

**Tuomas Pitkänen**

**Syvät konvoluutioneuroverkot kuvien laadun ja resoluution  
parantamisessa**

Tietotekniikan kandidaatintutkielma

1. kesäkuuta 2019

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

**Tekijä:** Tuomas Pitkänen

**Yhteystiedot:** tuomas.t.pitkanen@student.jyu.fi

**Ohjaaja:** Sanna Mönkölä

**Työn nimi:** Syvät konvoluutioneuroverkot kuvien laadun ja resoluution parantamisessa

**Title in English:** Using deep neural networks to improve image quality and resolution

**Työ:** Kandidaatintutkielma

**Sivumäärä:** 26+0

**Tiivistelmä:** Tutkielma on kirjallisuuskatsaus SRCNN-menetelmästä (engl. *super-resolution convolutional neural network method*) ja sen eri versioista. Näillä menetelmillä voidaan skaalata kuvia suuremmiksi ilman, että lopputuloksesta tulee pikselöitynyt. Yksittäisten kuvien suurentamisen ja pienentämisen lisäksi niitä voidaan käyttää myös videoiden skaalamiseen sekä häviöllisestä pakkaamisesta syntyvien jäänteiden vähentämiseen.

**Avainsanat:** neuroverkot, syvät konvoluutioneuroverkot, resoluution suurentaminen, kuvanlaadun parantaminen

**Abstract:** This thesis is a literature review of a method called super-resolution convolutional neural network (SRCNN) and its different versions. These methods can be used to upscale images without causing pixelation. In addition to upscaling and downscaling single images, they can also be used to scale videos and to reduce artifacts caused by lossy compression.

**Keywords:** neural networks, deep convolutional neural networks, super-resolution, improving image quality

## **Kuviot**

Kuvio 1. Posterisointiefekti. ....	6
Kuvio 2. Rengastumisartefakteja.....	6
Kuvio 3. Kuvanskaalausmenetelmiä skaalauskerroimella 3 (Dong ym. 2014). ....	11

## Sisältö

1	JOHDANTO .....	1
2	KUVANLAATU JA -KÄSITTELY .....	3
	2.1 Tiedon häviäminen ja vääristymät .....	4
	2.2 Kuvan skaalaaminen .....	7
3	NEUROVERKOT .....	8
4	SRCNN JA SEN VARIAATIOIT .....	11
	4.1 SRCNN .....	11
	4.2 ARCNN .....	13
	4.3 FSRCNN .....	14
5	MENETELMIEN VERTAILUA JA SOVELLUSKOHTEITA .....	17
	5.1 Menetelmien vertailua .....	17
	5.2 Sovelluskohteita .....	18
6	YHTEENVETO .....	20
	LÄHTEET .....	21

# 1 Johdanto

Kamerat ovat yleistyneet päivittäin käytettävissä ja helposti mukana kuljetettavissa mobiililaitteissa, ja tämän myötä myös kuvaamisen sekä kuvien määrä ovat lisääntyneet. Otettuja kuvia ladataan ja jaetaan aktiivisesti sosiaalisessa mediassa sekä muissa verkkopalveluissa. Tallennustilaa ja verkon kaistanleveyttä on kuitenkin rajoitetusti, joten kuvia joudutaan usein pakkaamaan häviöllisesti. Lisäksi kameroiden tarkkuus on rajallinen erityisesti halvemmissa puhelimissa ja erilaisissa valvontakameroissa. Ongelmallisia ovat myös vanhat digitaaliset kamerakuvat sekä moneen kertaan häviöllisesti pakatut ja tallennetut tiedostot. Näistä syistä on monia kuvia, joiden resoluutio on pieni tai kuvanlaatu huono.

Kuvanlaatuongelmia pyritään ratkaisemaan monin tavoin. Suoraviivaisin näistä on tallennuskapasiteetin ja siirtonopeuksien nostaminen, mutta tästä aiheutuvat kustannukset estävät sen käytön ainoana ratkaisuna. Mainoksien näyttämällä ja käyttäjätietojen keräämisellä toimintansa rahoittavat yritykset eivät nimittäin pysty laajentamaan kapasiteettiaan loputtomiin pysyessään samalla ilmaisina loppukäyttäjilleen. Toisaalta monet maksavatkaan käyttäjät eivät ole valmiita maksamaan entistä korkeampaa hintaa muuten samantasoisesta palvelusta.

Teknologian kehitys mahdollistaa parempien pakkausmenetelmien kehittämisen, mutta kokonaan uusien tiedostomuotojen käyttöönotto on silti hidasta. Esimerkiksi reaali maailman valokuvien tallentamiseen käytetään yhä tavallisesti JPEG-tiedostomuotoa, joka standardoitiin jo vuonna 1990 (World Wide Web Consortium 2019).

Jos kuvanlaatu on heikko, kuvassa on luultavasti kohinaa. Sitä voidaan vähentää kohinanpoistosuodattimilla, mutta niillä on valitettavasti sivuvaikutuksia. Koska kohinaa esiintyy usein satunnaisesti, sen erottaminen halutusta tiedosta on vaikeaa. Tämän takia kohinan poistamisen seurauksena myös muista kohdista kuvaa katoaa tietoa. Jos kaivattua tietoa katoaa liikaa, kuvasta tulee sumea.

Kuvien pienen resoluution suurentamiseen taas käytetään erilaisia interpolaatiomenetelmiä, joita hyödynnetään muun muassa kuvankatselu- ja kuvankäsittelyohjelmissa sekä verkkoselaimissa. Menetelmät arvioivat kuvapisteen väriarvon sen lähellä olevien pikseleiden perusteella. Myös näiden tekniikoiden heikkoutena on kuvan sumeus.

Matalaa kuvanlaatua ja resoluutiota voidaan ajatella myös käänteisongelmana. Siinä kuvan ottaminen ja pakkaaminen nähdään joukkona peräkkäisiä funktioina, joissa syötetiedoista muodostuu vaiheittain lopullinen kuva. Tällöin tavoitteena on palauttaa lopputilanteesta alkutilanteen syötetiedot ja täten alkuperäinen, tarkka, kuva. Tehtävään on luotu useita menetelmiä, joista monet hyödyntävät neuroverkkoja. Niitä voidaan käyttää palauttamaan kuviin tarkkuutta ja laatua.

Menetelmien runsaslukuisuuden vuoksi tämä tutkielma rajoittuu pieneen joukkoon käänteisongelmiin perustuvia, Dongin tutkimusryhmän kehittämiä, menetelmiä. Tutkittavina ovat läheisesti toisiinsa liittyvät menetelmät super-resoluutio konvoluutioneuroverkoilla (SRCNN, engl. *super-resolution convolutional neural network*) (Dong ym. 2014), artefaktien vähennys konvoluutioneuroverkoilla (ARCNN, engl. *artifacts reduction convolutional neural networks*) (Dong ym. 2015) ja nopea super-resoluutio konvoluutioneuroverkoilla (FSRCNN, engl. *fast super-resolution convolutional neural networks*) (Dong, Loy ja Tang 2016). Näille kaikille on yhteistä se, että ne perustuvat SRCNN:ään. Lyhyesti kuvailtuna SRCNN-menetelmä ikään kuin piirtää kuvan tarkempana uudelleen. Yksittäisten kuvien lisäksi mahdollinen sovelluskohde ovat videot, sillä ne koostuvat peräkkäisistä kuvista.

Tutkielma toteutetaan kirjallisuuskatsauksena tarkastelemalla SRCNN-menetelmän ja sen eri versioiden toimintaa teknisestä näkökulmasta lähdemateriaalin ja taustalähteiden avulla. Tutkimuskysymykset ovat seuraavat: Miten SRCNN:n eri versiot toimivat? Mitkä ovat menetelmien väliset erot ja yhtäläisyydet? Mitä mahdollisia sovelluskohteita on? Lisäksi taustatietoina on lyhyt selvitys kuvatiedostoista ja neuroverkkojen toimintaperiaatteista.

