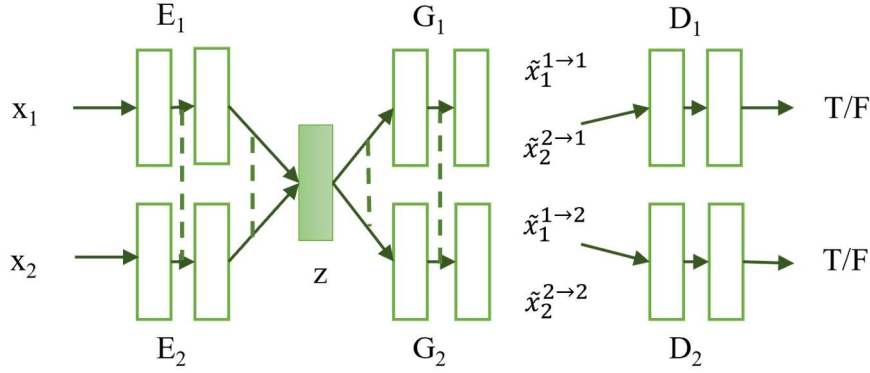
Näin ollen toisen verkoston generaattori  $G_2$  voi tuottaa ensimmäisen verkoston generaattorin  $G_1$  tuottamasta kuvasta rekonstruktion  $G_2(G_1(x))$ . CycleGAN:in arvofunktion toisen häviötermin avulla pyritään minimoimaan tämän rekonstruktion poikkeamaa alkuperäisestä syötteestä. Kokonaisuudessaan arvofunktio on muotoa:

$$\begin{aligned} \min_{G_1, G_2} \max_{D_X, D_Y} \mathcal{L}_{\text{CycleGAN}}(G_1, G_2, D_X, D_Y) \\ = \mathcal{L}_{\text{cGAN}}(G_1, D_Y, X, Y) + \mathcal{L}_{\text{cGAN}}(G_2, D_X, Y, X) \\ + \lambda_{\text{cyc}} \mathcal{L}_{\text{cyc}}(G_1, G_2) \end{aligned} \quad (10)$$

missä  $\lambda_{\text{cyc}}$  on syklisen häviön painoarvoa kuvaava parametri. CycleGAN:in rakenteen esittelivät Zhu, Park ja Isola (2017).

### 3.3.3. Erilliset enkooderit: UNIT

Toinen parittomaan kuvasynteesiin soveltuva, tarkoituksensa mukaisesti nimetty verkkoarkkitehtuuri UNIT (*Unpaired Image-to-Image Translation*) on esitetty alla kuvassa 12. Se yhdistää kaksi GAN-verkostoa ja kaksi VAE (*Variational Auto Encoder*) -enkooderia. Enkooderit vastaavat siitä, että GAN-verkostojen syötteet tulevat samasta *latenttilasta*  $z$ . Tämä saadaan aikaan pakottamalla kummankin enkooderin viimeisten ja kummankin generaattorin ensimmäisten kerrosten painot vastaamaan toisiaan. Tällaisen latenttilan olemassaolo on voimakas oletus, johon myös yllä esitellyn CycleGAN:in rekonstruktiohäviö täsmällisesti ottaen nojaa. (Liu, Breuel & Kautz, 2017.)



Kuva 12: UNIT-rakenne (mukaillen Liu ym., 2017). G ja D ovat generaattori ja diskriminaattori kuten yllä ja E on enkooderi.  $x_1$  ja  $x_2$  ovat datajoukkoja ja  $z$  niiden oletettu yhteinen latenttitila.

### 3.3.4. Bayesilainen lähestymistapa: KL-divergenssihäviö

Niin kutsutuissa Bayesilaisissa neuroverkoissa optimointiin tuodaan mukaan stokastiikkaa esimerkiksi *dropout*-kerrosten avulla. Dropout-kerrosten tarkoituksena on sattumanvaraisesti katkaista verkon neuronien välisiä yhteyksiä ja siten hävittää osa jo opitusta informaatiosta. Tällä saadaan aikaan se, että neuroverkon painoilla voidaan ajatella determinististen arvojen sijaan olevan jokin tietty jakauma. (Zhaoa, Meyerand & Birn, 2020.) Tällöin arvofunktiossa voidaan pyrkiä minimoimaan kahden todennäköisyysjakauman vertailuun pohjautuvaa KL (Kullback-Leibler) -divergenssiä (Kullback & Leibler, 1951), jota vastaava arvofunktion häviötermi on muotoa:

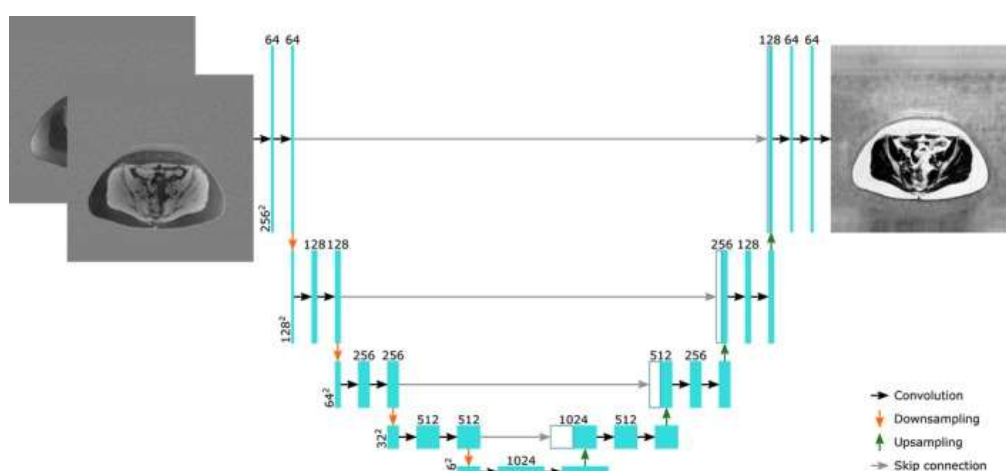
$$\hat{\mathcal{L}}_{\text{MC}}(\theta) = -\frac{N}{n} \sum_{i \in S} \log p(y_i | f^{g(\theta, \epsilon)}(x_i)) + \sum_{\text{Dropout}} \frac{1}{2} l^2 (1 - p) \|M\|^2 - K \mathcal{H}(p) \quad (11)$$

missä  $\theta$  on verkon painojen kokoelma,  $N$  on kaikkien havaintojen määrä,  $n$  (alkuperäisessä:  $M$ ) ko. opetuskierroksella käytettävien havaintojen määrä,  $g(\theta, \epsilon)$  on painoja edustavien satunnaismuuttujien kokoelma,  $x_i$  on yksi opetusesimerkki ja  $y_i$  sitä vastaava tavoitejakauman esimerkki,  $f^{g(\theta, \epsilon)}(x_i)$  on ennuste verkon tuottamalle tulosteelle,  $l$  on hyperparametri,  $M$  on dropout-kerrosta edeltävän kerroksen painojen matriisi,  $K$  on dropout-kerroksen syötekanavien määrä ja  $\mathcal{H}$  on Bernoullin jakaumaa noudattavan satunnaismuuttujan entropia todennäköisyydellä  $p$ :  $\mathcal{H} = -p \log p - (1 - p) \log(1 - p)$ , missä edelleen  $p$  on dropout-todennäköisyys. (Zhaoa ym., 2020.)

### 3.3.5. Muita arvofunktion lisätermejä

Edellä esitettyjen lisäksi neuroverkon oppimista voidaan ohjata muillakin arvofunktion lisätermillä. Esimerkiksi Yu, Zhou, Wang, Shi, Fripp ja Bourgeat (2019) saivat lupaavia tuloksia niin kutsutulla EA (*Edge-Aware*) -GAN:illa, jonka arvofunktioon oli sisällytetty 3D-Sobel-reunantunnistusfilteriin pohjautuva rakennehäviö. Liu (2018) puolestaan käytti segmentointitarkkuuteen pohjautuvaa häviötä.

## 3.4. Generaattorin arkkitehtuuri

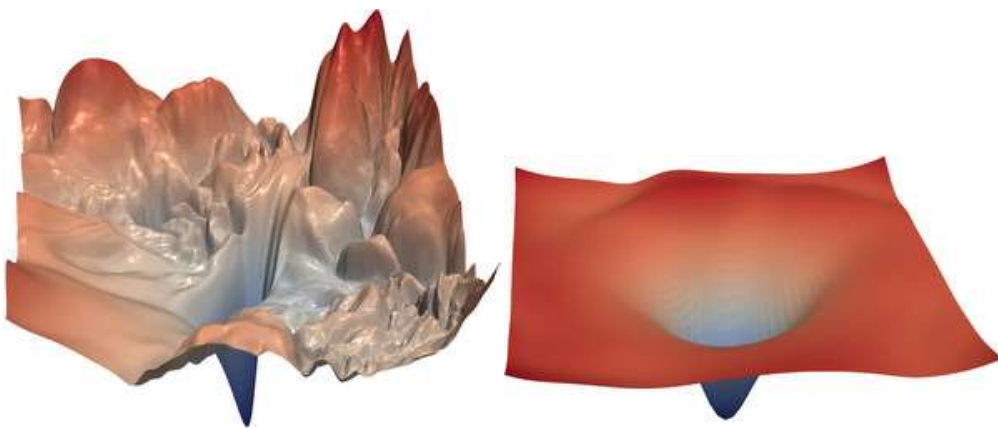


Kuva 13: U-net-arkkitehtuuri (Andersson, Ahlström & Kullberg, 2019). Palkin vasemmassa alakulmassa piirrekartan pikselikoko, yläpuolella piirteiden määrä.

