

Konvoluutioneuroverkkojen kehitys ja tulevaisuus konenäköön liittyvissä tehtävissä

TURUN YLIOPISTO
Tietotekniikan laitos
TkK-tutkielma
Tietotekniikka
Kesäkuu 2025
Aaron Aaltonen

TURUN YLIOPISTO
Tietotekniikan laitos

AARON AALTONEN: Konvoluutioneuroverkkojen kehitys ja tulevaisuus konenäköön
liittyvissä tehtävissä

TkK-tutkielma, 27 s.
Tietotekniikka
Kesäkuu 2025

Tutkielma käsittelee konvoluutioneuroverkkojen kehitystä ja tulevaisuutta konenäön näkökulmasta. Konvoluutioneuroverkkojen kehitystä tarkastellaan vuodesta 1998 aina tähän päivään asti keskittyen suurimpiin kehityspisteisiin arkkitehtuurissa aina LeNetista CovNextiin.

Tutkielmassa käydään läpi yleisarkkitehtuurin kehittyminen ja siihen liittyvät osa-alueet kirjallisuuteen pohjautuen. Lisäksi pohditaan, onko konvoluutioneuroverkolla nykypäivässä kuvamuuntimien aikakaudella paikkaa konenäön kehityksessä avaamalla aiempia arkkitehtuurirakenteen kehityksiä. Näihin pohjaamalla tutkielma hahmottaa tulevaisuuden kehitysmahdollisuuksia.

Konvoluutioneuroverkot ovat edelleen hyvin olennaisia ja niitä voidaan hyödyntää monipuolisesti. Erityisen merkittävä rooli niillä on kuvamuuntajien hyödyntämisessä hybridimalleissa. Konvoluutioneuroverkot toimivat paremmin tietyissä rajoitetuissa ympäristöissä kuvamuuntajiin nähden, tämän vahvistaa niiden roolia. Seuraavina vuosina lisätutkimuksessa tulisi keskittyä erityisesti hybridimalleihin yhdessä kuvamuuntajien kanssa sekä tehokkaamman arkkitehtuurin kehittämiseen. Tutkimuksen valossa konvoluutioneuroverkkojen tulevaisuus konenäössä on valoisa ja täynnä kehittämis- ja soveltamismahdollisuuksia.

Asiasanat: konvoluutioneuroverkko, transformerit, arkkitehtuuri

UNIVERSITY OF TURKU
Department of Computing

AARON AALTONEN: Konvoluutioneuroverkkojen kehitys ja tulevaisuus konenäköön
liittyvissä tehtävissä

Bachelor's Thesis, 27 p.
Tietotekniikan laitos
June 2025

The thesis explores the development and future of convolutional neural networks (CNNs) from the perspective of computer vision. The development of CNNs is examined from 1998 to the present day, focusing on the major architectural milestones from LeNet to ConvNeXt.

The thesis discusses the evolution of general CNN architecture and its related components based on existing literature. Furthermore, it considers whether CNNs still have a role in the era of vision transformers by analyzing past architectural advancements. Building on these insights, the thesis outlines possible future development directions.

CNNs remain highly relevant and versatile in modern applications. They play a particularly significant role in hybrid models that incorporate vision transformers. In fact, CNNs outperform vision transformers in certain constrained environments, which reinforces their importance. In the coming years, further research should focus especially on hybrid models and the development of more efficient architectures. Based on current research, the future of convolutional neural networks in computer vision appears bright, with numerous opportunities for both development and application.

Keywords: Convolutional Neural Network, Transformer, Architecture

Sisällys

| | | |
|----------|---|-----------|
| 1 | Johdanto | 1 |
| 2 | Neuroverkot | 4 |
| 2.1 | Neuronit ja aktivaatiofunktio | 5 |
| 2.2 | Syväoppiminen | 7 |
| 2.3 | Ohjattu ja ohjaamaton oppiminen | 7 |
| 3 | Konvoluutioneuroverkko arkkitehtuuri | 8 |
| 3.1 | Konvoluutiokerros | 9 |
| 3.2 | Poolauskerros ja kytketty kerros | 10 |
| 4 | Neuroverkkojen kehityskulku ja käyttökohteet | 12 |
| 4.1 | LeNet | 12 |
| 4.2 | AlexNet | 13 |
| 4.3 | ZFNet | 14 |
| 4.4 | GoogLeNet | 15 |
| 4.5 | ResNet | 16 |
| 4.6 | DenseNet | 17 |
| 4.7 | Kuvamuuntimet | 18 |
| 4.8 | EfficientNet | 19 |
| 4.9 | ConvNext | 20 |

| | |
|--|-----------|
| 4.10 Käyttötarkoituksia | 21 |
| 5 Konvoluutioneuroverkkojen tulevaisuus | 22 |
| 6 Yhteenveto | 26 |
| Lähdeluettelo | 28 |

Kuvat

| | | |
|-----|--|----|
| 2.1 | Kuvassa näkyy hyvin yksinkertainen syöte eteenpäin neuroverkossa (engl. feedforward neural network), joka koostuu kolmesta eri kerroksesta; syötekerroksesta (engl. input layer), piilokerroksesta (engl. hidden layer) ja ulostulokerroksesta (engl. output layer). | 5 |
| 3.1 | Kuvassa näkyy yksinkertainen konvoluutioverkko, jossa on merkittäviä kerrokset numerottain. Kuva lisensoitu (https://commons.wikimedia.org/wiki/File:Typical_cnn.png) | 9 |
| 3.2 | Kuvassa ovat eri poolausmenetelmät visuaalisesti kuvattuna. | 11 |
| 4.1 | Kuvassa on AlexNetin ja LeNetin eroavaisuuksista visuaalinen esitys. . | 14 |
| 4.2 | Kuvassa esitellään kuvamuuntimen piirteet. | 19 |
| 5.1 | Visuaalinen esitys peräkkäisestä yhdistelystä. | 22 |
| 5.2 | Visuaalinen esitys rinnakkaisesta yhdistelystä. | 23 |
| 5.3 | Visuaalinen esitys hierarkisesta yhdistelystä. | 23 |
| 5.4 | Kuvassa on alkuvaiheessa tapahtuva fuusio. | 23 |
| 5.5 | Kuvassa on loppuvaiheessa tapahtuva fuusio. | 24 |

Taulukot

| | | |
|-----|---|----|
| 1.1 | Hakutermit ja niitä vastaavat BibTeX-viitteet | 3 |
| 5.1 | CNN, ViT, Hybridi mallien vertailu | 25 |

1 Johdanto

Konvoluutioneuroverkkojen (engl. convolutional neural network) kehittyminen on ollut yksi merkittävimmistä edistysaskelista konenäössä, erityisesti kuvien tunnistuksessa ja luokitellussa. Niiden historia alkoi jo 1950-1960 -luvuilla, kun David Hubel ja Torsten Wiesel julkaisivat tutkimuksensa kissan visuaalisesta aivokuoresta [1]. Tämä tutkimus innoitti Kunihiko Fukushiman julkaisemaan ensimmäisen tutkimuksen konvoluutioneuroverkoista käyttäen Relu-aktivointitoimintoa vuonna 1969[2]. Vaikkakin konvoluutioneuroverkot on keksitty jo 1960-luvulla, niiden kehittäminen on lisääntynyt vasta 2000-luvun alkupuolella. Tämän tutkielman tarkoituksena on tutkia tätä kehitystä 2000-luvun alkupuolella ja selvittää näiden uusien neuroverkkomallien käyttötarkoituksia.