Tutkielman rakenne on seuraava: luvussa 2 kerrotaan kuvankäsittelystä ja kuvatiedostoista yleensä, luvussa 3 käydään läpi perusasiat neuroverkoista ja erityisesti konvoluutioneuroverkoista. Luvussa 4 käsitellään menetelmät SRCNN, ARCNN ja FSRCNN, jonka jälkeen luvussa 5 vertaillaan menetelmiä ja pohditaan niiden sovelluskohteita. Lopuksi, luvussa 6, on yhteenveto.

## 2 Kuvanlaatu ja -käsittely

Kuvanskaalaamisen ymmärtämiseksi on oleellista tietää, miten kuva tyypillisesti koodataan ja pakataan tiedostoksi. Tämä prosessi käydään läpi JPEG-tiedostomuodon avulla, koska se on tyypillinen kuvatiedostomuoto reaali maailman kuville ja siten yleinen syöte kuvanskaalausmenetelmille.

Tämä kappale perustuu ISO/IEC-standardiin (1993, s. 14–17). JPEG-tiedostomuodon häviöllinen pakkausvaihe toimii seuraavasti: kuva jaetaan  $8 \times 8$  pikselin osiin, jotka sitten diskreettiä kosinimuunnosta (engl. *discrete cosine transform, DCT*) käyttäen muutetaan DCT-kertoimiksi (engl. *DCT coefficients*). Ensimmäinen näistä (DC) määrää lohkon värin lähtötason, loput 63 (AC) muutokset siihen nähden yksittäisille pikseleille. Seuraavaksi nämä kertoimet kvantisoidaan (engl. *quantize*) eli muutetaan häviöllisesti pienempään skaalaan arvoja. Lisäksi DC-termit muutetaan kvantisoinnissa muotoon, jossa käsiteltävä DC on likimääräinen erotus edeltävästä DC-arvosta. Tilan säästämiseksi kvantisoidut muuttujat vielä pakataan häviöttömästi. Kvantisointi ja häviötön pakkaaminen tehdään erikseen määriteltujen koodaustaulujen mukaisesti. Taulut määrittävät toisiaan vastaavat koodatut ja koodaamattomat arvot, joten niitä tarvitaan myös kuvien dekodeeraamiseen esittämistä varten.

JPEG FIF on alustariippumaton toteutus JPEG-tiedostomuodosta (Hamilton 1992). Siihen viitataan usein pelkällä lyhenteellä JPEG, sillä muut JPEG-tiedostomuodot eivät ole yleisesti käytössä (World Wide Web Consortium 2019; Poynton 2012, s. 491).

Tyypillinen kuvatiedosto koostuu metadatatista ja kuvan datasta (Hamilton 1992; World Wide Web Consortium 2003). Metadata kertoo itse tiedostosta. Se voi sisältää tietoa esimerkiksi kuvan resoluutiosta, väriavaruuden koosta ja kuvan pakkaamismenetelmistä. Kuvan data taas sisältää yksittäiset kuvapisteen esitettyinä jossakin väriavaruudessa.

RGB-väriavaruudessa kuva koodataan kolmen päävärin vahvuuksien yhdistelminä. Värit ovat punainen (R), vihreä (G) ja sininen (B). Käytetty tarkkuus on tyypillisesti 8 bittiä väriä kohden, yhteensä 24 bittiä. RGB on hyvin tunnettu väriavaruus, koska sitä hyödyntävät mm. tavalliset näytöt, kuvankäsittelyohjelmat ja PNG-kuvaformaatti. Värien koodaaminen tällä tavalla on yksinkertaista, mutta heikkoutena on verrattain huono pakkautuvuus: kirkkauden

muutokset vaikuttavat jokaiseen värikanavaan.

YCbCr on toisenlainen tapa esittää väriavaruus. ITU-R (2011) standardisoi värikanavat seuraavasti: kuva esitetään yhdellä luma-kanavalla (Y) ja kahdella krominanssikanavalla (Cb ja Cr). Luma on gammakorjattu arvo luminanssista (Y'), joka taas on valon intensiteetin mitta. Luma- tai luminanssikanava voidaan mieltää mustavalkoversioksi kuvasta, koska valon intensiteetti määrää kuvan piirteet. Poyntonin (2012, s. 123, 365–366) mukaan YCbCr-väriavaruuden krominanssi-kanavat esittävät kuvan värit kelta–siniakselilla (Cb) sekä vihreä–punakselilla (Cr).

Tämän väriavaruuden pakkautuvuus on parempi kuin RGB-väriavaruudessa, koska kirkkaiden vaihtelut rajoittuvat luma-kanavaan. Haittapuolena on se, että YCbCr pitää muuttaa takaisin RGB-muotoon ennen kuin se voidaan esittää näytöllä. JPEG FIF käyttää YCbCr-väriavaruutta ilman luminanssin gammakorjausta (Hamilton 1992).

Väriavaruuksien välillä voidaan tehdä muutoksia. Esimerkiksi JPEG FIF -tiedostomuodon 8-bittinen RGB saadaan muunnettua YCbCr-muotoon seuraavalla kaavalla (Hamilton 1992):

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B, \\ Cb &= -0.1687R - 0.3313G + 0.5B + 128, \\ Cr &= 0.5R - 0.4187G - 0.0813B + 128. \end{aligned} \tag{2.1}$$

Esimerkin muunnos on lineaarinen ja pyöristysvirheitä lukuun ottamatta häviötön. Koska häviöllisen pakkaamisen tarkoituksena on kuitenkin säästää tallennustilaa, käytettyjen värien määrää tyypillisesti vähennetään värisävyjen aliotannalla (engl. *chroma subsampling*) (Poynton 2012, s. 124–125). Siinä missä ihmissilmä on herkkä luminanssin muutoksille, se erottaa läheisiä värisävyjä huonosti (Timofte, De ja Gool 2013; Poynton 2012, s. 11). Tästä syystä värisävyjen kohtuullinen vähentäminen ei alenna aistittua kuvanlaatua merkittävästi.

## 2.1 Tiedon häviäminen ja vääristymät

Kuvista katoaa tietoa niitä fyysisesti ottaessa ja häviöllisesti pakatessa. Huomionarvoista on se, että kuvan uudelleentallentaminen toistaa häviöllisen pakkaamisprosessin. Häviöllisen



pakkaamisen ongelmana onkin koodausmenetelmien kyvyttömyys erottaa kohinaa muusta kuvasta, jolloin toistuva uudelleenpakkaaminen esimerkiksi sosiaalisessa mediassa alentaa kuvanlaatua jokaisella kerralla entisestään.

Kun kuvasta häviää liikaa tietoa, siihen muodostuu näkyviä vääristymiä. Niitä kutsutaan yleisesti pakkausartefakteiksi (engl. *compression artifact*). Seuraavaksi esitellään JPEG-kuville ominaisia pakkausartefaktityyppejä.