Kuten edeltävässä luvussa on perusteltu, konvoluutioneuroverkot soveltuvat erityisen hyvin kuvien prosessointiin. Kuvankäsittelyyn suunnitelluissa GAN:eissa generaattorin arkkitehtuuri onkin tyypillisesti jonkinlainen kahden konvoluutioverkon symmetrinen yhdistelmä ja noudattaa oleellisilta osiltaan niin kutsuttua U-net -rakennetta (Ronneberger, Fischer & Brox, 2015). U-net -arkkitehtuuri on saanut nimensä U-kirjainta muistuttavasta kaksihaarisesta rakenteestaan, joka on esitetty yllä kuvassa 13. U:n vasenta haaraa alas mentäessä verkon kerrosten syötedatan resoluutio pienenee (kuvassa palkin korkeus) ja eroteltujen piirteiden määrä kasvaa (kuvassa palkin leveys) perättäisten kokoavien konvoluutio- ja *pooling*-operaatioiden seurauksena. Oikeassa haarassa resoluutio vastaavasti kasvaa ja piirteitä yhdistetään toisiinsa.



Kuvien synteessissä suuri osa piirteistä on sellaisia, joiden halutaan säilyvän transformaatioissa muuttumattomana: esimerkiksi selkeät reunat. Tällaisten piirteiden siirtoa varten U-net rakenteessa käytetään niin kutsuttuja ohitusyhteyksiä (engl. *skip-connection*), joita edustavat yllä olevassa kuvassa harmaat poikkiviivat. Erottelevan haaran tuottamia piirrekarttoja yhdistetään niiden kautta suoraan vastaavaan kokoavaan vaiheeseen. Ohitusyhteyksien käytöllä saavutetaan etuja paitsi laskentatehokkuudessa, myös lopputulosten laadussa (Isola ym., 2017). Erityisesti ne helpottavat oppimista silottamalla arvofunktion optimointimaisemaa (Li, Xu, Taylor, Studer & Goldstein, 2018).

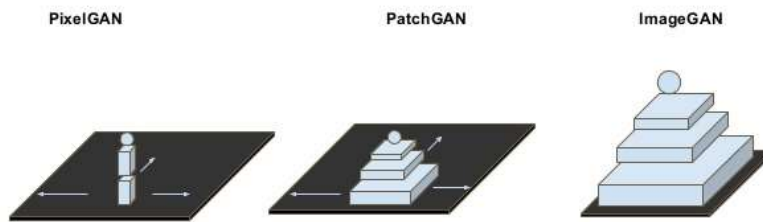


Kuva 14: Esimerkki arvofunktion visualisoinnista ilman ohitusyhteyksiä ja niiden kanssa (Li ym., 2018).

### 3.5. Diskriminaattorin arkkitehtuuri

Diskriminaattorin arkkitehtuuri on generaattoria yksinkertaisempi, sillä sen tarvitsee tuottaa vain yksiulotteinen tuloste: todennäköisyys sille, että syötekuva kuuluu johonkin kuvajoukkoon. Kokonaisrakenteeltaan diskriminaattori on yksinkertainen usein vain muutaman kerroksen konvoluutioneuroverkko, viimeisenä elementtinään jokin aktivaatiofunktio, joka tuottaa arvon väliltä  $[0,1]$ . Diskriminaattorin syötekerroksen koko suhteessa syötedataan kuitenkin ratkaisee, millä tasolla se syötettä arvioi. Niin kutsutussa ImageGAN-diskriminaattorissa syöteen vastaanottava kerros kattaa koko kuvan. Tyypillisesti käytetyssä PatchGAN-generaattorissa puolestaan syöte jaetaan  $N \times N$  kokoisiin ikkunoihin, joiden yli diskriminaattori liikkuu. Lopullinen ennuste lasketaan kuvaikkunoiden keskiarvona. Sama koskee pixelGAN:iä; siinä alkuperäinen kuvaikkuna vain on yhden pikselin kokoinen.

## Discriminator - Patch GAN



Kuva 15: Diskriminaattorin arkkitehtuurin vaihtoehtoja (Garcia, 2016).

### 3.6. Haasteita

Aiemmin esitettyjen tavanomaisten neuroverkkojen ongelmakohtien lisäksi kilpailevien neuroverkkojen opettamisessa on omat haasteensa. Yksi tyypillisimmistä ongelmista johtuu diskriminaattorin ylivoimaisesta asemasta. Erityisesti opetusprosessin alkuvaiheessa diskriminaattorin tehtävä erottaa generaattorin tuotokset alkuperäisestä datajoukosta on verrattain helppo verrattuna generaattorin tehtävään, mikä johtaa suuriin eroihin arvofunktion gradienttien jyrkkyydessä. Tämä pyritään välttämään hidastamalla diskriminaattorin oppimista päivittämällä sen painoja harvemmin, vain murto-osassa generaattorin opetuskierroksia (Goodfellow ym., 2014).

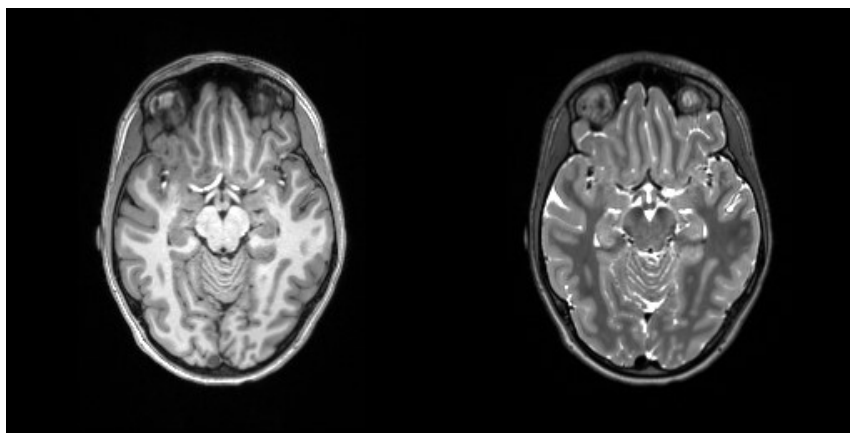
Erityisesti parittoman opetusaineiston tapauksessa, jolloin tarkoitus on generoida uutta aineistoa, mallin valmiutta, eli opetusprosessin konvergenssia on vaikea arvioida. Yksi keino on tarkkailla generaattorin ja diskriminaattorin häviöiden suhteellista käyttäytymistä. Esimerkiksi CycleGAN:in tapauksessa on mahdollista tarkastella myös rekonstruktion poikkeamaa alkuperäisestä syötteestä.

Kenties vaikeimmin ratkaistavissa oleva GAN:in ongelma on generaattorin tuottaman datajakauman romahtaminen yksittäisiin tavoitejakauman moodeihin (engl. *mode collapse*). Tämä on erityisesti puhtaan GAN:in ongelma; mikäli generaattorin ainoana tavoitteena on harhauttaa diskriminaattoria, tarvitsee sen keskittyä vain yhteen toimivaksi todettuun moodiin. Ongelma voidaan yrittää välttää lisäämällä verkkoon dropout-kerroksia, joissa sattumanvaraisesti katkaistaan neuronien välisiä yhteyksiä. (Srivastava, Hinton, Krizhevsky, Sutskever & Salakhutdinov, 2014).

## 4. GAN:ien soveltaminen magneettikuvapainotusten syntetisoinnissa

### 4.1. Magneettikuvien painotukset

Magneettikuvaus perustuu protonien, eli vetyatomien spinin käyttäytymiseen muuttuvassa magneettikentässä. Magneettikuvauslaite mittaa voimakkaalla magneettikentällä virittämiensä protonien vastetta radiotaajuiselle signaalille. Pehmytkudoksissa, kuten aivoissa, saatu signaali riippuu paitsi kudoksen protonitiheydestä, myös protonien vuorovaikutuksista ympäristönsä kanssa. Näitä vuorovaikutuksia saadaan esiin varioimalla kuvauslaitteen lähettämän radiosignaalin pulssitaajuutta. Tuloksena on eri tavoin painotettuja kuvia, joissa kussakin tietyt kudostyypit erottuvat selkeimmin. Kuvissa pikselin kirkkaus edustaa mitatun signaalin intensiteettiä. Tyypillisimpiä painotuksia, eli kontrasteja ovat T1-, T2-, ja PD-painotukset, joista tässä työssä keskitytään ensisijaisesti kahteen ensin mainittuun niiden diagnostisen tärkeyden vuoksi.



Kuva 16: T1- ja T2-painotetut pään magneetikuvat.

T1-painotus perustuu verrokiaan tiheämpään pulssitaajuuteen (lyhyempään toistoaikaan). T1-kuvassa kirkkaimpina erottuu rasvakudos. Pään alueella sitä on erityisesti ihossa ja aivojen keskiosan valkeassa aineessa, sillä se sisältää runsaasti rasvapitoista myeliiniä. Rasvattomat, vesipitoiset kudokset, kuten aivo-selkäydinneste näkyvät kuvassa mustana.

T2-painotus puolestaan perustuu vastaavasti pienempään pulssitaajuuteen ja siinä kirkkaana erottuvat nimenomaan vesipitoiset kudokset. Rasvakudokset näkyvät harmaan eri sävyissä. Kahden edellä mainitun ohella kirjallisuudessa, johon tutkielmassa viitataan, on käsitelty myös niin kutsutun T2-FLAIR (*Fluid-Attenuated Inversion Recovery*) -painotuksen syntetisointia. T2-FLAIR-kuvat perustuvat samaan T2-pulssitaajuuteen, mutta niissä signaalia on muutettu siten, että aivo-selkäydinnesteen intensiteetti ei olisi niin korostunut.