Tutkimuskysymykset tässä tutkielmassa ovat seuraavat:

1. Miten konvoluutioneuroverkot ovat kehittyneet vuosien aikana?
2. Mihin konvoluutioneuroverkkomalleja käytetään konenäön kannalta?
3. Onko konvoluutioneuroverkoilla tulevaisuutta?

Tutkielmassa käsitellään ensin taustaa konvoluutioneuroverkoille, tämän jälkeen kolmannessa luvussa tutkielma etenee kuvaamaan tarkemmin konvoluutioneuroverkkoja ja niiden arkkitehtuuria, kuten konvoluutiokerrosta, poolauskerrosta ja täysin kytkettyä kerrosta. Tutkielman neljännessä luvussa siirrytään käsittelemään, milaista kehitystä on tapahtunut konvoluutioneuroverkkomalleissa ja niiden nykyisissä

käyttötarkoituksissa konenäön osalta. Tutkittaviksi malleiksi rajattiin tutkimusky-symysten näkökulmasta olennaisimmat mallit, jotka ovat vaikuttaneet merkittävästi konvoluutioneuroverkkojen kehitykseen. Viides luku pohtii konvoluutioneuroverkkojen tulevaisuutta. Tutkielman viimeinen luku on yhteenvetoluku.

Tämän tutkielman tutkimusartikkelien tiedonhakuun on hyödynnetty useita eri tietokantoja. Tietokannat, joita on hyödynnetty ovat Web of science, Volter ja Association for Computing Machinery digital library. Lisäksi tiedonhaussa ja käsitteiden ymmärtämisessä on hyödynnetty tekoälyä. Taulukossa 1.1 on kuvattuna tiedonhakuun käytetyt termit.

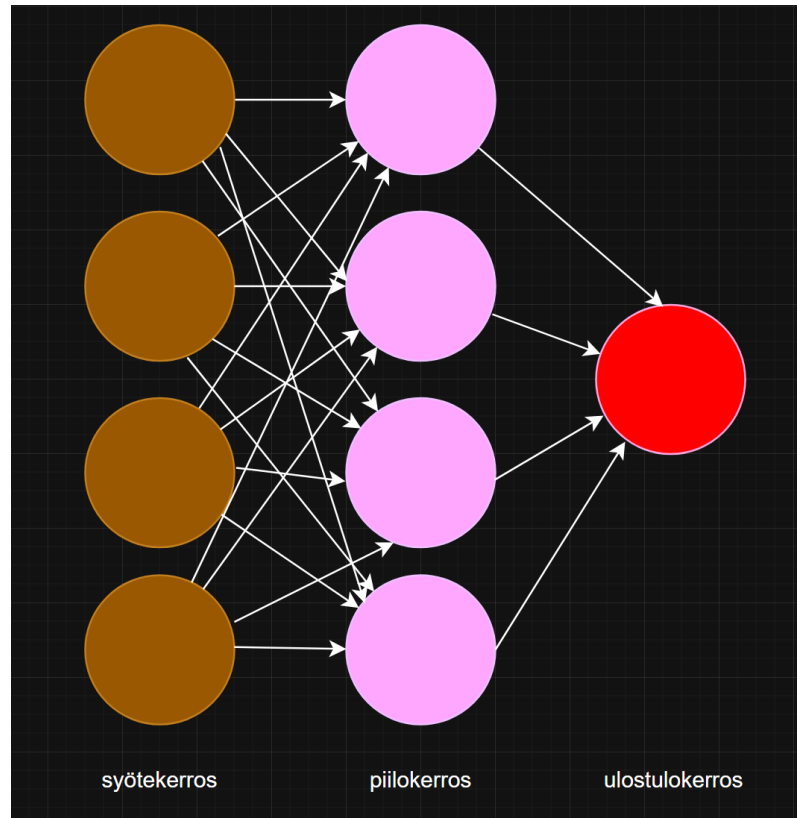
Taulukko 1.1: Hakutermit ja niitä vastaavat BibTeX-viitteet

| Hakutermi | Viitteet |
|---|---------------|
| "CNN architecture"AND "convolutional neural network architecture" | [3], [4], [5] |
| "LeNet"AND "convolutional network" | [6] |
| "AlexNet"AND "ImageNet" | [7], [8] |
| "ZFNet"AND "visualization" | [9] |
| "VGGNet"AND "3x3 convolutions" | [10] |
| "GoogLeNet"OR "Inception module" | [11] |
| "ResNet"AND "skip connections" | [12] |
| "DenseNet"AND "feature reuse" | [13] |
| "Vision Transformer"OR "ViT"AND "CNN hybrid" | [14], [15] |
| "EfficientNet"AND "model scaling" | [16] |
| "ConvNeXt"AND "modern CNN" | [17] |
| "CNN"AND "application"AND "healthcare" | [18], [19] |
| "CNN"AND "fashion" | [15], [20] |
| "CNN"AND "embedded systems" | [19], [21] |
| "human-centered AI"AND "artificial neural networks" | [22] |
| "deep learning"AND "neural networks" | [23] |
| "bio-inspired computing"AND "neurocomputing" | [24] |
| "deep learning"AND "representation learning" | [25] |
| "dropout"AND "overfitting prevention" | [26] |
| "handwritten character recognition"AND "CNN" | [27] |
| "Hubel"AND "Wiesel"AND "Cat" | [1] |
| "Fukushima"AND "Multilayered Network" | [2] |

2 Neuroverkot

Neuroverkot (engl. artificial neural network, ANN) ovat laskennallisia järjestelmiä, joiden perusta tulee vahvasti biologisten hermostojen toiminnasta. Neuroverkot koostuvat suuresta määrästä toisiinsa kytketyistä laskentayksiköistä eli keinotekoisista hermoista (engl. artificial neurons). Nämä työskentelevät yhdessä hajautetulla tavalla oppiakseen syötteen avulla ja kehittääkseen mahdollisimman optimoidun ulossyötteen. [3] Neuronit koostuvat painon ja syötteen summasta, jota seuraa ei-lineaarinen aktivaatiofunktio. Aktivaatiofunktion täytyy olla derivoituva, jotta verkkoa voidaan opettaa tehokkaasti takaisinkytkentäalgoritmin (engl. backpropagation) avulla, joka perustuu virheen gradientin laskemiseen. [22]

Käytännössä neuronit ovat algoritmeja, jotka ottavat vastaan useita syötteitä ja muuntavat ne painotetun laskennan kautta ulos tulokseksi. Tämä prosessi tapahtuu kuvan 2.1 esittämän neuroverkon piilokerroksessa. Yleisesti neuroverkoissa on useampia piilotettuja kerroksia. Piilokerrokset säätelevät painokertoimia siten, että verkon ulostulon ja tavoitearvon välinen virhe, eli kustannusfunktio (engl. cost function), pienenee. Tätä säätelee ohjaa takaisinkytkentäalgoritmi. Kyseistä prosessia kutsutaan oppimiseksi. Kun neuroverkossa on useampi piilokerros, sitä kutsutaan syväneuroverkoksi (engl. deep neural network) ja oppimisprosessia kutsutaan syväoppimiseksi (engl. deep learning). [3]



Kuva 2.1: Kuvassa näkyy hyvin yksinkertainen syöte eteenpäin neuroverkossa (engl. feedforward neural network), joka koostuu kolmesta eri kerroksesta; syötekerroksesta (engl. input layer), piilokerroksesta (engl. hidden layer) ja ulostulokerroksesta (engl. output layer).