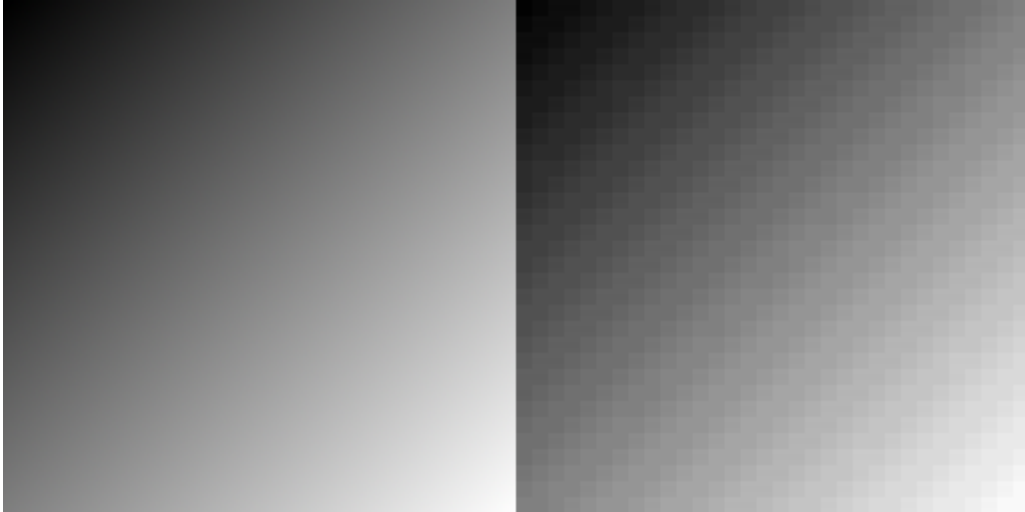
Kohinaksi (engl. *noise*) kutsutaan yleisesti ei-toivottuja, satunnaisia häiriöitä signaalissa. Sitä tulee kuvaan esimerkiksi kameran kennon rajallisen erottelukyvyn takia. Jos tiedetään kohinaton vertailukohta, signaalin oikean tiedon ja kohinan välinen suhde voidaan ilmoittaa SNR- tai PSNR-arvona (engl. [*peak*] *signal-to-noise ratio*). Se on mahdollinen tapa vertailla kuvanlaatuja samaa asiaa esittävien kuvien välillä.

Poyntonin (2012) mukaan palikoituminen (engl. *blocking*) ilmenee kuvassa  $8 \times 8$  pikselin lohkoina, joiden värit eroavat viereisistä lohkoista huomattavasti. Hän kertoo kvantisoinnin takia epätarkan DC-termin aiheuttavan palikoitumista. Koska DC-termistä lasketaan lohkon pikselien väriarvot (ISO/IEC JTC 1 1993), sen poikkeamat muuttavat koko lohkon kirkkautta yhtenäisesti. Kuvatiedoston matala kohdelaatu vahvistaa tätä ilmiötä pienentämällä käytettävissä olevaa väriskaalaa, jolloin vierekkäisten lohkojen värierot ovat suuremmat. Uudelleentallentaminen taas koodaa samat lohkot kuvasta uudelleen siten, että vääristymiä käsitellään muiden kuvan osien kanssa samanarvoisina. Palikoituminen näkyy selkeimmin tasavärisissä kohdissa kuvaa, missä mikään muu ei vie huomiota efektistä. Esimerkiksi tekstiä sisältävissä kuvissa tämä näkyy rakeisena taustana tekstin läheisyydessä.

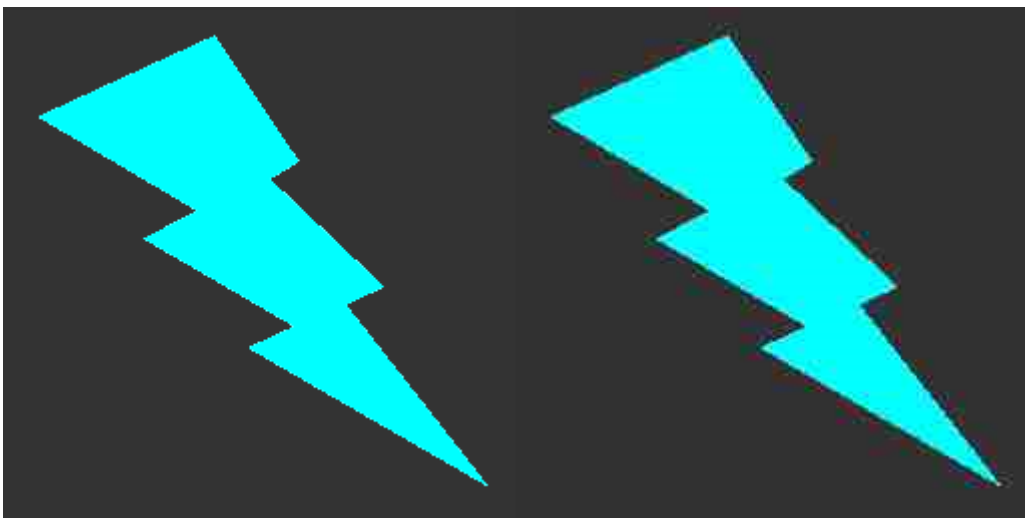
Rengastuminen (engl. *ringing effect*) on efekti, jossa terävien rajojen yhteyteen ilmestyy haloja (Popovici ja Withers 2007). Se johtuu kvantisoinnista aiheutuvasta tiedon katoamisesta: samaan lohkoon joutuu liikaa paljon toisistaan poikkeavia värejä.

Posterisointi (engl. *banding, posterization*) on epäjatkuvuutta tasaisissa värivaihdoksissa. Kvantisointi vähentää käytettävää väriskaalaa, joten pienet vivahteet värinsävyssä häviävät. Lopputuloksena kuvaan ilmestyy näkyviä yksivärisiä alueita paikoille, joissa värin kuuluisi muuttua vähitellen.

Kuvioissa 1 ja 2 demonstroidaan erilaisia artefaktityyppejä. Kuvaparien vasemmanpuoliset kuvat on tallennettu häviöttömästi PNG-muodossa, oikeanpuoliset taas JPEG-muodossa laatuasetuksella 20/100. Kuvat on tuotettu Paint.NET-kuvankäsittelyohjelmalla. Kuviossa 1 näkyy posterisointiefekti sekä palikoitumisartefakteja. Kuviossa 2 salaman ääri rajojen yhteydessä esiintyy rengastumis- ja palikoitumisartefakteja.



Kuvio 1. Posterisointiefekti.



Kuvio 2. Rengastumisartefakteja.

## 2.2 Kuvan skaalaaminen

Kuvien skaalaamiseen on kehitetty lukuisia menetelmiä. Eräs yksinkertaisimmista menetelmistä on lähimmän naapurin interpolaatio (engl. *nearest-neighbor interpolation*), jossa kuva nähdään joukkona jatkumattomia pisteitä. Uusi pikseli kopioidaan suoraan lähimmästä tunnetusta pikselistä, joten värien jatkuvuutta ei otetaan millään tavalla huomioon. Menetelmä on nopea, mutta liian yksinkertainen valokuvien jatkuvien värien interpolointiin.

Lineaarinen interpolaatio (engl. *bilinear interpolation*) tuottaa etäisyydellä painotetun keskiarvon lähimmistä pikseleistä. Menetelmässä sovitetaan ensimmäisen asteen polynomikäyriä

$$\tilde{g}(x) = cx + d \quad (2.2)$$

$2 \times 2$  pikseliruudukon väriarvoihin. Sovittaminen tehdään ensin vaakasuuntaisille arvoille, sijoitetaan pikselin suhteellista sijaintia vastaava  $x$ -koordinaatin arvo ja sovitetaan uusi käyrä saatuihin arvoihin. (Poynton 2012, s. 226–229) Tällainen interpolaatio tuottaa lähimmän naapurin menetelmää parempaa jälkeä, mutta ei sovellu valokuvien skaalaamiseen hyvin. Oletus, että pikselien väliset väriarvot ovat tasaisesti jatkuvat, on virheellinen erityisesti esineiden äärirajoilla.