Aivoissa pääosaan patologisista prosesseista liittyy nesteen kertyminen, mikä erottuu parhaiten epänormaalin korkeana signaalina T2-kuvissa (van der Plas, 2015). Tämä ei päde ainoastaan kasvaimiin, vaan myös esimerkiksi MS-taudin taustalla oleva demyelinaatio erottuu epänormaalina nestepitoisuutena (Given, Stevens & Lee, 2004). Molemmilla painotuksilla on kuitenkin oma diagnostinen arvonsa: T1-kuvassa esimerkiksi aivokuori erottuu valkeasta aineesta huomattavasti selkeämmin kuin T2-kuvassa.

## **4.1. Lääketieteellisten kuvien synteeseistä**

### **4.1.1. Tarpeet**

GAN:eillä on lääketieteellisen kuvantamisen kontekstissa hyvin monipuoliset sovellusmahdollisuudet. Kuten Yi, Walia ja Babyn (2019) ovat kattavassa yhteenvedossaan näyttäneet, samankaltaisia menetelmiä kuin tässä tutkielmassa on esitetty voidaan soveltaa myös esimerkiksi kuvantamisaineiston rekonstruktioon (häiriöiden poistoon), segmentointiin, luokitteluun sekä esimerkiksi eri *modaliteettien* väliseen rekisteröintiin tai niiden yhdistämiseen diagnostisia tarkoituksia varten. Tässä modaliteeteilla viitataan paitsi magneettikuvapainotuksiin myös muilla kuvantamismenetelmillä tuotettuihin toisiaan anatomisesti vastaaviin kuviin.

Modaliteettisynteessin pääasiallisena tavoitteena on luoda aineistoa tutkimustyöhön ja koneoppimisen kehittämiseen. Esimerkiksi koneoppimiseen pohjautuvan segmentoinnin tai automaattisen diagnostiikan kehittäminen on hyvin dataintensiivistä ja tietyn tyyppisistä patologioista dataa on hyvin vähän saatavilla. Saatavuutta rajoittavat entisestään kuvantamisen kalleus ja suostumukselliset seikat (Yi ym., 2019).

Syntetisoiduista kuvista voi olla hyötyä myös kliinisessä diagnostiikassa, jos kuvantamisaineisto on muuten puutteellista tai osa kuvista on häiriöiden vuoksi käyttökelvottomia.

#### **4.1.2. Kilpailevien verkkojen vahvuuksia**

Ensisilmäyksellä saattaisi vaikuttaa siltä, että mustavalkoisen magneettikuvan kuvaus toiseksi ei vaatisi mitään lineaarista regressiota monimutkaisempaa, mutta kuten taustalla oleva fysiikkakin antaa olettaa, eri painotuksilla tuotettujen kuvien suhde on hyvin epälineaarinen. Tämä on helppo huomata silmämääräisestikin esimerkiksi siitä, että muista poikkeavuuksista huolimatta molemmissa kuvissa aivokuoren sävy on likimain sama. Syvän koneoppisen erityispiirteenä on sen erinomainen kyky mallintaa juuri tällaisia epälineaarisia, spatiaalisesti vaihtelevia riippuvuuksia, joita muilla menetelmin on vaikeaa saada esiin.

Toinen merkittävä syy tavanomaisten pikselistä-pikseliksi-tyyppisten regressiomenetelmien toimimattomuudelle on se, että lääketieteellisten kuvien tapauksessa opetusdata ei todellisuudessa ole koskaan täysin parillista, vaan eri painotuksilla otetut kuvat joudutaan aina rekisteröimään toisiinsa laskennallisesti. Rekisteröinnillä tarkoitetaan sitä, että kahden kuvan pikselit pyritään saamaan vastaamaan toisiaan mahdollisimman täsmällisesti. Rekisteröinti on kuitenkin aina jossain määrin epätäydellistä, ja siten kuvien pikseliarvoja ei voi täysin luotettavasti verrata toisiinsa (Dar, Yurt, Karacan, Erdem A., Erdem E., & Cukur, 2019).

Sen lisäksi, että generatiivisen kilpailevan verkoston tuotteena saadaan generaattori kuvien syntetisointi varten, myös koulutetun diskriminaattorin hyödyntämismahdollisuuksista uuden aineiston luokittelussa on tutkittu muissa konteksteissa (Mao X., Su, Tan, Chow & Wang, 2019). Jos verkoston opetuksessa käytetty aineisto on esimerkiksi koostunut normaaleista, terveistä kuvista, diskriminaattori voisi kyetä erottamaan epänormaalit kuvat myös uudesta aineistosta.

## 4.2. Tulosten arviointi

### 4.2.1. Kvantitatiivinen arviointi: tunnuslukuja

Kun käytettävissä on parillista, rekisteröityä testiaineistoa, kuvasynteesin tuloksia voidaan arvioida kvantitatiivisesti. Tavallisimmin käytettyjä tunnuslukuja ovat pikselikohtainen keski(neliö)virhe MSE tai MAE, huippusignaali-kohina-suhde PSNR, sekä rakenteellista vastaavuutta kuvaava indeksi SSIM. Kaksi ensin mainittua perustuvat suoraan pikselikohtaisten absoluuttisten erojen vertailuun. SSIM puolestaan pyrkii ottamaan huomioon myös rakenteelliset riippuvuudet; se lasketaan  $N \times N$ -kokoisille kuvaikkunoille erikseen ja raportoidaan esimerkiksi koko kuvan keskiarvona. Rakenteellisen informaation säilymistä voidaan arvioida jopa ilman parillista testidataa – siis myös eri painotuksien välillä – keskinäisinformaation MI avulla. Se vertaa pikseliarvojen jakaumia kuvien välillä.

$$MAE/(MSE)(x, y) = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n |x_{ij} - y_{ij}|^{(2)} \quad (12)$$

$$PSNR(x, y) = 10 \log_{10} \left( \frac{[\max(\max(x), \max(y))]^2}{MSE} \right) \quad (13)$$

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (14)$$

$$MI(x, y) = \sum_{m \in x} \sum_{n \in y} p(m, n) \log \frac{p(m, n)}{p(m)p(n)} \quad (15)$$

SSIM:n kaavassa parametrit  $c_1$  ja  $c_2$  liittyvät laskennan stabilointiin nollakeskiarvojen tai -varianssien tapauksessa.

Erityisesti sovelluksissa, joissa kuvasynteesin motivaationa on automaattisen segmentoinnin kehittäminen, on hyödyllistä tarkastella segmentoinnin onnistumista synteettisen kuvan pohjalta. Isola ym. (2017) käyttävät tähän ajatukseen pohjautuvasta tunnusluvusta nimeä *FCN-score*, joka viittaa yksinkertaisesti segmentointiin käytettävään täysin konvolutiiviseen neuroverkkoon (*Fully Convolutional Network*). Metriikan taustalla on oletus siitä, että aidon kuva-aineiston avulla koulutettu neuroverkko pystyisi segmentoimaan samalla tavalla myös vastaavat synteettiset kuvat.

Segmentoinnin tarkkuutta voidaan mitata pikseli- tai luokkakohtaisen onnistumisprosentin avulla tai esimerkiksi Sørensen–Dice-indeksillä:

$$Dice = \frac{2|X \cap Y|}{|X| + |Y|} \quad (16)$$

#### 4.2.2. Kvalitatiivinen arviointi

Osassa tutkimuksia syntetisoinnin tuloksia on tarkasteltu myös kvalitatiivisesti asiantuntija-arvioinnin kautta, mikä on perusteltua siksi, että numeerisin tunnusluvuin hyväksi arvioidut kuvat eivät välttämättä näytä lainkaan realistisilta. Esimerkiksi pikselikohtaisten poikkeamien minimointi johtaa usein selvästi sumentuneeseen lopputulokseen, kuten jo neuroverkkojen arvofunktion määrittelyn yhteydessä todettiin.

Tutkimuksissa aistinvarainen arviointi oli toteutettu niin, että asiantuntijoille annettiin kuvapareja, joista heidän tuli erottaa, kumpi kuvista on aito ja kumpi synteettinen. Synteesin onnistumista kuvaa tällöin asiantuntijan erehtymistodennäköisyys. On kuitenkin kyseenalaista, saadaanko tällaisella asetelmalla arvioitua mitään kuvien todellisuudessa toivottua ominaisuutta: se, että synteettinen kuva näyttää aidolta ei takaa sitä, että se sisältäisi esimerkiksi oikeaa diagnostista informaatiota.

#### 4.2.3. Mallin luotettavuus

Tyypillisesti neuroverkon tuottamien ennusteiden tarkkuutta on vaikea arvioida. Verkon mahdollisia normalisointi- ja dropout-kerroksia voidaan kuitenkin testausvaiheessa hyödyntää mallin luotettavuuden arvioinnissa. Erilaisen dropout-maskien tuottamien ennusteiden varianssin perusteella voidaan estimoida ennusteen epävarmuutta. (Teye, Azizpour & Smith, 2018.)

### 4.3. Magneettikuvien T1↔T2 -synteesi: kirjallisuudessa esitetyjä sovelluksia

Taulukkoon 1 on koottu yhteenveto julkaisuista, joissa kilpailevien neuroverkkojen arkkitehtuuria on käytetty erityisesti T1↔T2-magneettikuvasynteesiin. Seuraavassa esitellään tarkemmin kunkin sovelluksen tarkempia yksityiskohtia ja saavutettuja tuloksia.