2.1 Neuronit ja aktivaatiofunktio

Piilokerros koostuu joukosta neuroneja, jotka käsittelevät edellisten kerrosten syötettä. Kukaan neuroni on yksittäinen laskentayksikkö, mutta kun ne toimivat yhdessä ne muodostavat piilokerroksen, joka vastaa tietyn tason piirteiden tunnistamisesta. Jotta piilokerroksen neuronit voisivat oppia ei-lineaarisia riippuvuuksia syötteen ja tavoitevastauksen välillä, se käyttää aktivaatiofunktiota. Aktivaatiofunktion tavoitteena on määrittää, aktivoituuko neuroni annetulla syötteellä vai ei. Jos aktivaatiofunktiota ei olisi, kerrokset voisivat toimia vain lineaarisina yhdistelminä eivätkä pystyisi mallintamaan monimutkaisia ilmiöitä. [23] Yleisimpiä aktivaatiofunktiota

ovat seuraavat:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.1)$$

Sigmoid-aktivaatiofunktio tuottaa arvon välillä 0 ja 1. Se soveltuu erityisesti binääriluokittelutehtäviin, mutta syvissä verkkoarkkitehtuureissa sen käyttö on vähentynyt gradienttien häviämisiongelmiin vuoksi.

$$\text{ReLU}(x) = \max(0, x) \quad (2.2)$$

ReLU (engl. rectified linear unit) on yksinkertainen ja tehokas aktivaatiofunktio, joka suodattaa negatiiviset arvot nolaksi ja säilyttää positiiviset sellaisinaan. Se nopeuttaa oppimista ja on nykyään yleisin aktivointifunktio syväverkoissa.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.3)$$

Tanh-aktivaatiofunktio tuottaa arvon välillä -1 ja 1 . Se on symmetrinen nollan suhteen ja toimii usein paremmin kuin sigmoid, koska sen keskiarvo on lähempänä nollaa, mikä auttaa verkon oppimista. [24]

Aktivaatiofunktion täytyy olla derivoituva, jotta verkkoa voidaan opettaa takaisinkytkentäalgoritmeilla, joka perustuu virhegradientin laskemiseen ja verkon painojen säätämiseen. [23] Piilokerroksen neuronien tuottama aktivaatio lähetetään eteenpäin seuraavalle kerrokselle, jotta verkko muodostaa monitasoisen hierarkisen rakenteen piirteiden oppimiseen. Näin piilokerrokset yhdessä aktivaatiofunktioiden kanssa muodostavat neuroverkon ytimen. [25]

2.2 Syväoppiminen

Syväoppiminen (engl. deep learning) on prosessi, jonka kautta monikerroksiset neuroverkot oppivat. Oppiminen tapahtuu, kun neuroverkon sisäisten painokertoimien arvoja muutetaan neuroverkon tuloksen perusteella. Jokaisella neuroverkon tuloksella on numeraalinen arvo, jota opetusprosessin kustannusfunktio (engl. cost function) vertaa haluttuun vastaukseen. Jos vastaus ei vastaa haluttua arvoa, neuroverkko muokkaa itsensä vastavirta-algoritmia (engl. back-propagation) hyödyntäen. Vastavirta-algoritmi toteuttaa gradienttivakautuksen taaksepäin kulkevana virtauksena, joka peilaa eteenpäin kulkevaa aktivointisignaalia, joka lasketaan monikerroksisen neuroverkon neuroneissa. [22]

2.3 Ohjattu ja ohjaamaton oppiminen

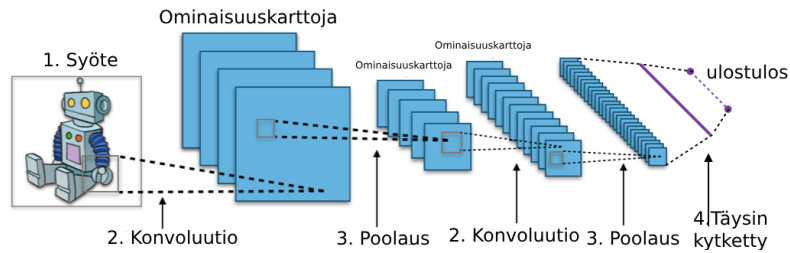
Ohjatussa oppimisessa (engl. supervised learning) neuroverkko oppii käyttämällä esimerkkejä, joilla on ennalta määritellyt tunnisteet (engl. labels). Jokaisessa harjoitusesimerkissä on joukko syötearvoja (vektoreita) ja yhteen tai useampaan niistä liittyvä ennalta määritelty tulostearvo. Tämän oppimismuodon tavoitteena on vähentää mallin kokonaisluokitteluvirhettä opettamalla se laskemaan harjoitusesimerkkien tulostearvot oikein. [3]

Ohjaamattomassa oppimisessa (engl. unsupervised learning) harjoitusdata ei sisällä tunnisteita. Neuroverkon menestys määritellään yleensä sen mukaan, pystyykö verkko vähentämään tai lisäämään liittyvää kustannusfunktiota. Konvoluutioneuroverkot keskittyvät yleensä ohjattuun oppimiseen. [3]

3 Konvoluutioneuroverkko arkkitehtuuri

Konvoluutioneuroverkon perustoiminta voidaan jakaa neljään eri kerrokseen. Nämä vaiheet ovat nähtävissä kuvassa 3.1.

1. **Syötekerros** vastaanottaa kuvan pikseliarvot matriisimuodossa.
2. **Konvoluutiokerros** laskee paikallisia piirteitä syötteestä käyttämällä konvoluutiotoimintoa. Aktivointifunktiota, kuten ReLU, sovelletaan elementtikohdaisesti aktivaatiotuloksiin.
3. **Poolauskerros** suorittaa alinäytteenoton, joka pienentää ominaisuuskartan (engl. feature map) kokoa ja parantaa laskennallista tehokkuutta.
4. **Täysin kytketyt kerrokset** käyttäytyvät kuten klassisissa neuroverkoissa ja tuottavat lopulliset luokittelutulokset. [3]



Kuva 3.1: Kuvassa näkyy yksinkertainen konvoluutioverkko, jossa on merkittynä kerrokset numerottain. Kuva lisensoitu (https://commons.wikimedia.org/wiki/File:Typical_cnn.png)

Seuraavat alakappaleet avaavat enemmän konvoluutiokerrosta ja poolauskerrosta.