Kuutiollinen interpolaatio (engl. *bicubic interpolation*) on laajennos lineaariseen interpolaatioon. Sen sijaan että kuvan värit oletettaisiin tasaisesti jatkuviksi, nyt otetaan huomioon myös värien muutosnopeus. Ensin sovitetaan  $4 \times 4$  ruudukon jokaiselle vaakariville kolmannen asteen polynomikäyrä

$$\tilde{g}(x) = ax^3 + bx^2 + cx + d. \quad (2.3)$$

Kuten lineaarisessa interpolaatiossa, seuraavaksi lasketaan arviot  $x$ -koordinaatin perusteella ja sovitetaan pystysuuntainen käyrä. Siitä lasketaan lopullinen arvo pikselin  $y$ -akselin sijainnin mukaan. (Poynton 2012, s. 226–229) Kuutiollinen interpolaatio toimii kuville kohtuullisesti, mutta sekin sumentaa niiden piirteitä.

Menetelmien etuliite *bi* viittaa interpolaation soveltamiseen sekä vaaka- että pystysuunnassa. Lineaarisen ja kuutiollisen menetelmän englanninkieliset nimet eivät ole täysin vakiintuneet, joten ne esiintyvät joskus ilman etuliitettä (*linear*, *cubic*). Toisaalta monista kuutiollisen menetelmän muunnelmistakin käytetään muotoa *bicubic interpolation*.

### 3 Neuroverkot

Tutkimuksessa tarkasteltavat menetelmät hyödyntävät neuroverkkoja. Neuroverkot ovat erityisen kiinnostavia kuvankäsittelytehtävissä siksi, että kuvien vaihtelevuus tuottaa ongelmia perinteisille menetelmille. Kuvassa voi olla esimerkiksi ihminen, maisema tai vaikkapa yksittäinen pinta. Kuvat voivat myös olla erikokoisia ja vaihtelevan laatuista. Näin laajan skaalien käsittelyyn on haastavaa luoda yleiskäyttöisiä ja tehokkaita algoritmeja pelkästään ihmisvoimin. Neuroverkot tuovat tähän ongelmaan ratkaisun niiden oppimiskyvyllä, sillä ne kykenevät oppimaan esimerkkien avulla.

Neuroverkot (engl. [*artificial*] *neural network*, ANN) ovat hieman aivojen toimintaa jäljitteleviä matemaattisia malleja (Alpaydin 2010, s. 233; Goodfellow, Bengio ja Courville 2016, s. 13). Ne koostuvat toisiinsa liittyneistä neuroneista, joista jokainen tuottaa saamastaan syötteestä jonkin välituloksen. Välitulos syntyy aktivointifunktiolla, joka muuntaa saamansa arvon epälineaarisesti uudeksi arvoksi (Goodfellow, Bengio ja Courville 2016, s. 171–172). Tämä lisää oppimiskykyä, koska monet mallinnettavat ilmiöt eivät ole lineaarisia.

Neuronien vastaanottamat syötteet ovat tyypillisesti vektori- tai matriisimuotoisia arvoja, joita kutsutaan parametreiksi. Niihin liittyvät läheisesti painot, jotka määrittelevät jokaisen parametrin painoarvon (Goodfellow, Bengio ja Courville 2016, s. 107–110).

Alpaydin (2010, s. 237–245) esittää neuroverkon perusosaksi perseptronin (engl. *perceptron*). Hänen mukaansa se laskee parametreittain painotetun operaation saamastaan syötteestä. Esimerkkinä hän mainitsee perseptronin, jonka laskema painotettu summa jakaa syöteavaruuden kahteen luokkaan. Jos summa ylittää tietyn rajan, syöte kuuluu luokkaan  $C_1$ , muussa tapauksessa se kuuluu luokkaan  $C_2$ .

Neuroverkoissa on kolme erilaista pääkerrostyyppiä: syötekerros, piilokerrokset ja ulostulokerros. Syötekerroksella otetaan vastaan kaikki ne parametrit, joista neuroverkko lopulta tuottaa tuloksen (Alpaydin 2010, s. 246). Parametrit lähtevät syötekerroksen neuroneista ensimmäisen piilokerroksen neuroneille, missä niistä lasketaan painotettu summa, joka taas jatkaa matkaansa joko seuraavaan piilokerrokseen tai ulostulokerrokseen (Alpaydin 2010, s. 246). Tämä toistuu jokaisessa piilokerroksen neuronissa. Viimeiseksi ulostulokerrokselta

saadaan lopputulos (Alpaydin 2010, s. 245–247; Goodfellow, Bengio ja Courville 2016, s. 168–169).

Ennen käyttöönottoa neuroverkko pitää kouluttaa, mikä vaatii koulutusaineistoa sekä keinon mitata tuotetun tuloksen hyvyyden. Kouluttamisen tavoitteena on optimoida neuroneissa käytettävien painojen suuruudet, jotka puolestaan vaikuttavat lopputulokseen. Goodfellow, Bengio ja Courville (2016, s. 85–86) määrittelevät käsitteen oppimisnopeus (engl. *learning rate*) siksi määräksi, jolla neuroverkon kouluttamisessa päivitetään painoja. Suuri arvo tuottaa nopeasti tuloksia, toisaalta pieni oppimisnopeus on tarkempi.

Koulutukseen käytettävä data jaetaan koulutus- ja testijoukkoihin (Goodfellow, Bengio ja Courville 2016, s. 110–116). Koulutusjoukkoa käytetään pelkästään kouluttamiseen ja testijoukkoa ainoastaan koulutetun neuroverkon testaamiseen. Näin pyritään välttämään ylioppimista, jolla tarkoitetaan sitä, että ilmiötä kuvaava malli myötäilee koulutusdataa liian yksityiskohtaisesti (Goodfellow, Bengio ja Courville 2016, s. 110–116).

Virhefunktio mittaa lasketun ja oikean arvon välistä eroa. Goodfellow’n, Bengion ja Courvillen (2016, s. 129–130) mukaan sitä voidaan käyttää erilaisten neuroverkkokonfiguraatioiden paremmuuden vertailemiseen sekä neuroverkkojen kouluttamiseen. Mitä pienemmän arvon virhefunktio antaa testidatalle, sitä tarkemman tuloksen neuroverkko tuottaa. Eräs käytetty virhefunktio on keskineliövirhe, jonka kaava on

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2, \quad (3.1)$$

missä  $n$  on aineiston koko,  $Y$  on laskettu arvo ja  $\hat{Y}$  on oikea arvo (Goodfellow, Bengio ja Courville 2016, s. 129–130). Syötedatajoukon MSE saadaan laskemalla keskiarvo jokaisen syöteen neliövirheestä, joka on arvioidun arvon neliöllinen poikkeama oikeasta arvosta. MSE on neliöllisyytensä takia positiivinen virheen mitta, joten sen pienin ja paras arvo on 0.

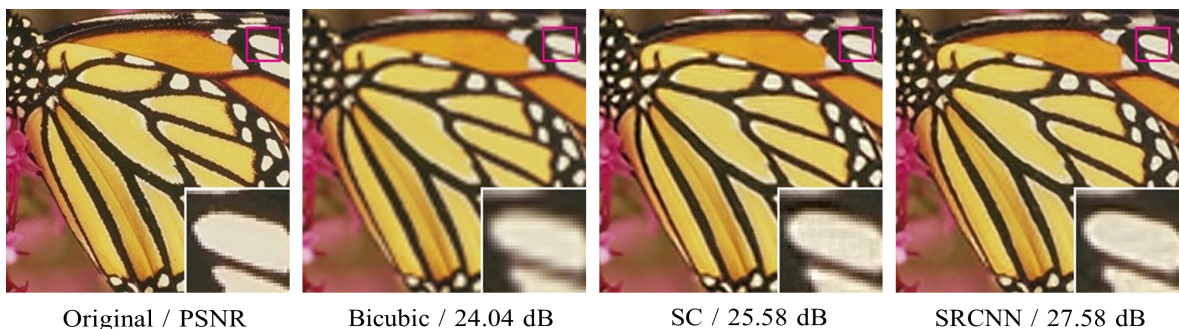
Yleinen kuviin liittyvissä tehtävissä käytettävä neuroverkkotyyppi on konvoluutioneuroverkko (engl. *convolutional neural network*, *CNN* tai *ConvNet*). Konvoluutiossa kahdesta signaalista muodostetaan kolmas signaali, joka taas voidaan yhdistää seuraavan signaalin kanssa. Konvoluutioneuroverkoissa vastaava vaikutus saadaan käyttämällä useaa peräkkäistä piilokerrosta. Goodfellow, Bengio ja Courville (2016) toteavat konvoluutioneuroverkkojen ole-

van tehokkaita toistuvien operaatioiden tekemiseen syötteen osajoukoille. Tämä sopii hyvin kuvankäsittelyyn, koska siinä halutaan usein käsitellä kuvan jokaista pikseliä siten, että vierekkäiset pikselit vaikuttavat lopputulokseen.