Yang, Li, Zhao, Fan, Chang ja Xu (2020) käyttivät parillista, osin rekisteröimätöntä opetusaineistoa, jonka rekisteröinti oli upotettu verkkoarkkitehtuuriin. Ensisijaisena GAN-arkkitehtuurina toimi pGAN pikselikohtaisella L1-häviöllä, jonka havaittiin pärjäävän tunnuslukuvertailussa aina puhdasta cGAN:ia paremmin. Vertailussa oli mukana myös CA-GAN (*Context-Aware GAN*, (Nie, Trullo, Lian, Petitjean, Ruan, Wang & Shen, 2017)). Tutkimuksen päätavoitteena oli tuottaa aineistoa automaattisen segmentoinnin ja rekisteröinnin kehittämistä varten. Segmentoinnin tarkkuutta saatiinkin parannettua ilman lisädatan tarvetta. Mallin toimivuutta testattiin useissa aineistoissa, joista osa koostui terveistä, ja osa patologioita sisältävistä kuvista. Useimmissa tapauksissa pGAN saavutti tavanomaisiin menetelmiin (esim. *Random Forest* (Jog, Carass & Prince, 2013)) verrattuna kvantitatiivisesti parempia tuloksia. Kuitenkin kasvaimia sisältävissä kuvissa jopa Random Forest ylsi parempiin tunnuslukuihin. Visuaalisesti arvioituna pGAN tuotti kuitenkin aina parempia tuloksia. Asiantuntija-arviointi teetettiin kolmella radiologilla ja viidellä lääketieteen opiskelijalla, jotka erehtyivät luulemaan pGAN:in tuottamaa kuvaa aidoksi 25 ja 27.6 prosentissa heille esitetyistä aito-syntetisoiduista kuvapareista.

Myös Dar ym. (2019) käyttivät osin parillista, osin paritonta aineistoa ja GAN-arkkitehtuurina toimi niin ikään pGAN. Parittoman aineistoon käsittelyyn hyödynnettiin CycleGAN:iä (josta artikkelissa käytetään epä johdonmukaisesti termiä cGAN). Käytetty parillinen data oli etukäteen rekisteröimätöntä; rekisteröinti oli sisällytetty prosessiin ja sitä oli kokeiltu painotusten välillä eri suuntiin. Kuten edellä, pikselikohtaisen häviön sisällyttäminen GAN-arvofunktion tuotti rekisteröidyn opetusaineiston tapauksessa parempia tuloksia kuin puhdas cGAN. CycleGAN tuotti visuaalisesti arvioituna hyviä tuloksia myös parittomalla opetusaineistolla. Tutkimuksessa arvioitiin myös 3D-MRI-aineiston kolmea peräkkäistä kerrosta



edustavien kuvien sarjan käytön etuja, koska peräkkäisten kerrosten ajateltiin tukevan toistensa rakenteellista informaatiota. Tämän havaittiinkin parantavan sekä kvantitatiivisia että kvalitatiivisia tuloksia, erityisesti pGAN:in tapauksessa. GAN-arkkitehtuurit päihittivät verrokkinsa (*Multimodal* (Chartsias, Joyce, Giuffrida & Tsiftaris, 2017) ja *Replica* (Jog, Carass, Roy, Pham & Prince, 2017)) myös patologioita sisältävien kuvien tapauksessa.

Welander, Karlsson ja Eklund (2018) vertailivat tutkimuksessaan kahta parittomaan kuvasynteesiin kykenevää verkkoarkkitehtuuria CycleGAN:iä ja UNIT:ia. Käytetty aineisto oli kuitenkin täysin parillista ja rekisteröityä, ja lisäksi koostui vain yhden kerroksen kuvista, jotka edelleen oli rekisteröity yhteiseen sapluunaan. Tarkasteltavien menetelmien verrokkeina toimivat CycleGAN\_s ja Generator\_s, joihin oli lisätty pikselikohtainen MAE-häviö. CycleGAN\_s vastasi siis pGAN:iä ja Generator\_s yksinkertaista – ei kilpailevaa – konvoluutioverkkoa. Generator\_s pärjasi parhaiten kaikkien käytettyjen tunnuslukujen (MAE, PSNR, MI) perusteella, mutta sen tuottamat kuvat olivat hyvin sumeita, ja aistinvaraisesti arvioituna selkeästi huonoja. CycleGAN, CycleGAN\_s ja UNIT olivat tunnuslukuvertailussa melko tasaväkisiä. Yllättävää oli se, että CycleGAN\_s:n lisätty MAE-häviö ei tuonut T1-kuvien synteesiin lainkaan kvantitatiivista lisäarvoa; puhtaan CycleGAN:in tunnusluvut olivat jopa hieman paremmat. Ensimmäisen kirjoittajan itsensä suorittamassa aistinvaraisessa arvioinnissa CycleGAN:in ja UNIT:in tuotokset osoittautuivat parhaiksi. Mielenkiintoisena havaintona tutkimuksessa kävi ilmi, että vaikka syntetisoidut T2-kuvat olivat kvantitatiivisesti arvioiden huonompia kuin T1-kuvat, niitä oli aistinvaraisesti vaikeampi erottaa aidoista. Tämä johtuu luultavasti yksinkertaisesti siitä, että T2-kuvien keskimääräinen intensiteetti on luonnollisesti matalampi, eli ne ovat tummempia. Lisäksi huomattiin, että kuvissa olevilla selkeillä rajoilla, joissa intensiteetin muutos on suuri, synteettisissä kuvissa on suurimpia virheitä.

Liu:n ym. (2018) sovelluksessa tarkoituksena oli yhdistää painotussynteesi ja segmentointi. GAN-arkkitehtuurina käytettiin CycleGAN:iä, johon oli lisätty pikselikohtainen segmentointihäviö. Toisin kuin kaikissa muissa tässä työssä käsitellyissä julkaisuissa, aineistona oli polven magneettikuvia. Tutkimuksesta tekee kuitenkin mielenkiintoisen se, että se on ainoa, jossa käytetty aineisto oli aidosti paritonta: opetukseen käytetyt eri painotuksia edustavat kuvat olivat peräisin täysin eri

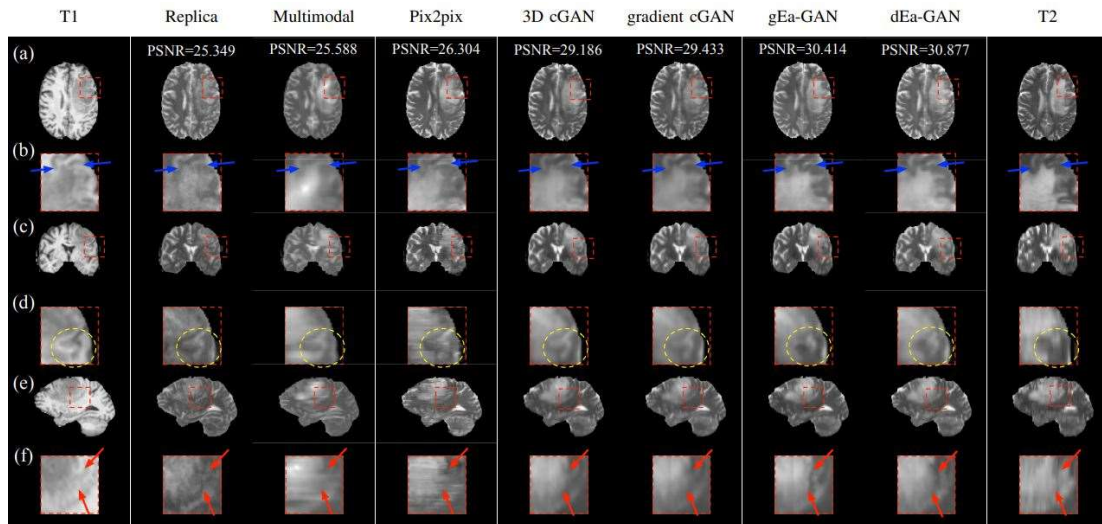
lähteistä. Sen sijaan lisäinformaationa käytettiin manuaalisesti luotuja kudossegmentaatiomaskeja. Parillista aineistoa ei käytetty edes mallin testaamiseen, vaan tuloksia arvioitiin synteettisten kuvien segmentointitarkkuuden perusteella ja visuaalisesti. Synteettisten kuvien keskimääräinen pikselikohtainen segmentointitarkkuus oli 73 % ( $T1 \rightarrow PD$ ) ja 79 % ( $T1 \rightarrow T2$ ), siinä missä aidoille kuville vastaavat arvot olivat 78 ja 83 %. Huomionarvoista on, että sekä segmentointihäviön painoarvon kasvattaminen että sen pienentäminen arvofunktiossa näyttivät heikentävän tuloksia. Tämä ilmentää arvofunktion kiinteiden parametrien säätämisen vaikeutta.

(Zhaoa ym., 2020) on tässä esitellyistä sovelluksista ainoa, jossa hyödynnettiin niin kutsuttua Bayesilaista neuroverkkoa. Käytännössä tämä tarkoittaa sitä, että verkkorakenteeseen oli sisällytetty dropout-kerroksia, jotka mahdollistavat testausvaiheessa myös mallin luotettavuuden arvioinnin. GAN-arkkitehtuurissa oli käytetty pikselikohtaisen L1-normin lisäksi KL-divergenssihäviötä. Tutkimuksen päätavoitteena oli arvioida kahden erilaisen dropout-menetelmän (*concrete dropout ja Monte Carlo -dropout*) hyödyllisyyttä mallin luotettavuuden arvioinnissa. Näiden menetelmien toimintaan ei ole tarkoituksenmukaista tässä työssä syventyä, mutta niihin liittyvä huomionarvoinen tulos tutkimuksessa oli se, että kummallakin menetelmällä saatiin kohtuullisen hyvin tunnistettua kuvista kohdat, joissa synteetissä tapahtui virheitä. Toisin sanottuna kohdissa, joissa oli virheitä, myös ennusteelle estimoitu keskihajonta oli korkea. Virheen *suuruuden* kanssa keskihajonta ei kuitenkaan yksittäisten kuvien tasolla suoraan korreloinut.