3.1 Konvoluutiokerros

Tärkein rakenneosia konvoluutioneuroverkolle on konvoluutiokerros, joka koostuu joukosta konvoluutiosuodattimia eli ytimiä (engl. kernels). Jokainen verkon neuronit toimii ytimenä, joka oppii tunnistamaan tiettyjä visuaalisia piirteitä. Konvoluutiokerros toimii jakamalla kuvan pienempiin alueisiin, joita kutsutaan reseptiivisiksi kentiksi (engl. receptive field). Tällainen alueellinen jaottelu mahdollistaa paikallisten piirteiden, kuten reunojen ja muotojen, tehokkaan tunnistamisen. Konvoluutiokerroksessa käytettävät suodattimet jakavat painonsa koko syötteen alueelle, mikä mahdollistaa saman piirteiden tunnistamisen eri kohdissa kuvaa, tätä kutsutaan painojen jakamiseksi (engl. weight sharing). [4]

Konvoluution käyttö tapahtuu, kun ydin liikuu eli konvolvoituu syötekuvan yli ja suorittaa elementtikohtaisen kertolaskun reseptiivisen kentän ja ytimen välillä. Tulos muodostaa uuden kuvan eli esityksen ominaisuuskartan (engl. feature map), joka kuvaa kyseisen piirteiden esiintymistä syötteen eri sijainneissa. Matemaattisesti

prosessin voi esittää näin: [4]

$$f_l^k(\mathbf{p}, \mathbf{q}) = \sum_c \sum_{x,y} i_c(x, y) \cdot e_l^k(u, v)$$

Funktiossa $i_c(x, y)$ on syötekuvasensori I_C , joka kerrotaan elementin $e_l^k(u, v)$:llä, joka k :nnen konvoluutiokernelin k_l indeksin l :nnellä kerroksella. k :nnen konvoluutio-operaation tuoksena saatava ominaisuuskartta voidaan ilmaista muodossa [4]:

$$\mathbf{F}_l^k = [f_l^k(1, 1), \dots, f_l^k(p, q), \dots, f_l^k(P, Q)]$$

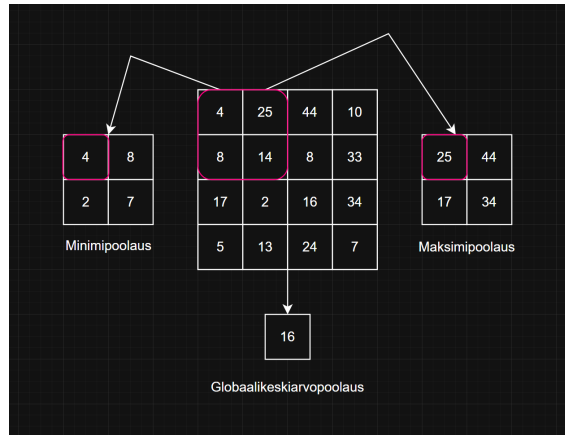
3.2 Poolauskerros ja kytketty kerros

Poolauskerroksen avulla tiivistetään konvoluutiokerrosten tiedot. Sen tehtävänä on siis yhdistää konvoluutiokerrosten tuottamat ominaisuuskartat ja supistaa ne pienemmiksi tallentaen niiden tärkeimmät piirteet, mutta samalla vähentäen datan ulottuvuuksia ja laskentatehoa. Poolauskerros lisää verkon invarianssia pienten siirtymien ja vääristymien suhteen. Yleisimmät poolausmenetelmät ovat:

- **Maksimipoolaus (engl. max pooling)** valitsee suurimman arvon kohdealueelta.
- **Minimipoolaus (engl. min pooling)** valitsee pienimmän arvon.
- **Globaalikeskiarvopoolaus (engl. global average pooling)** laskee keskiarvon koko ominaisuuskartalta. [5]

Kuvasta 3.2 näkee esityksen näistä eri poolausmenetelmistä. Poolauskerros vähentää

ylipassiivisuuden (engl. overfitting) riskiä, nopeuttaa oppimista ja säilyttää tärkeimmät visuaaliset piirteet seuraavia kerroksia varten. [5]



Kuva 3.2: Kuvassa ovat eri poolausmenetelmät visuaalisesti kuvattuna.

Täysin kytkettyä kerrosta käytetään useimmiten verkon lopussa luokittelua varten. Toisin kuin poolaus- ja konvoluutiokerrokset, se on globaalioperaatio eli tulkitsee kaikkia ominaisuuskarttoja. Kytketty kerros ottaa syötteen ominaisuuksien erotusvaiheista ja analysoi globaalisti kaikkien edeltävien kerrosten tulokset. Näin ollen se muodostaa valittujen ominaisuuksien epälineaarisen yhdistelmän, jota käytetään datan luokitteluun. [5]

4 Neuroverkkojen kehityskulku ja käyttökohteet

Tässä luvussa käydään läpi konvoluutioneuroverkkojen kehitystä useamman ison kehityksen kautta. Ensin kuvataan, miten konvoluutioneuroverkkojen kehitys alkoi LeNetin kautta. Sen jälkeen käsitellään konvoluutioneuroverkkojen kehitystä 2010-luvulla useamman mallin sekä erilaisten käyttötapausten kautta. Mallit, joita käsitellään, on valittu niiden kehittämien edistysaskelien kautta.

4.1 LeNet

LeNetin esitteli ensikerran Le Cun vuonna 1998. Tämä malli on tunnettu sen historiallisesta merkityksestä ensimmäisenä konvoluutioneuroverkkona, joka saavutti merkittävän suorituskyvyn käsinkirjoitettujen numeroiden tunnistustehtävissä. LeNet kykeni tunnistamaan numeroita ilman, että siihen vaikuttivat pienet virheet, kuvan kääntäminen, paikan siirtäminen tai mittakaavan muuttaminen. [6]

LeNet loi monta keskeistä konseptia, jotka ovat tulleet vakinaisiksi konvoluutioneuroverkoarkkitehtuureissa. Näitä konsepteja olivat esimerkiksi konvoluutio- ja poolauskerrosten käyttö, paikalliset reseptiiviset kentät, jaetut painot sekä vastavirta-algoritmi verkon kouluttamisessa. LeNet loi pohjan kaikille seuraavien vuosikymmenien kehityskuluille konvoluutioarkkitehtuurissa. [6]

4.2 AlexNet

Vuonna 2012 kehitetty AlexNet oli tärkeä läpimurto syväoppimisen hyödyntämisessä konenäössä. Mallin kehittivät Alex Krizhevsky, Ilya Sutskever ja Geoffrey Hinton Toronton yliopistosta. Alexnet voitti ImageNet Large Scale Visual Recognition Challenge -kilpailun, jossa sen virhemarginaali oli paljon pienempi kuin muilla kilpailun menetelmillä. [7]