## 4 SRCNN ja sen variaatiot

Kuvan suurentaminen siten, että se pysyy todenmukaisena, on vaikeaa. Yang ym. (2008) määrittelevät yhden kuvan super-resoluution (engl. *single-image super-resolution*) seuraavasti: tehtävänä on palauttaa matalaresoluutioisesta kuvasta  $Y$  korkearesoluutioinen kuva  $X$ . Heidän mukaansa se on erittäin huonosti asetettu ongelma, koska jokaista syötettä  $Y$  kohden on ääretön määrä ratkaisuja  $X$ . Tässä luvussa käsiteltävät SRCNN-johdannaiset menetelmät on tarkoitettu ratkaisemaan yhden kuvan super-resoluutiotehtäviä. Luvussa käydään läpi menetelmien toimintaperiaatteet.

Kuviossa 3 on erilaisia kuvaskaalausmenetelmiä. Ne ovat vasemmalta oikealle: korkearesoluutioinen alkuperäiskuva (Original), kuvaa sumentava kuutiollinen interpolaatio (Bicubic), harvan koodauksen menetelmä (SC, engl. *sparse-coding-based method*) ja SRCNN. Menetelmän nimen vieressä on PSNR-arvo, jonka suuruus kertoo kuvanlaadusta. Kuvaa sumentava kuutiollinen interpolaatio tekee kuvasta vähemmän pikselöityneen näköisen kuvan terävyyden kustannuksella. Kuviossa 3 on huomionarvoista SRCNN-menetelmällä skaalatun kuvan ääri rajojen tarkkuus.



Kuvio 3. Kuvaskaalausmenetelmiä skaalauskerroimella 3 (Dong ym. 2014).

### 4.1 SRCNN

Super-resoluutio konvoluutioneuroverkoilla on ensimmäinen Dongin ym. (2014) julkaisema versio kuvaskaalaukseen tarkoitettu syvästä konvoluutioneuroverkosta. Vaikka se voidaan kouluttaa mielivaltaiselle skaalauskerroimelle, tässä tekstissä keskitytään tarkastelemaan ti-

lannetta, jossa menetelmää käytetään kuvan suurentamiseen.

SRCNN koostuu kolmesta pääkerroksesta sekä yhdestä lisäkerroksesta, jotka tekevät seuraavat operaatiot: kuutiollinen interpolaatio, tilkkujenkerääminen ja -esitys (engl. *patch extraction and representation*), epälineaarinen kartoittaminen (engl. *non-linear mapping*) sekä rekonstruointi (engl. *reconstruction*) (Dong ym. 2014).

Ensin kuva skaalataan kohderesoluutioon kuutiollisella interpolaatiolla (Dong ym. 2014). Menetelmä käsiteltiin osiossa 2.2. Koska se on yleinen operaatio lukuisilla valmiilla toteutuksilla, sitä ei kouluteta erikseen SRCNN-menetelmää varten (Dong ym. 2014). Samasta syystä operaatiota ei lasketa varsinaiseksi SRCNN:n osaksi, vaan se on lisäkerros.

Sitten skaalattu kuva jaetaan lomittaisiksi, tasasivuisiksi pikseliruudukoiksi (Dong ym. 2014), joita kutsutaan tilkuiksi (engl. *patch*). Ne esitetään moniulotteisina vektoreina, ja niiden ulottuvuuskokouksen lukumäärä on suoraan verrannollinen kuvan värikanavien lukumäärään (Dong ym. 2014). Vektorien koko ja ulottuvuuskokouksen lukumäärä vaikuttavat niihin kohdistuvien operaatioiden suoritusajaksiin, mutta toisaalta myös kuvanlaatuun.

Epälineaarisen kartoittamisen kerroksella tilkkuvektoreista tuotetaan uudet vektorit, joilla on mahdollisesti eri määrä ulottuvuuksia (Dong ym. 2014). Operaatio tuottaa uutta tietoa, mikä ilmenee uusien tilkkujen suuremmasta resoluutiosta ja mahdollisesta ulottuvuuskokouksen lukumäärän kasvusta.

Rekonstruointikerroksella muodostetaan lopullinen kuva edellisessä vaiheessa tuotetuista, korkearesoluutioisista tilkuista (Dong ym. 2014). Dongin ym. (2014) mukaan kerros toimii oletettavasti joukkoina suodattimia, jotka laskevat keskiarvon. Limittäisten tilkkujen käyttäminen auttaa ehkäisemään uusia palikoitumisartefakteja, sillä tilkkuja ei käsitellä täysin erillisinä kuten esimerkiksi JPEG-tiedostomuodon tapauksessa.

Dong ym. (2014) kertovat, että SRCNN perustuu Yangin ym. (2008) kehittämään harvan koodauksen menetelmään. Kuten SRCNN, se käsittelee syötekuvaansa tilkkuina, jotka muutetaan tarkemmiksi ja joista lopulta tuotetaan korkearesoluutioinen kuva laskemalla keskiarvo (Yang ym. 2008). Menetelmille on yhteistä myös kuutiollisen interpolaation käyttö resoluution kasvattamiseen. Tästä syystä Dong ym. (2014) toteavat, että harvan koodauksen



menetelmä voidaan toteuttaa syvänä konvoluutioneuroverkkona.

Dong ym. (2014) varmistavat SRCNN:n reilun vertailtavuuden kouluttamalla sen samalla tavalla kuin Timofte, De ja Gool (2013) ovat kouluttaneet oman menetelmänsä. Timofthen, Den ja Goolin (2013) mukaan koulutuskuvien käyttäminen suoraan ei tuota hyvin yleistyvää mallia, joten kuvat esikäsitellään parempaan muotoon. Käsittelyssä kuvasta tuotetaan kuutiollisella interpolaatiolla matalaresoluutioinen versio, jota verrataan alkuperäiseen kuvaan ja josta luodaan sen perusteella normalisoitu kuva. Lisäksi käsittelyyn kuuluu muita vaihteita, jotka siistivät kuvia.

SRCNN-menetelmän vahvuuksiin kuuluu yksinkertainen rakenne, sillä siinä on vain kolme pääkerrosta. Tämän ansiosta se on nopea ja helppo ymmärtää, ohjelmoida sekä kouluttaa. Dongin ym. (2014) julkaisemien tulosten perustella skaalauskerroin ei vaikuta menetelmän suoritusajkaan, joten suuretkin skaalaukset onnistuvat kelvollisessa ajassa. Haittapuolena tälle on se, että jokaista käytettävää kerrointa kohden neuroverkko pitää kouluttaa uudelleen.

Ongelmia tuottavat pakkausartefaktit ja muu kohina, sillä SRCNN ei osaa erotella niitä halutusta datasta, jolloin myös tuotettuun kuvaan päätyy artefakteja (Dong ym. 2015). Lisäksi verrattain hidas suoritusajka estää menetelmän hyödyntämisen sellaisenaan reaaliaikaisissa käyttökohteissa. Esimerkiksi  $512 \times 512$ -resoluutioisen kuvan skaalaaminen kestää noin puoli sekuntia tutkimukseen käytetyllä Nvidia GTX 770 -näytönohjaimella. Kuvanlaadun kannalta rajoittavana tekijänä on lähdekuvan tarkkuus, sillä täysin tyhjästä ei pystytä interpoloimaan informaatiota.