(Lei, Fu, Mao, Curran, Liu & Yang, 2020) esittelemä GAN-generaattori hyödynsi verkkoon sisällytettyjä residuaaliosioita (engl. *residual blocks*) siten, että verkon oli mahdollista oppia useita eri kuvauksia samanaikaisesti: siis yhdestä painotuksesta moneksi. Residuaaliosioilla tarkoitetaan sitä, että verkkoon on rakennettu sellaisia yhteyksiä, jotka ohittavat osan kerroksista. Verkon generaattori otti syötteen sekä muokattavan kuvan että halutun synteettisen modaliteetin tunnuksen. Testiesimerkkinä tutkimuksessa suoritettiin transformaatio T2-painotuksesta kolmeen muuhun, T1-, T1c- ja T2-Flair-painotuksiin. Viiden ryhmän ristiinvalidioidinnin keskimääräisenä tuloksena raportoitiin normalisoiduksi keskivirheeksi (NMAE)  $0.018 \pm 0.003$   $0.014 \pm 0.002$   $0.022 \pm 0.005$  (T1, T1c, T2-Flair) ja huippusignaali-kohinasuhteeksi  $30.1 \pm 3.7$   $36.3 \pm 3.5$

30.4±4.7 dB (T1, T1c, T2-Flair). Visuaalisesti tulosten väitettiin olevan hyviä ja artikkelissa esitetyt kuvat vaikuttavatkin teräviltä. Osassa aito-synteettinen-kuvapareista kuitenkin erottuu selvä sävyero, mikä saattaa kieliä keskiarvoistavien numeeristen tunnuslukujen ovat epäluotettavuudesta.

Yu:n ym. (2019) esittämästä *Edge-Aware GAN*:istä erikoislaatuinen teki se, että GAN-arvofunktioon oli sisällytetty 3D-Sobel-reunantunnistusfiltterin avulla laskettu rakennehäviö. Toisin kuin muut tässä esitetyt sovellukset, verkko oli myös suunniteltu käsittelemään suoraan kolmeulotteista MRI-ainestoa. Laskennallisista rajoitteista johtuen verkkoa ei kuitenkaan opetettu kokonaisilla 3D-volyymeilla vaan pienemmillä, osin päällekkäisillä  $128^3$  pikselin tilkuilla. Tutkimuksessa ehdotettiin kahta erilaista uutta verkostoa: ensimmäisessä, *gEa-GAN*:issä, rakennehäviö oli sisällytetty vain generaattorin arvofunktioon ja toisessa, *dEa-GAN*:issä, sekin opetettiin kilpailun kautta. Tutkimuksessa testattiin T2- ja T2-Flair-kuvien synteesiä T1-kuvien pohjalta. Testissä käytettiin sekä kokonaisia volyymejä, että kasvainalueisiin rajattuja osia. *dEa-GAN* osoittautui tunnuslukuvertailussa (PSNR, NMSE, SSIM) poikkeuksetta kaikkia verrokkejaan (Replica, Multimodal, Pix2pix, 3D cGAN, gradient cGAN) paremmaksi. Näistä 3D cGAN edustaa Pix2pix:in 3D-versiota ja gradient cGAN:issa käytettiin Sobel-filtterin sijaan gradienttiperustaista rakennehäviötä. Myös *gEa-GAN* jäi *dEa-GAN*:ille lähes poikkeuksetta kakkoseksi; vain yhdessä tapauksessa ero SSIM:issä ei ollut tilastollisesti merkitsevä. Ehdotettujen uusien mallien ulkopuolelta Pix2pix:in ja 3D cGAN:in välinen vertailu osoitti myös, että 3D-kontekstuaalinen informaatio parantaa synteesituloksia merkittävästi. Synteettisten kuvien normalisoiduksi keskineliövirheeksi raportoitiin  $0.088\pm0.223$  ja  $0.105\pm0.174$  (T2 ja T2-Flair) ja SSIM:iksi  $0.967\pm0.016$  ja  $0.963\pm0.016$ . Tutkimuksessa ei tehty asiantuntija-arviota, mutta esitetyt kuvat ovat vakuuttavia.



Kuva 17: Eri menetelmin saavutettuja tuloksia T1→T2 synteesissä (Yu ym., 2019).

(Huang, Y., Zheng, Cong, Huang, W., Scott & Shao, 2020) esittelemä verkoston hyödynsi niin ikään kolmeulotteista kuvantamisaineistoa. Lisäksi arkkitehtuuri oli suunniteltu toimimaan parittomalla opetusaineistolla. Eritelty MCMT (*Multi-Task Coherent Modality Transferable*)-GAN-arkkitehtuuri pohjautui CycleGAN:iin, mutta sen arvofunktioon oli lisätty rakenteellisia ominaisuuksia ja segmentointitarkkuutta edustavat häviötermit. Koska verkon tarkoitus oli hyödyntää paritonta aineistoa, näiden häviöiden muotoilu ei ole yksiselitteistä. Segmentointihäviö oli sisällytetty malliin kilpailullisesti diskriminaattoreiden kautta: arkkitehtuurissa kummankin CycleGAN-diskriminaattorin parina oli vastaava FCN-segmentoija. Lisäksi malliin oli sisällytetty jakaumapohjaisia regularisointitermejä, joiden yksityiskohdat ylittävät tämän tutkielman aihealueen. Mallin toimintaa arvioitiin kahdessa transformaatiossa:  $T2 \leftrightarrow PD$  ja  $T1 \leftrightarrow T2$ . Verrokkeina toimivat *MIMECS* (Roy, Carass & Prince, 2013), puhdas GAN, *DOTE* (Huang, Y., Shao & Frangi, 2017), CycleGAN ja *MSGAN* (Mao, Q., Lee, Tseng, Ma & Yang, 2019). Lisäksi esitettyjen jakaumapohjaisten häviötermien tuottamaa lisäarvoa arvioitiin ottamalla vertailuun mukaan myös MCMT-GAN:in riisuttuja versioita, joista osa regularisointitermeistä oli jätetty pois. Kahden ryhmän ristiinvalidoinnissa esitetty MCMT-GAN päihitti kaikki verrokkinsa kvantitatiivisten tunnuslukujen (PSNR, SSIM sekä segmentointitarkkuutta kuvaava Dice) osalta. Erot olivat tilastollisesti merkitseviä lähes kaikissa tapauksissa, lukuun ottamatta toisen, merkittävästi pienemmän, aineiston  $T2 \rightarrow T1$ -transformaatioita, jossa *DOTE* ja CycleGAN ylsivät lähes vastaaviin tuloksiin. Jakaumapohjaiset häviötermit paransivat

synteesituloksia jonkin verran, mutta tärkeämpi rooli oli CycleGAN- ja segmentointihäviöillä.  $T1 \rightarrow T2$  synteesille raportoidut tulokset olivat 33.53 dB, 0.9575 ja 78.83 % (PSNR, SSIM ja Dice) ja  $T2 \rightarrow T1$  synteesille 30.96 dB, 0.9222 ja 74.91 %.

Kaikissa tässä työssä tarkastelluissa julkaisuissa, joissa verkon opetusmenetelmät oli raportoitu riittävällä tarkkuudella, käytettiin Adam-optimoijaa. Myös sen parametrivalinnat vaikuttivat olevan hyvin vakiintuneet: oppimisnopeus oli 0.0002 ja beta-parametrit 0.5 ja 0.999. Tästä poikkeuksena, Dar ym. (2019) ja Yu ym. (2019) käyttivät 100 epookin jälkeen lineaarisesti hidastuvaa oppimisnopeutta.

Julkaisu	Transformaatiot	GAN-arkkitehtuuri	Generaattori-arkkitehtuuri	Diskriminaattori-arkkitehtuuri	Optimointi	Data
(Yang ym., 2020)	T1 $\leftrightarrow$ T2 T1 $\rightarrow$ T2-Flair T2 $\rightarrow$ T2-Flair T2 $\leftrightarrow$ PD	pGAN: $\lambda_{L1} = 100$ Vertailussa myös puhdas cGAN ja CA-GAN	U-net: 8 konvoluutiokerrosta: eränormalisointi ja Leaky ReLU (k = 0.2) 8 dekonvoluutiokerrosta: eränormalisointi, ReLU, viimeisessä tanH	70X70 PatchGAN: 4 konvoluutiokerrosta: eränormalisaatio, ReLU	Adam: $\beta_1 = 0.5$ , $\beta_2 = 0.999$ , oppimisnopeus = 0.0002, eräkokko = 1	Parillista, osin rekisteröimätöntä (upotettu rekisteröinti), terveitä ja patologioita sisältäviä kuvia  BraTs2015, Iseg2017, MRBrain13, ADNI, RIRE
(Dar ym., 2019)	T1 $\leftrightarrow$ T2	pGAN  cGAN = CycleGAN	U-net:  Adoptoitu: (Ronneberg ym., 2015), ei raportoitu tarkemmin	Adoptoitu: (Johnson, Alahi & Fei-Fei, 2016), ei raportoitu tarkemmin	Adam: $\beta_1 = 0.5$ , $\beta_2 = 0.999$ , oppimisnopeus = 0.0002 100 epookkia, sitten lineaarisesti nolnaan, eräkokko = 1	Parillista, osin rekisteröimätöntä, terveitä ja patologioita sisältäviä kuvia  pGAN:ille rekisteröitynä, CycleGAN:ille ilman  MIDAS, IXI, BraTs2015
(Welander ym., 2018)	T1 $\leftrightarrow$ T2	CycleGAN  UNIT	3 konvoluutiokerrosta: eränormalisointi, ReLU  8 residuaalikerrosta: konvoluutio $\rightarrow$ eränormalisointi $\rightarrow$ ReLU $\rightarrow$ konvoluutio $\rightarrow$ yhdistäminen  3 dekonvoluutiokerrosta: eränormalisointi (paitsi viimeisessä), ReLU, (viimeisessä tanH)	PatchGAN:  5 konvoluutiokerrosta: eränormalisointi (paitsi ensimmäisessä), Leaky ReLU (k = 0.2)	Adam: $\beta_1 = 0.5$ , $\beta_2 = 0.999$ , oppimisnopeus = 0.0002, eräkokko = 1	Parillista, (mutta käytetty kuin paritonta), rekisteröityä, vain yhdeltä korkeudelta, rekisteröity sapluunaan  Human Connectome Project