AlexNet koostuu kahdeksasta kerroksesta, joista viisi on konvoluutiokerroksia ja loput kolme on täysin kytkettyjä kerroksia. Kuva 4.1. esittelee LeNetin ja AlexNetin rakenteelliset erot visuaalisessa muodossa. Mallissa hyödynnettiin ReLU-aktivaatiofunktioita, joka mahdollisti huomattavasti nopeamman oppimisen verrattuna LeNetin hyödyntämiin käytettyihin sigmoid-funktioihin tai yleisesti hyödynnettyyn tahn-funktioon. AlexNetin ratkaisuna tilallisen resoluution vähentämiseen oli hyödyntää maksimipoolauskerroksia keskiarvopoolaustakerrosten sijaan toisin kuin LeNetissä. Näin AlexNetin verkon laskennallista monimutkaisuutta saatiin pienennettyä ilman, että malli menetti tärkeimmät piirteensä. [7]

Aiemmin mainittujen piirteiden lisäksi Alexnetiin lisättiin tiputustekniikka (engl. dropout), jolla pyrittiin estämään verkon ylipassiivisuus erityisesti täysin kytketyissä kerroksissa. Tämä toimii käytännössä niin, että jokaisen koulutuskerroksen aikana satunnainen osa neuroneista poistetaan verkon käytöstä todennäköisyydellä p . Tiputustekniikka pakottaa verkon oppimaan useita redundanteja esityksiä tiedosta, sillä mikään neuroni ei voi luottaa siihen, että tietyt toiset neuronit ovat aina mukana. [26]



Kuva 4.1: Kuvassa on AlexNetin ja LeNetin eroavaisuuksista visuaalinen esitys.

4.3 ZFNet

ZFNet kehitettiin vuonna 2013 AlexNetin pohjalta parantamaan tilallista tarkkuutta ja verkon tulkittavuutta. Ennen ZFNetiä konvoluutioneuroverkkojen testaaminen oli paljon hankalampaa, sillä ei voitu tietää miten tai miksi muutokset rakenteessa saivat parannettua konvoluutioneuroverkon toimintaa. ZFNet hyödynsi AlexNetin perusrakennetta, mutta teki siihen muutoksia muuttaen muun muassa konvoluutiokerrosten suodatinkokoa ja askelpituutta (engl. stride). ZFNetissä pienennettiin AlexNetin suodatinkokoa 11×11 :stä kokoon 7×7 , ja samalla kerroksen askelarvoa vähennettiin kahteen, mikä mahdollisti vahvemman resoluutiopiirteiden havaitsemisen syötteen alkuvaiheessa. Näin malli pystyi analysoimaan tarkemmin kuvan yksityis-

kohtia heti mallin ensimmäisestä konvoluutiokerroksesta. [9]

ZFNetin dekonvoluutiopohjainen visualisointimenetelmä teki siitä merkittävän konvoluutioneuroverkkojen tulkinnan kannalta. Sen avulla tutkijat pystyivät visualisoimaan, mitä kukin kerros oppi. Tämän tekniikan avulla pystyttiin näkemään yksittäisten tasojen luomat ominaisuuskartat. Näin verkon toiminnasta saatiin aiempaa läpinäkyvämpi kuva ja se helpotti käytännön virheiden ja vahvuuksien tulkinnassa. ZFNetti mahdollisti esimerkiksi tulkinnan siitä, mikä osa AlexNetin suodattimista oli täysin passiivisia, tämä puolestaan johti parempaan suodattimien ja parametrien suunnitteluun. [9]

ZFNetin kehittämät konvoluutioneuroverkkojen tulkintamenetelmät mullistivat konvoluutioneurverkkojen ymmärryksen, vaikkakin ZFNet ei ollut arkkitehtuuristen muokkausten tasolla yhtä radikaali uudistus kuin seuraavien vuosien GoogLeNet tai ResNet olivat. [9]

4.4 GoogLeNet

GoogLeNet esiteltiin vuonna 2014 ja se on yksi konvoluutioneuroverkkojen merkittävimmistä innovaatioista. Sen kehittivät Christian Szegedy ja kollegat, heidän kehittämänsä malli tunnetaan myös nimellä Inception-v1. [11]

GoogLeNetin keskeinen innovaatio oli niin kutsuttu Inception-moduuli, jossa on useita eri kokoisia konvoluutioita kuten 1×1 , 3×3 ja 5×5 . Näitä konvoluutioita suoritetaan rinnakkain samassa kerroksessa. Tämä mahdollisti yksityiskohtaisten visuaalisten piirteiden havaitsemisen samanaikaisesti ja teki verkosta entistä ilmaisukykyisemmän ilman merkittävää kasvua sen parametrimäärässä. Lisäksi GoogLeNet hyödynsi 1×1 -konvoluutioita niin sanottuina pullonkauloina, joilla pyrittiin vähentämään kanavien määrää ja pienentää laskennallista kuormaa mallissa. [11]

Toinen merkittävä piirre GoogLeNetissä oli täysin kytkettyjen kerrosten poistaminen verkon lopusta. Tämän sijaan käytettiin globaalia keskiarvopoolauusta, joka

johti parametrimäärän vähenemiseen ja paransi yleistämiskykyä. GoogLeNetin kokonaisparametrimäärä oli vain noin 5 miljoonaa, mikä oli huomattavasti vähemmän kuin esimerkiksi ZFNetin yli 90 miljoonaa, mutta silti GoogLeNet saavutti tehokkaampia tuloksia. [11]

GoogLeNet edusti muutosta pois yksinkertaisesti syvempien verkkojen rakentamisesta kohti älykkäästi suunniteltuja arkkitehtuureja, joissa tehokkuus ja monipuolisuus yhdistyivät modulaariseen rakenteeseen. Se loi pohjan myöhemmille Inception-malleille ja oli suunnannäyttäjänä moderneille CNN-kehityksille.

4.5 ResNet

ResNet esiteltiin vuonna 2015 ja se merkitsi ratkaisevaa edistysaskelta syvien neuroverkkojen kehityksessä. ResNetin suurin innovaatio oli residuaaliyhteydet (engl. skip connections), jotka mahdollistivat erittäin syvien verkkojen tehokkaan koulutuksen ilman, että mallin suorituskyky heikkeni kerrosmäärän lisääntyessä. Tätä ennen syvien verkkojen oppiminen kärsi usein katoavan gradientin ongelmasta (engl. vanishing gradient problem), jossa virheen gradientti pieneni matkalla kohti alkuvaiheen kerroksia estäen mallia oppimasta. [12]

Residuaaliyhteydet ratkaisevat tämän ongelman lisäämällä identiteettikytkentöjä, joiden kautta alkuperäinen syöte pystyttiin ohjaamaan myöhempisiin kerroksiin suoraan. Tällöin verkon tarvitsee oppia vain niin sanottu jäännöstoiminto (engl. residual function) eli kuinka paljon se poikkeaa suoraan syötteestä. Tämä mahdollisti sen, että myös hyvin syvät verkot, kuten 50-, 101- tai jopa 152-kerroksiset mallit, voitiin kouluttaa ilman merkittävää ylipassiivisuutta tai tarkkuuden heikkenemistä. [12]

Residuaalilohkot ovat sen jälkeen tulleet osaksi lukemattomia moderneja arkkitehtuureja, mukaan lukien DenseNet, EfficientNet ja ConvNeXt. ResNet osoitti, että verkkojen syvyyttä voi kasvattaa hallitusti, kunhan arkkitehtuuri tukee tehokasta

tiedonkulkua myös syvissä kerroksissa. Tämä teki ResNetistä erään kaikkien aikojen vaikutusvaltaisimmista CNN-malleista.