## 4.2 ARCNN

Kuvissa esiintyvät pakkausartefaktit aiheuttavat ongelmia skaalausmenetelmille. Tämän vuoksi Dong ym. (2015) ovat luoneet SRCNN:n pohjalta ARCNN-menetelmän, joka on kuvanskaalaamisen lisäksi erityisesti optimoitu poistamaan häviöllisistä pakkausmetodeista syntyviä artefakteja. Se on artefaktien sattumanvaraisuuden takia haastava tehtävä.

Dong ym. (2015) rajoittuvat tutkimuksessaan vain YCbCr-väriavaruuden luminanssiosaan. Kuten JPEG-pakkauksen värinsävyjen aliotannassa, ARCNN:ssä panostetaan tämän silmälle

helpoiten näkyvimmän värikanavan laatuun. Näin loput kanavat voidaan jättää huomiotta ja täten kouluttaa neuroverkko nopeammin.

ARCNN:n rakenne on muuten sama kuin SRCNN:n, mutta siihen on lisätty uusi kerros tilkkujenkeräämis- ja esityskerroksen sekä epälineaarisen kartoittamiskerroksen väliin. Dongin ym. (2015) mukaan tämän uuden kerroksen tehtävänä on vähentää kohinaa tilkuista kartoittamalla ne uusiksi piirrekartoiksi. Tähän on toiminut innoituksena Xiongin, Sunin ja Wun (2009) menetelmässään käyttämä piirteenvahvistusvaihe, joka suodattaa, interpoloi ja sitten vähentää esikäsittelystä aiheutunutta sumeutta kuvasta.

ARCNN:llä ja SRCNN:llä on pitkälti samat vahvuudet ja heikkoudet niiden rakenteiden samankaltaisuuden takia. Kummallakin pystytään skaalaamaan hyvälaatuisia kuvia kohtuullisessa suoritusajassa. Tärkein ero on ARCNN:n kyky sietää ja vähentää monenlaisia pakkausartefakteja. Se on hyvin hyödyllinen ominaisuus, sillä suuri osa internetissä esiintyvistä kuvista on häviöllisesti pakattuja.

SRCNN:ään nähden uusi ongelma on ARCNN:n vaikea koulutettavuus. Neuroverkon kerrosten lisääminen hidastaa kouluttamista kasvattamalla optimoitavien parametrien lukumäärää. Tämän seurauksena aikavaativuus kasvaa räjähdysmäisesti, sillä peräkkäisten kerrosten parametrien arvot vaikuttavat toisiinsa, jolloin mahdollisten yhdistelmien määrä on verrannollinen kerrosten parametrien tuloon (Goodfellow, Bengio ja Courville 2016, s. 155–157). Simonyan ja Zisserman (2014) ratkaisivat tämän ongelman kouluttamalla neuroverkkonsa vaiheittain, aloittamalla pienestä konfiguraatiosta. Dong ym. (2015) hyödyntävät samaa tekniikka siirtymällä asteittain haastavimpiin tehtäviin: ensin matala neuroverkko koulutetaan korkealaatuisemmilla kuvilla, sitten verkkoa syvennetään ja koulutetaan matalampilaatuisilla kuvilla. Tällöin neuroverkolla on vähemmän yhtä aikaa optimoitavia parametreja, joten ne konvergoituvat nopeammin.

### **4.3 FSRCNN**

Dong, Loy ja Tang (2016) ovat kehittäneet SRCNN:ään perustuen myös FSRCNN:n, tällä kertaa tavoitteena skaalata kuva aikaisempaa huomattavasti nopeammin. Vaikka se on nopeampi kuin SRCNN, sen tuottama kuvanlaatu on silti parempi (Dong, Loy ja Tang 2016).

FSRCNN:n rakenne on muutettu lähes kokonaan toiseksi kuin SRCNN:n. Uudessa rakenteessa on viisi osaa, jotka ovat suorittamisjärjestyksessä seuraavat: piirreirroitus (engl. *feature extraction*), kutistaminen (engl. *shrinking*), epälineaarinen kartoittaminen (engl. *non-linear mapping*), laajentaminen (engl. *expanding*) ja dekonvoluutio (engl. *deconvolution*) (Dong, Loy ja Tang 2016).

Piirreirroituskerros kerää tilkkuja kuten SRCNN:nkin vastaava, mutta koska kuvaa ei ole enää kuutiollisesti interpoloitu, tilkut voivat olla aiempaa pienempiä ilman kuvanlaadun heikkenemistä (Dong, Loy ja Tang 2016). Pienempiresoluutioisista tilkuista seuraa nopeusettu SRCNN:ään nähden, sillä tilkkujen määrä pysyy samassa kokoluokassa.

Kutistamiskerros vähentää tilkkujen ulottuvuuksia (Dong, Loy ja Tang 2016), mikä johtaa pienempään määrään neuroverkon parametreja (Alpaydin 2010, s. 267–270) ja täten nopeuttaa laskemista. Ulottuvuuksien vähentämisen vaikutus suoritusaikaan kertautuu, sillä käsiteltävän aineiston määrä vähenee usealta kerrokselta.

Epälinearisessa kartoitusvaiheessa tilkut muunnetaan epälineaarisesti toiseen muotoon, mikä on Dongin, Loyn ja Tangin (2016) mukaan tärkein lopputuloksen laadun ja suoritusnopeuden kannalta. Väitettä tukee se, että neuroverkon epälineaarisuuden kasvattaminen lisää oppimiskykyä. Toisin kuin SRCNN:n:ssä, FSRCNN:ssä käytetään useita pienempiä kerroksia tähän operaatioon (Dong, Loy ja Tang 2016). Näin saadaan parempi kuvanlaatu kuin yksikerroksisena, mutta syvempi neuroverkko on hitaampi ja vaikeampi kouluttaa.

Laajentamiskerroksen tehtävänä on palauttaa aiemmassa vaiheessa laskentatehokkuussyistä kutistettujen tilkkujen ulottuvuuksien lukumäärä takaisin (Dong, Loy ja Tang 2016). Tämä vaihe parantaa kuvanlaatua huomattavasti.

Dekonvoluutiokerros skaalaa kuvan ja yhdistää käsitellyt tilkut käyttämällä useita dekonvoluutiosuodattimia (engl. *deconvolution filter*) (Dong, Loy ja Tang 2016). Skaalauskerroin saadaan valittua asettamalla käytettyjen suodattimien askelkooksi (engl. *stride*) suoraan haluttu kerroin (Dong, Loy ja Tang 2016). Askelkoko määrää sen, kuinka pitkän matkan suodatin liikkuu laskuoperaatioiden välillä.

Koska FSRCNN:ssä on kerrosten lukumäärään liittyen suuri määrä muuttujia, Dong, Loy ja

Tang (2016) ovat jakaneet muuttajat herkkiin ja epäherkkiin muuttujiin verkon tarkastelemista ja kouluttamista varten. He määrittelevät muuttujan herkäksi, jos sen pienetkin muutokset voivat merkittävästi vaikuttaa koko verkon tuottamaan kuvanlaatuun. Epäherkille muuttujille asetetaan kohtuulliset oletusarvot, herkät muuttajat sen sijaan määritellään manuaalisesti (Dong, Loy ja Tang 2016). He käyttävät herkkiä muuttujia erilaisten neuroverkkokonfiguraatioiden vertailemiseen.