<b>(Liu, 2018)</b>	T1 $\leftrightarrow$ T2  (T1 $\leftrightarrow$ PD)	CycleGAN  + segmentointihäviö:  $\lambda_{cyc} = 10$  $\lambda_{seg} = 5$	Modifioitu U-net:  Tuottaa kaksi kuvaa: transformaation ja segmentoinnin.  10 konvoluutiokerrosta: eränormalisointi ja Leaky ReLU (k = 0.2)  10 dekonvoluutiokerrosta: eränormalisointi ja ReLU, segmentaatiopuolella viimeisenä softmax	PatchGAN:  5 konvoluutiokerrosta: eränormalisointi (paitsi kahdessa viimeisessä), Leaky ReLU (k = 0.2) (paitsi ensimmäisessä)	Adam:  oppimisnopeus 0.0002, eräkokoko = 3, $\beta$ - parametreja ei raportoitu	Polven kuvia, täysin paritonta, manuaalisesti tehdyt segmentointimaskit  The Segmentation of Knee Images 2010
<b>(Zhaoa ym., 2020)</b>	T1 $\rightarrow$ T2	pGAN  + KL-divergenssi:  $\lambda_{L1} = 100$  $\lambda_{KL} = 100$	Modifioitu U-net:  Dropout-kerrokset 2. 3. ja 4. dekonvoluutiokerroksen jälkeen.	5 konvoluutiokerrosta, ei raportoitu tarkemmin	Adam:  $\beta_1 = 0.5$ , $\beta_2 = 0.999$ , oppimisnopeus = 0.0002, eräkokoko = 16	Parillista, rekisteröityä, patologioita sisältäviä kuvia, rekisteröity sapluunaan  BraTs2018
<b>(Lei ym., 2020)</b>	T1 $\rightarrow$ T2  T1 $\rightarrow$ Flair  (PD $\rightarrow$ T2)	Multimodaliteetti-GAN, häviöitä ei raportoitu tarkemmin	Modifioitu U-net:  Residuaaliosioita, jotka mahdollistavat useat tavoitemodaliteetit	3 konvoluutio + pooling - kerrosta, lainattu: (Lei, Harms, Wang, Liu, Y., Shu, Jani, Curran, Mao, Liu, T., & Yang, 2019)  Tuottaa ennusteen synteettinen/aito ja lisäksi modaliteetin tunnuksen.	Ei raportoitu	Parillista, rekisteröityä, patologioita sisältäviä kuvia, rekisteröity sapluunaan  BraTs2015

(Yu ym., 2019)	T1→T2 T1→Flair (PD→T2)	gEa-GAN = generator-induced Egde-aware GAN  dEa-GAN = discriminator-induced Egde-aware GAN  3D-Sobel-reunahäviö + L1-normi  $\lambda_{reuna} = 0 \rightarrow 100$ ensimmäisen 20 epookin aikana, sitten vakio  $\lambda_{L1} = 300$	3D-U-net:  7 konvoluutiokerrosta: eränormalisointi, Leaky ReLU ( $k = 0.2$ )  7 dekonvoluutiokerrosta: eränormalisointi, ReLU	5 konvoluutiokerrosta: eränormalisointi, Leaky ReLU ( $k = 0.2$ )	Adam:  oppimisnopeus = 0.0002 100 epookkia, sitten lineaarisesti nolnaan, eräkokoko = 1, $\beta$ -parametreja ei raportoitu	Parillista, rekisteröityä, terveitä ja patologioita sisältäviä volyymeja  BraTs2015, IXI
(Huang, Y. ym., 2020)	T1→T2 (T2→PD)	MCMT-GAN = multi-task coherent modality transferable GAN = CycleGAN  + segmentointihäviö  + jakaumapohjaisia säätelytermejä	3 konvoluutiokerrosta: eränormalisointi, ReLU  6 residuaalikerrosta: 2 konvoluutiota  3 dekonvoluutiokerrosta: eränormalisointi, ReLU, viimeisessä tanH	70x70x70 PatchGAN:  eränormalisointi, Leaky ReLU, kerrosmäärää ei raportoitu	Adam:  Oppimisnopeus = 0.0002, eräkokoko = 1, $\beta$ -parametreja ei raportoitu	Parillista, rekisteröityä, mutta ristiinvalidoinnissa käytetty toisiaan vastaamattomia aineiston puolikkaita, terveitä volyymeja  IXI, NAMIC

Taulukko 1: GAN-sovelluksia magneettikuvien T1 ja T2 -painotusten syntetisoinnissa.



## 5. Yhteenveto

Edellä esitettyjen tulosten perusteella kilpailevien verkkojen arkkitehtuuria hyödyntämällä magneettikuvasynteessissä saadaan lähes poikkeuksetta parempia tuloksia kuin tavanomaisilla regressiomenetelmillä tai muilla syväoppimismenetelmillä. Mitä kilpaileviin verkkoihin itseensä tulee, esitetyistä tutkimuksista käy ilmi se, että ohjaamalla verkon oppimista erilaisilla arvofunktion lisätermillä saavutetaan niin kvantitatiivisesti kuin kvalitatiivisestikin parempia tuloksia kuin puhtaalla kilpailevalla verkolla. Esimerkiksi pikselikohtaisen häviön tapauksessa tämä on melko ennalta arvattavaa, mutta tuntuvia etuja saavutetaan myös korkeamman tason rakenteellisen informaation vaalimisella esimerkiksi segmentaatiomaskien, reunantunnistusfilterin, ohitusyhteyksien tai 3D-kontekstuaalisen informaation avulla.

Useimmissa tässä esitetyissä sovelluksissa rakennehäviö kuitenkin nojaa parillisen opetusaineiston käyttöön: esimerkiksi synteettisen kuvan Sobel-rakennetta verrattiin aidon kohdekuvan rakenteeseen (Yu ym., 2019). Kiinnostavaa olisikin tietää, voisiko vastaavanlaisen rakennehäviön upottaa esimerkiksi CycleGAN:in rekonstruktiohäviön sisään. Tällöin sen tuomasta lisäarvosta päättäisiin hyötymään myös parittoman aineiston tapauksessa vastaavalla tavalla kuin Huang, Y. ym. (2020) esittivät segmentointihäviön osalta.

Rakenteeseen liittyen tutkimuksissa kävi niin ikään ilmi, että opetusaineiston 3-ulotteisen kontekstuaalisen informaation hyödyntäminen, joko kokonaisten 3D-volyymien (Yu ym., 2019) tai jopa vain kolmen päällekkäisen kuvan sarjojen muodossa (Dar ym., 2019) parantaa synteesituloksia. 3D-aineiston käyttöä rajoittaa kuitenkin saatavuuden lisäksi myös laskentakapasiteetti. Toisaalta on huomattava, että kuten Yu ym. (2019) näyttivät, verkon opettamiseen ei tarvitse käyttää kokonaisia volyymejä, vaan ne voidaan osittaa pienemmiksi 3D-tilkuiksi.

Aineiston vaatimusten osalta kilpailevien verkkojen ja tässä erityisesti syklisiä rekonstruktiohäviötä hyödyntävän CycleGAN:in ehdoton vahvuus on se, että sen avulla kuvasynteessissä on periaatteessa mahdollista hyödyntää myös täysin paritonta

opetusaineistoa. Kuitenkin tässä esitetyissä tutkimuksissa, joissa CycleGAN:iä sovellettiin, käytetty aineisto oli tosiasiassa parillista ja rekisteröityä ja kuten todettu, Welandar ym. (2018) tapauksessa sen lisäksi koostui vain yhden aksiaalisen kerroksen kuvista, jotka oli edelleen rekisteröity yhteiseen sapluunaan. Siitä huolimatta, että synteesissä ei suoranaisesti hyödynnetty näitä ominaisuuksia, on oletettavaa, että näin pitkälle yhdenmukaistetun aineiston käsittelyyn rakennettu malli ei välttämättä toimi toivotulla tavalla heterogeenisemmän parittoman aineiston tapauksessa.

Yksi seikka, joka esitettyjen tulosten perusteella on selvää, on se, että kvantitatiivisilla tunnusluvuilla harvoin saadaan arvioitua synteettisten kuvien visuaalista laatua. Osin tästä syystä – ja osin puutteellisen normalisoinnin takia – myös eri tutkimuksissa saavutettujen tulosten vertaaminen suoraan toisiinsa on haasteellista. Kvantitatiivisten tunnuslukujen valinnassa korostuu myös käyttötarkoitusspesifisyys: mikäli synteesin tavoitteena on esimerkiksi automaattisen segmentoinnin kehittäminen, on paitsi verkon opetuksessa myös valmiin mallin arvioinnissa mahdollisesti painotettava eri asioita kuin silloin, jos tavoitteena on esimerkiksi tuottaa visuaalisesti mahdollisimman realistisia kuvia.