4.6 DenseNet

DenseNet esiteltiin vuonna 2017, ja se toi uudenlaisen lähestymistavan syvien konvoluutioneuroverkkojen rakenteeseen. Mallin taustalla oli havainto siitä, että syvissä verkoissa voi esiintyä tiedonhukkaa ja tehottomuutta. Tätä ilmenee, kun jokainen kerros vain välittää tietoa seuraavalle kerrokselle ilman suoraa pääsyä alkuvaiheen piirteisiin. DenseNet ratkaisi tämän ongelman tiheillä liitoksilla (engl. dense connections) eli jokainen kerros vastaanotti syötteen kaikista sitä edeltävistä kerroksista, eikä vain sitä edeltävästä. [13]

Tiheät liitokset tekivät DenseNetistä erityisen tehokkaan piirteiden uudelleenkäytössä. Kun piirteet virtaavat läpi, koko verkon mallin ei tarvitse oppia samoja asioita uudelleen joka kerroksessa. Tämän ansiosta DenseNet-mallit ovat, sekä laskennallisesti tehokkaita, että tarkkoja ja ne vaativat vähemmän parametreja kuin perinteiset arkkitehtuurit samalla suorituskyvyllä. Lisäksi tiheä liitettävyys toimii ikään kuin luonnollisena säännöllistämisenä, joka vähentää ylipassiivisuuden riskiä. [13]

DenseNet koostuu useista tiheistä lohkoista (engl. dense blocks), joiden välissä käytetään siirtokerroksia (engl. transition layer) piirreavaruuden koon hallintaan. Tämä arkkitehtuuri mahdollistaa syvien verkkojen rakentamisen ilman, että tiedon siirtymiseen tarvitaan erikseen residuaaliyhteyksiä kuten ResNetissä. DenseNetin piirteet siirtyvät automaattisesti eteenpäin kerros kerrokselta tiheiden yhteyksien kautta.[13]

4.7 Kuvamuuntimet

Kuvamuuntimet (engl. vision transformers, ViT) pohjautuvat 2017 kehitettyyn transformer-arkkitehtuuriin (engl. transformers). Toisin kuin konvoluutioneuroverkot ViT-mallit jakavat kuvan palasiin (engl. patches) ja käsittelevät jokaista palasta osana isompaa palapeliä ja käsittelevät ne transformeri enkooderin (engl. encoder) läpi saadakseen ymmärryksen kuvan piireteistä. Kuvamuuntimet koostuvat viidestä vaiheesta, jotka on esitetty myös kuvassa 4.2:

1. Kuvien paloittelu (engl. image patching): Kuva, jonka koko on $H \times W \times C$, jaetaan N kuvapalaan, joiden koko on $P \times P \times C$. Tässä H on kuvan korkeus, W leveys ja C kanavien määrä. P tarkoittaa yksittäisen kuvapalan resoluutiota.

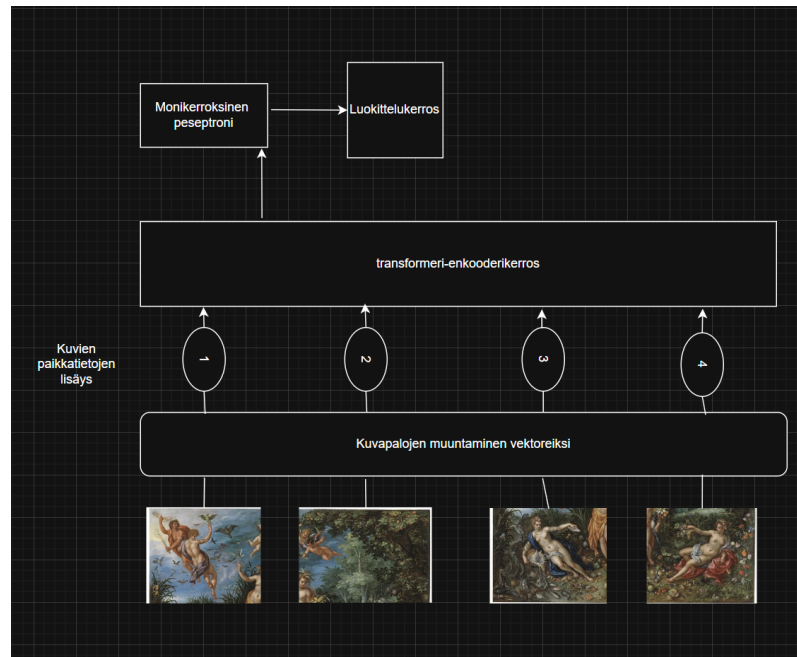
2. Kuvapalojen muuntaminen vektoreiksi (engl. linear transformation of patches to vectors): Jokainen kuvapala litistetään (engl. flatten) niin, että sen muodoksi tulee vektori kokoa $1 \times P^2 \times C$. Näin jokainen pala muutetaan yksiulotteiseksi vektoriksi, jota voidaan käyttää mallin syötteenä.

3. Paikkatietojen lisääminen (engl. adding position tokens): Kuvapalojen upotuksiin (engl. patch embeddings) lisätään paikkaupotukset (engl. position embeddings), jotta säilytetään sijaintitieto. Lisäksi lisätään erityinen luokittelutunniste (engl. CLS-token) paikkakoodauksen yhteyteen. Paikkatiedolla varustetut vektorimuotoiset kuvapalat syötetään Transformer-enkooderille (engl. transformer encoder).

4. Enkooderikerros (engl. encoder layer): Transformer-enkooderi koostuu vuorottelevista kerroksista, joissa on monipäinen itsehuomio (engl. MSA, multi-head self-attention) ja monikerroksinen perseptroni (engl. MLP, multilayer perceptron). Kerrosnormalisointi (engl. layer normalization) tehdään ennen sekä itsehuomiolohkoa että MLP-lohkoa. Residuaaliset yhteydet (engl. residual connections) lisätään kummankin lohkon jälkeen oppimisen helpottamiseksi.