Vaikka FSRCNN on syvempi kuin SRCNN, siinä on paljon vähemmän parametreja (Dong, Loy ja Tang 2016). Goodfellow'n, Bengion ja Courvillen (2016, s. 197–201) mukaan syvemmät mallit pystyvät yleistymään paremmin, joten parametrien vähentäminen ei heikennä tuotetun kuvan laatua. Neuroverkon muuttujien määrää tämä sen sijaan ei vähennä, koska jokainen konvoluutiokerros vaatii sen kokoon ja muuhun toimintaan liittyviä asetuksia.

Edellä käsiteltyihin menetelmiin verrattuna FSRCNN:llä on lukuisia vahvuuksia, sillä se tuottaa niistä parhaan kuvanlaadun ja toimii nopeiten. Nopeus etu syntyy pienemmästä parametrimäärästä sekä siitä, että se ei tarvitse esikäsitellynä kuutiollista interpolaatiota. Kuutiollisen interpolaation puuttuminen tuo myös toisen edun: (Dong, Loy ja Tang 2016) kertovat verkon konfiguraation olevan siirrettävissä, joten uudet skaalauskerrokset vaativat pelkästään dekonvoluutiokerroksen uudelleen kouluttamisen. Tämän ansiosta lukuisten kerroin vaihtoehtojen tarjoaminen käytännön sovelluksissa on mahdollista.

On mahdollista, että kuutiollisen interpolaation poisjättäminen parantaa kuvanlaatua, mitä tukee sillä skaalattujen kuvien sumeus. Koska käsitellyt menetelmät käyttävät pelkästään edellisten kerrosten syötteitä, suuren mutta epätarkan kuvan tuottava esikäsitely saattaa heikentää lopputuloksen laatua. FSRCNN sen sijaan tekee skaalaamisen vasta viimeisellä kerroksella (Dong, Loy ja Tang 2016), joten interpoloitu syöte ei häiritse kuvanlaatua.

FSRCNN:n huonona puolena on sen rakenteen monimutkaisuus. Suurempi kerrosmäärä tekee menetelmän tarkastelusta ja muokkaamisesta vaikeampaa.

## 5 Menetelmien vertailua ja sovelluskohteita

Tässä luvussa vertaillaan menetelmiä keskenään sekä tutkitaan mahdollisia kehitysalueita ja sovelluskohteita. Tarkasteltavat menetelmät ovat rakenteiltaan ja päätehtäviltään samanlaisia, joten niitä on järkevää verrata toisiinsa. Niiden parhaita kerroksia yhdistelemällä ja uusia algoritmeja lisäämällä voi olla mahdollista tuottaa versioita, jotka toimivat aiempia paremmin.

### 5.1 Menetelmien vertailua

Menetelmät onnistuvat päätehtävässään, hyvälaatuisten kuvien tuottamisessa pienistä ja epätarkoista kuvista kohtuullisessa ajassa. Ne toimivat kuluttajatasoisellakin laitteistolla (CPU ja GPU) käyttökelpoisella nopeudella (Dong ym. 2014; Dong ym. 2015; Dong, Loy ja Tang 2016).

FSRCNN on SRCNN:ää nopeampi, mutta tuottaa siitä huolimatta parempia tuloksia (Dong, Loy ja Tang 2016). Se siis soveltuu korvaamaan SRCNN:n.

Dong, Loy ja Tang (2016) eivät erikseen mainitse pakkausartefaktien poistoa, joten ilmeisesti ARCNN ja FSRCNN eivät ole tarkoitettu samaan käyttötarkoitukseen, eivätkä ne tällöin suoraan kilpaile keskenään. Tämä on loogista, koska kuvien pakkausartefaktien määrä ei riipu suoraan resoluutiosta, jolloin oletettavasti myös paras skaalaustapa vaihtelee.

Käytännön sovellusten kannalta kahden kilpailevan, mutta samanlaisen menetelmän ylläpitäminen ei ole kannattavaa: FSRCNN on SRCNN:ää huomattavasti nopeampi ja parempaa jälkeä tuottava (Dong, Loy ja Tang 2016), joten läheisesti SRCNN:ään liittyvä ARCNN (Dong ym. 2015) ei todennäköisesti voi saavuttaa samaa tasoa. Toisaalta skaalausta vaativissa kuvissa on usein runsaasti pakkausartefakteja, joten FSRCNN:n käyttökohteet ovat rajalliset. ARCNN:n artefaktienpoisto-ominaisuuden sulauttaminen FSRCNN:n valinnaiseksi osaksi mahdollistaisi kummankin menetelmän hyvien puolien hyödyntämisen. Tällöin artefaktienpoiston saisi kytkettyä päälle tarpeen vaatiessa, ja muulloin voitaisiin hyötyä nopeammasta ja tarkemmasta skaalauksesta.

Kaikki kolme menetelmää käyttävät virhefunktionaan MSE-arvoa, ja mm. MSE-arvoon perustuvaa PSNR-arvoa kuvien laadun tarkasteluun. Yang, Ma ja Yang (2014) toteavat PSNR:n ja SSIM:n mittaavan kuvanlaatua huonosti, joten parempien mittausmenetelmien kehittäminen voisi tuottaa parempia koulutustuloksia. MSE eli ole sopiva mittari, koska se on absoluuttinen, eikä se tällöin ota huomioon silmän vaihtelevaa vastetta kontrastiin ja värien muutoksiin.

## 5.2 Sovelluskohteita

Kuvanskaalausmenetelmille on monia mahdollisia sovellusalueita. Niitä voisi esimerkiksi soveltaa sähköisten kuvien entisöinnissä, jos alkuperäinen kuva on pieni tai muuten huonolaatuinen.

Kuutiollista interpolaatiota parempi kuvanskaalausmenetelmä on varmasti kiinnostava myös muun kuvankäsittelyn saralla. Jokaisen skaalaukertoimen erillinen kouluttaminen vaikeuttaa menetelmien integraatiota ohjelmiin, mutta ongelmaa voisi lievittää esikouluttamalla yleisiä kertoimia. Niistä voisi sitten valita lähimmän halutun konfiguraation ja käyttää sitä likiarvoisesti, muuttaen vain dekonvoluution askelkokoja.

Verkkopalvelujen olisi mahdollista säästää tiedonsiirtokustannuksissaan lähettämällä pienikokoisia kuvia ja skaalaamalla ne vasta vastaanottajan laitteella. Tätä kuitenkin rajoittaa käytettyjen menetelmien laskennallinen vaativuus, joka johtaisi kannettavien laitteiden akkujen nopeaan tyhjenemiseen. Toinen ongelma on se, että iso osa internetin selaamisesta tapahtuu laitteilla, joilla ei ole pöytätietokonetasoista laskentatehoa. Tällöin verkon selaaminen olisi hidasta. Käyttöönottoa rajoittaisi myös ohjelmistotuen puute, sillä menetelmää ei voisi hyödyntää ilman erillistä selaimen liitännäistä, suoraa integraatiota sovellukseen tai erillistä asiakasohjelmaa. Menetelmät kuitenkin tuottavat niin suuren parannuksen kuvanlaatuun, että suoritusajavaativuudeltaan optimoidun version lisääminen esimerkiksi selaimiin parantaisi käyttäjäkokemusta merkittävästi.

Skaalauksen kohteiden ei tarvitse olla alun perin sähköisiä, sillä analogisiin formaatteihin tallennetut kuvat voidaan skannata. Tämän ansiosta esimerkiksi VHS-nauhan sisältö voidaan skaalata tarkempaan muotoon. Menetelmät ovat ajankohtaisia, koska analoginen tal-

lennusmedia haalistuu ajan myötä. Niiden käsittely vaatii kuitenkin muutoksia menetelmiin, sillä artefaktit syntyvät häviöllisten pakkausmenetelmien lisäksi tallennusmedian fyysisistä ominaisuuksista. Artefaktien erilaisesta luonteesta huolimatta erityisesti ARCNN olisi tähän sovelias menetelmä, sillä se käsittelee nyky muodossaankin sattumanvaraisia kuvan häiriöitä. Tietyissä formaateissa on eroa myös resoluution skaalaamisen tarpeessa, koska analogiset kuvat voidaan muuttaa sähköisiksi korkearesoluutioisella laitteistolla.