Mitä kuvien realismi tulee, on edelleen muistettava se, että mahdollisimman realistisen *näköisillä* kuvilla ei välttämättä ole mitään todellista arvoa, jos ne eivät välitä toivottua diagnostista informaatiota. Jotta synteettisten kuvien käyttöä kliinisessä diagnostiikassa voitaisiin harkita, niitä tulisikin asiantuntija-arvioida tarkemmin nimenomaan diagnostisen informaation säilymisen kannalta. Diagnostisen käytön kannalta olennaista ja lupaavaa on se, että Zhaoa ym. (2020) osoittivat Bayesilaisten neuroverkkojen ainakin jollain tasolla mahdollistavan synteettisten kuvien luotettavuuden arvioinnin.

Esitellyissä tutkimuksissa arvofunktion eri häviöiden painoja edustavien parametrien valinnasta ei juuri keskusteltu. Parametrien valinta on kuitenkin kiistatta haasteellista ja mitä useampia niitä arvofunktiossa on, sitä vaikeammaksi optimaalisen tasapainon löytäminen tulee. Kilpailevien verkkojen tapauksessa oman lisähaasteensa arvofunktion parametrien optimoinnin ohelle tuo generaattorin ja diskriminaattorin oppimisnopeuksien tasapainottaminen. Useimmiten sopivat parametriarvot löydetäänkin vain empiirisesti.

Sen, kuinka suuri arvo tämän työn keskiössä olevalla kahden perinteisen kuvakontrastin vaihdolla ylipäättään tulevaisuudessa on, ratkaisee magneettikuvaustekniikan ja kuvien tulkinnan perinteiden kehitys. Perinteisissä kuvakontrasteissa pikseliarvot eivät suoraan vastaa mitään kudosten fysikaalista ominaisuutta ja magneettikuvantamista onkin jo pitkään pyritty kehittämään entistä kvantitatiivisempaan suuntaan (Korvenoja, Ruuth & Kuusela, 2018). Esimerkiksi niin kutsuttua MDME (Multiple-Dynamic Multiple-Echo) -tekniikkaa käyttämällä kuvantamiskohteen magneettisista ominaisuuksista voidaan tuottaa yhdellä sekvenssillä sellaista kvantitatiivista informaatiota, jonka pohjalta voidaan suoraan syntetisoida useita perinteisiä painotuksia vastaavia kuvia. Näiden kuvien on laajassa arvioinnissa todettu olevan laadultaan täysin ja käytännön diagnostiselta arvoltaan lähes tavanomaisten kuvien veroisia. Lisäksi synteettisissä kuvissa esiintyi vähemmän häiriöitä kuin tavanomaisissa kuvissa. (Tanenbaum, Tsiouris, Johnson, Naidich, DeLano, Melhem, Quarterman, Parameswaran, Shankaranarayanan & Goyen, 2017.) Voisikin ajatella, että tulevaisuudessa koneoppimiseen pohjautuvan kuvasynteesin tavoitteet voisivat olla perinteisten kuvakontrastien imitoimisen sijaan esimerkiksi suoraan sen selvittämisessä, miten edellä kuvatusta uudentlaisesta aineistosta saataisiin irti mahdollisimman paljon informaatiota niin kliinistä diagnostiikkaa kuin automaattista kuvatulkingintaakin ajatellen.

## Lähdeluettelo

Andersson, J. & Ahlström, H. & Kullberg, J. (2019). A visual representation of 1 of the networks. [Kuva]. Julkaisussa: Separation of water and fat signal in whole-body gradient echo scans using convolutional neural networks. Teoksessa: Magnetic resonance in medicine. Vol. 82. Nro. 3. S. 1177–1186. DOI: 10.1002/mrm.27786. Haettu osoitteesta: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6618066/figure/mrm27786-fig-0001/>

Chartsias, A. & Joyce, T. & Giuffrida, M.V. & Tsaftaris, S.A. (2017). Multimodal MR synthesis via modality-invariant latent representation. Teoksessa: IEEE Transactions on Medical Imaging. Vol. 37. Nro. 3. S. 803–814. DOI: 10.1109/TMI.2017.2764326.

Chollet, F. (2018). Deep learning with Python. Manning Publications Company. Shelter Island, New York, USA. 351 s. ISBN 9781617294433.

Dar, S.U. & Yurt, M. & Karacan, L. & Erdem, A. & Erdem, E. & Cukur, T. (2019). Image Synthesis in Multi-Contrast MRI With Conditional Generative Adversarial Networks. Teoksessa: IEEE transactions on medical imaging. Vol. 38. Nro. 10. S. 2375–2388. DOI: 10.1109/TMI.2019.2901750.

Garcia, V. (2016). Discriminator – Patch GAN. [Kuva]. UPC Computer Vision Reading Group 25/11/2016 slides. Haettu osoitteesta: <https://image.slidesharecdn.com/07-image-to-image-translation-with-conditional-adversarial-networks-161125155412/95/image-to-image-translation-with-conditional-adversarial-nets-upc-reading-group-25-638.jpg?cb=1480089468>.

Given, C.A. & Stevens, B.S. & Lee, C. (2004). The MRI appearance of tumefactive demyelinating lesions. Teoksessa: American Journal of Roentgenology, Vol. 182. Nro. 1. S. 195–199.

Goodfellow, I. & Bengio, Y. & Courville, A. (2016). Deep Learning. MIT Press. Saatavissa: <http://www.deeplearningbook.org>.

Goodfellow, I. & Pouget-Abadie, J. & Mirza, M. & Xu, B. & Warde-Farley, D. & Ozair, S. & Courville, A. & Bengio, Y. (2014). Generative adversarial nets. Teoksessa: Ghahramani, Z. & Welling, M. & Cortes, C. & Lawrence, N. & Weinberger, K.Q. (toim.) Advances in neural information processing systems 27. 28th Annual Conference on Neural Information Processing Systems 2014, NIPS 2014. Montreal, Kanada. 8.-13.12.2014. S. 2672–2680.

Huang, H. & Yu, P.S. & Wang, C. (2018). An introduction to image synthesis with generative adversarial nets. arXiv preprint arXiv:1803.04469.

Huang, Y. & Shao, L. & Frangi, A.F. (2017). Dote: Dual convolutional filter learning for super-resolution and cross-modality synthesis in MRI. Teoksessa: Descoteaux, M. & Maier-Hein, L. & Franz, A. & Jannin, P. & Collins, L. & Duchesne, S. (toim.) Medical Image Computing and Computer Assisted Intervention – MICCAI 2017, 20th International Conference, Proceedings, Part III. Quebec City, Kanada. 11.-13.9.2017. Springer. DOI: 10.1007/978-3-319-66179-7\_11

Huang, Y. & Zheng, F. & Cong, R. & Huang, W. & Scott, M.R. & Shao, L. (2020). MCMT-GAN: Multi-Task Coherent Modality Transferable GAN for 3D Brain Image Synthesis. Teoksessa: IEEE Transactions on Image Processing. Vol. 29. S. 8187–8198. DOI: 10.1109/TIP.2020.3011557.

Ioffe, S. & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

Isola, P. & Zhu, J. & Zhou, T. & Efros, A.A. (2017). Image-to-image translation with conditional adversarial networks. Teoksessa: Proceedings of the IEEE conference on computer vision and pattern recognition. Honolulu, HI, USA. 21.-26.7.2017 S. 5967–5976. DOI: 10.1109/CVPR.2017.632. Kuva haettu osoitteesta: [https://phillipi.github.io/pix2pix/images/teaser\\_v3.png](https://phillipi.github.io/pix2pix/images/teaser_v3.png).

Jog, A. & Carass, A. & Roy, S. & Pham, D.L. & Prince, J.L. (2017). Random forest regression for magnetic resonance image synthesis. Teoksessa: Ayache, N. Duncan, J. (toim.) Medical image analysis. Vol. 35. S. 475–488. DOI: 10.1016/j.media.2016.08.009.

Jog, A. & Carass, A. & Prince, J.L. (2013). Magnetic resonance image synthesis through patch regression. Teoksessa: IEEE 10th International Symposium on Biomedical Imaging. San Francisco, CA, USA. 7.-11.4.2013. S. 350–353. DOI: 10.1109/ISBI.2013.6556484.

Johnson, J. & Alahi, A. & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. Teoksessa: Leibe, B. & Matas, J. & Sebe, N & Welling, M. (toim.) ECCV - European conference on computer vision 2016, 14th European Conference, Proceedings, Part II. Amsterdam, Alankomaat. 11-14.10.2016. Springer. S. 694–711. DOI: 10.1007/978-3-319-46475-6\_43

Karras, D.A. & Mertziotis, G.B. (2009). A gray-scale MRI image and its edges detected by applying a Sobel filter to the original image. [Kuva]. Julkaisussa: New PDE-based methods for image enhancement using SOM and Bayesian inference in various discretization schemes. Teoksessa: Measurement Science and Technology. Vol. 20. Nro. 10. 104012 (8). DOI:10.1088/0957-0233/20/10/104012. Haettu osoitteesta: [https://www.researchgate.net/profile/Dimitrios\\_Karras2/publication/228370366/figure/fig1/AS:301982729424901@1449009701648/A-gray-scale-MRI-image-and-its-edges-detected-by-applying-a-Sobel-filter-to-the-original.png](https://www.researchgate.net/profile/Dimitrios_Karras2/publication/228370366/figure/fig1/AS:301982729424901@1449009701648/A-gray-scale-MRI-image-and-its-edges-detected-by-applying-a-Sobel-filter-to-the-original.png)

Kingma, D.P. & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Korvenoja, A. & Ruuth, R., & Kuusela, L. (2018). Synteettiset kontrastit magneettikuvauksessa. Duodecim. Vol. 134. Nro. 6. S. 621–626. Saatavissa: <http://www.duodecimlehti.fi/api/pdf/duo14227>

Kullback, S. & Leibler, R.A. (1951). On information and sufficiency. Teoksessa: The annals of mathematical statistics. Vol. 22. Nro. 1. S. 79–86.