5. Luokittelukerros (engl. classification layer): Luokittelupää (engl. classification

head) toteutetaan MLP:llä, jossa on piilokerros piirteiden käsittelyä varten sekä yksi lineaarinen kerros, joka viimeistelee kuvien luokittelun. Tämä kerros ottaa syötteenä enkooderin tuottaman lopullisen kuvan esityksen ja tuottaa luokitus pisteet kaikille mahdollisille luokille. [14]



Kuva 4.2: Kuvassa esitellään kuvamuuntimen piirteet.

4.8 EfficientNet

EfficientNet esiteltiin vuonna 2019, tämä oli merkittävä harppaus tehokkaiden ja skaalautuvien konvoluutioneuroverkkojen kehityksessä. Mallin kehittivät Mingxing Tan ja Quoc Le Googlen Brain -tutkimusryhmästä. EfficientNetin peruslähtökohtana oli ratkaista, miten konvoluutioneuroverkkoja voidaan skaalata optimaalisesti eri käyttötilanteisiin, niin suuriin palvelinmalleihin kuin pienitehoisiin mobiililaitteisiin. [16]

Ennen EfficientNetia verkkoja oli skaalattu manuaalisesti lisäämällä joko syvyyttä, leveyttä tai tulokuvan resoluutiota. EfficientNet esitteli yhdisteen skaalaus (engl.

compound scaling) nimisen menetelmän, jossa kaikkia kolmea ulottuvuutta skaalataan yhtenäisesti ja tasapainoisesti. Tämä johti edeltäjiään ResNetia tai DenseNetia tehokkaampiin malleihin, jotka saavuttivat huipputarkkuuden huomattavasti pienemmällä parametrimäärällä kuin aiemmat. [16]

EfficientNet-malleja julkaistiin useita versioita (B0–B7), jotka kasvavat asteittain niiden koossa ja tarkkuudessa. Mallin perusarkkitehtuuri perustuu MobileNetV2:sta tuttuun käänteispullonkaularakenteeseen (engl. inverted bottleneck), jonka avulla verkko ensin laajentaa piirreavaruuden ja kutistaa sen sitten takaisin. Lisäksi malli hyödynsi Swish-aktivaatiofunktiota sekä litistys- ja jännityslohkoja (engl. squeeze and excitation), jotka paransivat piirteiden painotusta dynaamisesti. [16]

4.9 ConvNext

Vuonna 2022 esiteltiin ConvNeXt, jonka taustalla oli kunnianhimoinen tavoite päivittää perinteinen konvoluutio-neuroverkko vastaamaan uutta teknologiaa neuroverkoissa. ConvNextin pyrki saavuttamaan kuvamuuntimen tason sekä tarkkuudessa, että arkkitehtuurillisessa modernisuudessa. Mallin kehittivät Zhuang Liu ja kollegat, jotka lähtivät liikkeelle ResNet-arkkitehtuurista ja sovelsivat siihen useita viime vuosien tutkimuksista peräisin olevia parannuksia. Tuloksena syntyi konvoluutio-neuroverkkomalli, joka säilyttää konvoluutiopohjaisen rakenteen, mutta hyödyntää nykyaikaisia rakennusperiaatteita. [17]

Yksi keskeinen muutos ConvNeXtissä on kerrosnormalisoinnin (engl. layer normalization) käyttö perinteisen eränormalisoinnin sijaan. Tämä parantaa verkon toimintaa erityisesti pienissä erissä ja tuo sen lähemmäksi ViT-mallien rakennetta. Tämän lisäksi ConvNeXt hyödyntää suuria 7×7 -konvoluutioita, jotka laajentavat vastaanottokenttää tehokkaasti ilman lisäkerroksia, sekä syvyydessä erotettavissa konvoluutio ratkaisuja, jotka pienentävät laskennallista kustannusta säilyttäen samalla tarkkuuden. [17]

ConvNeXtissä on myös hyödynnetty käänteispullonkaularakennetta. Tätä rakennetta hyödynnettiin myös EfficientNetissä. Näiden teknisten muutosten ansiosta ConvNeXt-mallit ovat suorituskyvyltään täysin kilpailukykyisiä ViT-pohjaisten mallien kanssa ImageNetissä, mutta samalla säilyttävät konvoluutioneuroverkkoarkkitehtuurin edut, kuten tehokkaan koulutuksen ja paremman mobiilisoveltuvuuden. [17]

ConvNeXt edustaa paluuta konvoluutiopohjaisen lähestymistavan ydintä modernisoituna ja optimoituna 2020-luvun laskenta- ja datavaatimuksiin. Se osoittaa, että konvoluutioverkot eivät ole vanhentuneita, vaan yhä erittäin kilpailukykyisiä, kunhan arkkitehtuuri päivitetään uusimman tutkimustiedon mukaiseksi. [17]

4.10 Käyttötarkoituksia

Vuosien aikana konvoluutioneuroverkkoja on hyödynnetty monissa eri käyttötarkoituksissa. Konenäön parissa konvoluutioneuroverkoilla on ollut monia eri käyttötarkoituksia. Lääketieteessä on hyödynnetty kuvien käsittelyä esimerkiksi Parkinsonin sairauden tunnistamisessa [18]. Konvoluutioneuroverkkoja on hyödynnetty myös muodin alalla [20], teollisuudessa [21] ja aakkosten tulkinnessa [27].

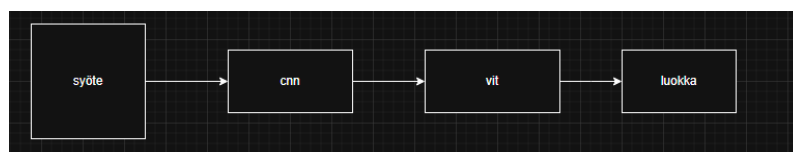
Konvoluutioneuroverkoilla on myös paljon käyttötarkoituksia sulautetuissa järjestelmissä, sillä konvoluutioneuroverkot käyttävät parametreja tehokkaasti, mikä saa ne viemään vähemmän tilaa. Konvoluutioneuroverkkoja voidaan myös pyörittää lokaalisti, mikä on optimaalista sulautetuille järjestelmille. Tämä on mahdollista, koska konvoluutioneuroverkot yleisesti vaativat vähemmän tehoa kuin ViT-mallit. Konvoluutioneuroverkkoja on hyödynnetty sulautetuissa järjestelmissä esimerkiksi terveysteknologian saralla kannettavissa terveystietojärjestelmissä [19].

5 Konvoluutioneuroverkkojen tulevaisuus

Onko konvoluutioneuroverkoilla tulevaisuutta ViT-mallien aikakaudella? Kyllä on. Vaikka ViT-mallit ovat päässeet valokeilaan viimeisinä vuosina, konvoluutioneuroverkot ovat yhä hyödyllisiä monissa käyttötarkoituksissa. Kuten edellisessä luvussa käytiin läpi, konvoluutioneuroverkoilla on melkein loputtomasti erilaisia käyttötarkoituksia. Tiivistetysti konvoluutioneuroverkkoja voidaan hyödyntää esimerkiksi muodissa, lääketieteessä, ja reaaliaikaisessa esineiden ja kuvien tunnistuksessa.