Kyky tuottaa matalaresoluutioista kuvista korkealaatuisia tuo mukanaan myös uhkia, sillä se voi mahdollistaa piratismia. Esimerkiksi julisteiden ja muiden painotuotteiden esikatse-lukuvista saatetaan voida tuottaa luvattomia kopioita. Tämä ei välttämättä ole suuri muutos nykytilanteeseen, sillä jo nyt tekijänoikeuksia rikkovat tahot voivat tilata yhden kappaleen kopioitavaa tuotetta, skannata tai muuten kopioida sen ja sitten tuottaa omia versioitaan.

Toinen mahdollinen haitta on kuvaväärennösten tunnistamisen vaikeuttaminen. Manipuloidun kuvan skaalaaminen super-resoluutiomenetelmällä voi peittää muokkaamisesta syntyviä artefakteja, mikä sotkee niiden tunnistamiseen perustuvien algoritmien toimintaa. Näin esimerkiksi kaksi toisistaan resoluutioltaan tai laadultaan poikkeavaa kuvaa voidaan yhdistää helpommin.

## 6 Yhteenveto

Kuvien resoluutionskaalaamiseen on kehitetty useita menetelmiä. Niille on tarvetta, koska suurten kuvatiedostojen säilyttäminen ja siirtäminen, erityisesti verkossa, on kallista. Lisäksi kuvanlaatua alentavat kameroiden tekniset rajoitteet sekä tiedostojen häviöllinen muokkaaminen ja uudelleentallentaminen.

Tässä tutkielmassa keskityttiin tarkastelemaan sellaisia kuvaskaalausmenetelmiä, joissa skaalattava kuva jaetaan tilkkuihin, tilkkujen piirteet vahvistetaan ja sitten yhdistetään tarkemmaksi kuvaksi. SRCNN on tällaisen menetelmän perusversio, josta on johdettu pakkausartefakteja poistava ARCNN sekä muuten tehokkaampi FSRCNN. Ne kaikki tuottavat hyvälaatuisia kuvia ja toimivat kohtuullisella nopeudella.

Käsiteltäviä menetelmiä voisi kehittää yhdistämällä niitä toisiinsa. SRCNN:ään perustuva ARCNN ei ole yhtä tehokas kuin FSRCNN, toisaalta FSRCNN ei poista skaalausta vaativissa kuvissa usein esiintyviä pakkausartefakteja kuten ARCNN. ARCNN:n artefaktinpoistokerroksen lisääminen FSRCNN:ään tuottaisi mahdollisesti yleiskäyttöisemmän ja nopeamman menetelmän kuin kumpikaan sen osista yksinään.

Yhden kuvan super-resoluutiossa riittää muutenkin tutkittavaa, sillä luonteeltaan se on huonosti asetettu ongelma äärettömällä määrällä ratkaisuja. Ongelmallisia siinä ovat erityisesti kappaleiden äärirajat, erilaiset kuviot ja lukuisten lähteiden aiheuttama satunnainen kohina. Parempien kuvanlaadun mittauskeinojen kehittäminen voi johtaa parempiin tuloksiin. Lisäksi mahdollisia neuroverkkokonfiguraatioita ja -tyyppejä on valtavia määriä.



## Lähteet

Alpaydin, Ethem. 2010. *Introduction to machine learning*. 2nd ed. Adaptive computation and machine learning. MIT Press.

Dong, Chao, Yubin Deng, Chen Change Loy ja Xiaoou Tang. 2015. “Compression Artifacts Reduction by a Deep Convolutional Network”. Teoksessa *2015 IEEE International Conference on Computer Vision (ICCV)*, 576–584. IEEE, joulukuu. doi:10.1109/ICCV.2015.73.

Dong, Chao, Chen Change Loy, Kaiming He ja Xiaoou Tang. 2014. “Learning a Deep Convolutional Network for Image Super-Resolution”. Teoksessa *Computer Vision – ECCV 2014*, toimittanut David Fleet, Tomas Pajdla, Bernt Schiele ja Tinne Tuytelaars, 8692:184–199. Springer International Publishing. doi:10.1007/978-3-319-10593-2\_13.

Dong, Chao, Chen Change Loy ja Xiaoou Tang. 2016. “Accelerating the Super-Resolution Convolutional Neural Network”. Teoksessa *Computer Vision – ECCV 2016*, toimittanut Bastian Leibe, Jiri Matas, Nicu Sebe ja Max Welling, 9906:391–407. Springer International Publishing. doi:10.1007/978-3-319-46475-6\_25.

Goodfellow, Ian, Yoshua Bengio ja Aaron Courville. 2016. *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.

Hamilton, Eric. 1992. *JPEG File Interchange Format v1.02*. Viitattu 18. helmikuuta 2019. <https://www.w3.org/Graphics/JPEG/jfif3.pdf>.

ISO/IEC JTC 1. 1993. *ISO/IEC 10918-1*. Viitattu 17. helmikuuta 2019. <https://www.w3.org/Graphics/JPEG/itu-t81.pdf>.

ITU-R. 2011. *BT.601: Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios*, maaliskuu. <https://www.itu.int/rec/R-REC-BT.601-7-201103-I/en>.

Popovici, Irina, ja Wm. Douglas Withers. 2007. “Locating Edges and Removing Ringing Artifacts in JPEG Images by Frequency-Domain Analysis”. *IEEE Transactions on Image Processing* 16, numero 5 (toukokuu): 1470–1474. doi:10.1109/TIP.2007.891782.

- Poynton, Charles A. 2012. *Digital Video and HD : Algorithms and Interfaces*. Nide 2nd ed. Morgan Kaufmann.
- Simonyan, Karen, ja Andrew Zisserman. 2014. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. *arXiv:1409.1556 [cs]* (4. syyskuuta). arXiv: 1409.1556. <http://arxiv.org/abs/1409.1556>.
- Timofte, Radu, Vincent De ja Luc Van Gool. 2013. “Anchored Neighborhood Regression for Fast Example-Based Super-Resolution”. Teoksessa *2013 IEEE International Conference on Computer Vision, 1920–1927*. IEEE, joulukuu. doi:10.1109/ICCV.2013.241.
- World Wide Web Consortium. 2003. “Portable Network Graphics (PNG) Specification (Second Edition)”. Viitattu 23. maaliskuuta 2019. <https://www.w3.org/TR/2003/REC-PNG-20031110/>.
- . 2019. “JPEG JFIF”. Viitattu 28. helmikuuta. <https://www.w3.org/Graphics/JPEG/>.
- Xiong, Zhiwei, Xiaoyan Sun ja Feng Wu. 2009. “Image hallucination with feature enhancement”. Teoksessa *2009 IEEE Conference on Computer Vision and Pattern Recognition, 2074–2081*. IEEE, kesäkuu. doi:10.1109/CVPR.2009.5206630.
- Yang, Chih-Yuan, Chao Ma ja Ming-Hsuan Yang. 2014. “Single-Image Super-Resolution: A Benchmark”. Teoksessa *Computer Vision – ECCV 2014*, toimittanut David Fleet, Tomas Pajdla, Bernt Schiele ja Tinne Tuytelaars, 8692:372–386. Springer International Publishing. doi:10.1007/978-3-319-10593-2\_25.
- Yang, Jianchao, John Wright, Thomas Huang ja Yi Ma. 2008. “Image super-resolution as sparse representation of raw image patches”. Teoksessa *2008 IEEE Conference on Computer Vision and Pattern Recognition, 1–8*. IEEE, kesäkuu. doi:10.1109/CVPR.2008.4587647.