Lei, Y. & Fu, Y. & Mao, H. & Curran, W.J. & Liu, T. & Yang, X. (2020). Multi-modality MRI arbitrary transformation using unified generative adversarial networks.

Teoksessa: Išgum, I. & Landman, B.A. (toim.) Proceedings of SPIE 11313, Medical Imaging 2020: Image Processing. Osa 1131303. DOI: 10.1117/12.2549794.

Lei, Y. & Harms, J. & Wang, T. & Liu, Y. & Shu, H. & Jani, A.B. & Curran, W.J. & Mao, H. & Liu, T. & Yang, X. (2019). MRI-only based synthetic CT generation using dense cycle consistent generative adversarial networks. Teoksessa: Williamson, J.F. (toim.) Medical physics, Vol. 46. Nro. 8. S. 3565–3581. DOI: 10.1002/mp.13617.

Li, H. & Xu, Z. & Taylor, G. & Studer, C. & Goldstein, T. (2018). The loss surfaces of ResNet-56 with/without skip connections. [Kuva]. Julkaisussa: Visualizing the loss landscape of neural nets. Teoksessa: Advances in Neural Information Processing Systems 31, S. 6389–6399. Haettu osoitteesta: <https://github.com/tomgoldstein/loss-landscape/tree/master/doc/images>.

Lim, S. & Young, S.R. & Patton, R.M. (2016). Example images from the MNIST dataset. [Kuva]. Julkaisussa: An analysis of image storage systems for scalable training of deep neural networks. Teoksessa: The seventh workshop on Big Data Benchmarks, Performance Optimization, and Emerging Hardware (in conjunction with ASPLOS'16) Atlanta, Georgia, USA. 3.4.2016. Haettu osoitteesta: [https://www.researchgate.net/profile/Steven\\_Young11/publication/306056875/figure/fig1/AS:393921575309346@1470929630835/Example-images-from-the-MNIST-dataset.png](https://www.researchgate.net/profile/Steven_Young11/publication/306056875/figure/fig1/AS:393921575309346@1470929630835/Example-images-from-the-MNIST-dataset.png)

Liu, F. (2018). SUSAN: segment unannotated image structure using adversarial network. Teoksessa: Magnetic resonance in medicine. Vol. 81. Nro. 5. S. 3330–3345. DOI: 10.1002/mrm.27627

Liu, M. & Breuel, T. & Kautz, J. (2017). Unsupervised Image-to-Image Translation Networks. Teoksessa: Guyon, I. & Luxburg, U.V. & Bengio, S. & Wallach, H. & Fergus, R. & Vishwanathan, S. & Garnett, R. (toim.) Advances in Neural Information Processing Systems 30 (NIPS 2017). S. 700–708.

Mao, Q. & Lee, H. & Tseng, H. & Ma, S. & Yang, M. (2019). Mode seeking generative adversarial networks for diverse image synthesis. Teoksessa: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Long Beach, CA, USA. 5.-20.6.2019. S. 1429–1437. DOI: 10.1109/CVPR.2019.00152.

Mao, X. & Su, Z. & Tan, P.S. & Chow, J.K. & Wang, Y. (2019). Is Discriminator a Good Feature Extractor?. arXiv preprint arXiv:1912.00789.

Mirza, M. & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint. arXiv:1411.1784.

Nag, D. (2017). GAN:in toiminta päähkinänkuoressa. [Kuva]. Julkaisussa: Generative Adversarial Networks (GANs) in 50 lines of code (PyTorch). [Verkkosivu]. Viitattu 18.10.2020. Saatavissa: <https://medium.com/@devnag/generative-adversarial-networks-gans-in-50-lines-of-code-pytorch-e81b79659e3f>. Kuva haettu osoitteesta: [https://miro.medium.com/max/875/1\\*-gFsbyY9oJUQJ-A3GTfeg.png](https://miro.medium.com/max/875/1*-gFsbyY9oJUQJ-A3GTfeg.png)

Nagy, G. (1991). Neural networks-then and now. Teoksessa: IEEE Transactions on Neural Networks, Vol. 2. Nro. 2. S. 316–318. DOI: 10.1109/72.80343.

Nie, D. & Trullo, R. & Lian, J. & Petitjean, C. & Ruan, S. & Wang, Q. & Shen, D. (2017). Medical image synthesis with context-aware generative adversarial networks. Teoksessa: Descoteaux, M. & Maier-Hein, L. & Franz, A. & Jannin, P. & Collins, L. & Duchesne, S. (toim.) Medical Image Computing and Computer Assisted Intervention – MICCAI 2017, 20th International Conference, Proceedings, Part III. Quebec City, Kanada. 11.-13.9.2017. S.417–425. DOI: 10.1007/978-3-319-66179-7\_48.

Ronneberger, O. & Fischer, P. & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. Teoksessa: Navab, N. & Hornegger, J. & Wells, W.M. & Frangi, A.F. (toim.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, 18th International Conference, Proceedings, Part III. München, Saksa. 5.-9.10.2015. Springer. S. 234–241. DOI: 10.1007/978-3-319-24574-4\_28.

Rosenblatt, F. (1957). The perceptron, a perceiving and recognizing automation. Cornell Aeronautical Laboratory Report. Nro. 85-460-1.

Roy, S. & Carass, A. & Prince, J.L. (2013). Magnetic resonance image example-based contrast synthesis. IEEE Transactions on Medical Imaging. Vol. 32. Nro. 12. S. 2348–2363. DOI: 10.1109/TMI.2013.2282126.

Rumelhart, D.E. & Hinton, G.E. & Williams, R.J. (1986). Learning representations by back-propagating errors. Nature. Vol. 323. Nro. 6088. S. 533–536. DOI: 10.1038/323533a0

Srivastava, N. & Hinton, G. & Krizhevsky, A. & Sutskever, I. & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research. Vol. 15. Nro. 1. S. 1929–1958.

Tanenbaum, L.N. & Tsiouris, A.J. & Johnson, A.N. & Naidich, T.P. & DeLano, M.C. & Melhem, E.R. & Quarterman, P. & Parameswaran, S.X. & Shankaranarayanan, A. & Goyen, M. 2017. Synthetic MRI for clinical neuroimaging: results of the Magnetic Resonance Image Compilation (MAGiC) prospective, multicenter, multireader trial. American Journal of Neuroradiology. Vol. 38. Nro. 6. S. 1103–1110.

Teye, M. & Azizpour, H. & Smith, K. (2018). Bayesian uncertainty estimation for batch normalized deep networks. Teoksessa: Dy, J. Krause, A. (toim.) Proceedings of the 35th International Conference on Machine Learning, PMLR 80. S. 4907–4916.

van der Plas, A. (2015). MRI-technique. [Verkkosivu]. Viitattu 3.10.2020. Saatavissa: <https://www.startradiology.com/the-basics/mri-technique/index.html>

Welander, P. & Karlsson, S. & Eklund, A. (2018). Generative adversarial networks for image-to-image translation on multi-contrast MR images – A comparison of CycleGAN and UNIT. arXiv preprint arXiv:1806.07777.

Wikimedia-käyttäjä Cecbur. Convolutional Neural Network with Color Image Filter. [Kuva]. (2019). Haettu osoitteesta: [https://commons.wikimedia.org/wiki/Category:Convolutional\\_neural\\_networks#/media/File:Convolutional\\_Neural\\_Network\\_with\\_Color\\_Image\\_Filter.gif](https://commons.wikimedia.org/wiki/Category:Convolutional_neural_networks#/media/File:Convolutional_Neural_Network_with_Color_Image_Filter.gif).

Yang, Q. & Li, N. & Zhao, Z. & Fan, X. & Chang, E.I. & Xu, Y. (2020). MRI Cross-Modality Image-to-Image Translation. Teoksessa: Scientific reports (Nature Publisher Group). Vol. 10. Nro. 1. Artikkelin nro. 3753.

Yi, X. & Walia, E. & Babyn, P. (2019). Generative adversarial network in medical imaging: A review. Teoksessa: Ayache, N. & Duncan, J.S. (toim.) Medical image analysis. Vol. 58. Nro. 101552. DOI: 10.1016/j.media.2019.101552.

Yu, B. & Zhou, L. & Wang, L. & Shi, Y. & Fripp, J. & Bourgeat, P. (2019). Ea-GANs: Edge-Aware Generative Adversarial Networks for Cross-Modality MR Image Synthesis. Teoksessa: IEEE Transactions on Medical Imaging. Vol. 38. Nro. 7. S. 1750–1762. DOI: 10.1109/TMI.2019.2895894.

Zhao, G. & Meyerand, M.E. & Birn, R.M. (2020). Bayesian Conditional GAN for MRI Brain Image Synthesis. arXiv preprint arXiv:2005.11875.

Zhu, J. & Park, T. & Isola, P. & Efros, A.A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. Teoksessa: Proceedings of the IEEE international conference on computer vision 2017. Venetsia, Italia. 22.-29.10.2017. S. 2242–2251, DOI: 10.1109/ICCV.2017.244.