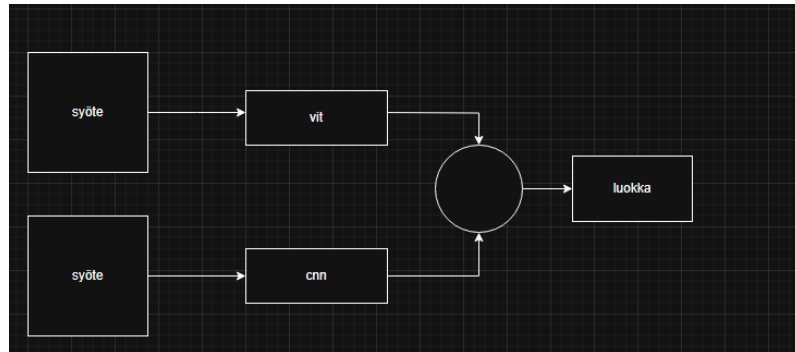
Konvoluutioneuroverkkojen tulevaisuutta ei välttämättä ole olemassa vain pelkänä itsenäisenä yksittäisenä arkkitehtuurina. Viimeisten vuosien aikana on kehitetty hybridiarkkitehtuurimalleja transformereiden kanssa. Näille arkkitehtuureille on kehitetty useista eri muotoja. Yleisimmät arkkitehtuuriyhdistelmät on kuvattu alla [14].

Peräkkäisessä yhdistelyssä konvoluutioneuroverkko poimii ensiksi piirteet syöteestä ja ViT käsittelee niitä globaaleina kokonaisuuksina. Rakenne voidaan nähdä kuvasta 5.1.



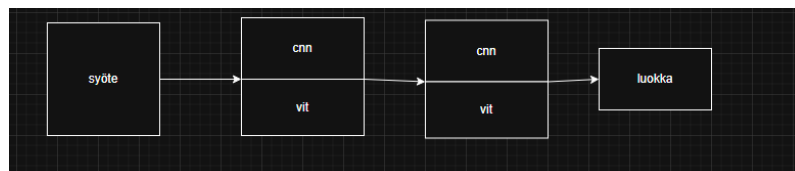
Kuva 5.1: Visuaalinen esitys peräkkäisestä yhdistelystä.

Rinnakkaisessa yhdistelyssä konvoluutioneuroverkko ja ViT toimivat samanaikaisesti ja näiden prosessien tulokset yhdistetään lopuksi yhteen. Rakenne voidaan nähdä kuvasta 5.2.



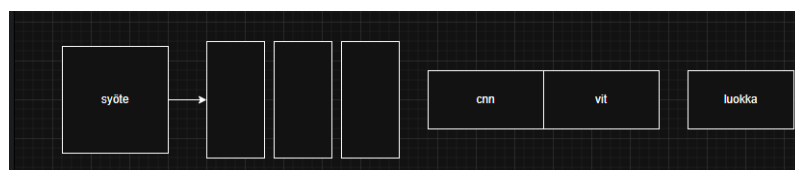
Kuva 5.2: Visuaalinen esitys rinnakkaisesta yhdistelystä.

Hierarkisessa yhdistelyssä konvoluutioneuroverkko käsittelee matalan tason piirteitä ja ViT keskittyy korkeampaan ymmärrykseen. Rakenne voidaan nähdä kuvasta 5.3.

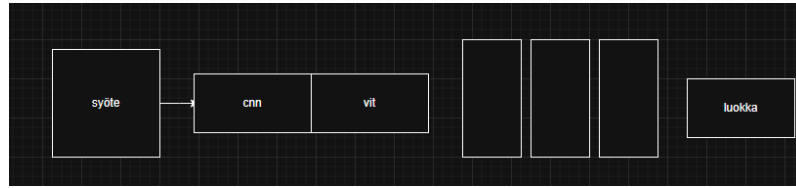


Kuva 5.3: Visuaalinen esitys hierarkisesta yhdistelystä.

Fuusioyhdistelyssä CNN- ja ViT-piirteet yhdistetään joko aikaisessa vaiheessa eli prosessin alkuprosessoinnissa tai myöhäisessä vaiheessa eli kummankin prosessin tulokset yhdistetään loppuprosessoinnissa. Kuvasta 5.4 näkee, miten alkupuolen fuusio toimii ja kuvastaja 5.5 näkee miten loppufuusio toimii.



Kuva 5.4: Kuvassa on alkuvaiheessa tapahtuva fuusio.



Kuva 5.5: Kuvassa on loppuvaiheessa tapahtuva fuusio.

Hybridimallien on todistettu toimivan paremmin kuin pelkät konvoluutioneuroverkot tai ViT omina rakenteina. [15], [14]

ConvNextistä ja EfficientNetin kautta on huomattu konvoluutioarkkiteuurissa olevan kilpailuasetelma, sillä ne pystyvät toimimaan joko yhtä hyvin tai paremmin tietyissä tilanteissa kuin ViT-mallit [17], [16]. Tutkimuksen myötä konvoluutioneuroverkot voivat kehittyä vielä toimimaan tehokkaammin tietyissä tehtävissä kuin ViT-mallit. Taulukosta 5.1 on nähtävissä, millaisissa tilanteissa yleensä kannattaa hyödyntää hybridi-, konvoluutioneuroverkko- tai ViT-mallia.

Konvoluutioneuroverkkoarkkitehtuuri toimii paljon paremmin kuin transformetit monissa erityisalueissa. Näitä tilanteita ovat esimerkiksi tilanteet, joissa data on rajattua ja joissa tarvitaan reaaliaikaista toimintaa, kuten sulatetuissa järjestelmissä. Hyödyllisyys korostuu myös tilanteissa, joissa tarvitaan selitettävyyttä tai laitteisto tarvitsee tekoälyä, kuten puhelimet. Näissä tarkoituksissa saatetaan hyödyntää vanhempia arkkitehtuureja kuten LeNettiä sen vaatiman pienemmän muisti- ja prosessointitehon takia. Tutkielman perusteella konvoluutioneuroverkkojen hyödyllisyys korostuu erityisesti edellä mainituissa käyttökohteissa.

| Kriteeri | CNN (konvoluutio- neuroverkko) | ViT (Vision Trans- former) | Hybridi (CNN + ViT) |
|---------------------------|--|--|--|
| Datan määrä | Pieni tai keskisuuri aineisto | Suuret aineistot | Keskisuuret tai suuret aineistot |
| Laskentaresurssit | Tehokas, vähäinen tai kohtalainen laskenta | Suuri laskenta, vaatii paljon muistia ja GPU-tehoa | Kohtalainen tai suuri laskenta |
| Induktiivinen oletus | Vahva (paikallisuus, siirtokesto) | Heikko (oppii kaiken datasta) | Tasapainoinen |
| Datan tarve koulutuksessa | Vähäinen, yleistyy pienellä datalla | Suuri, vaatii paljon dataa ylisovittamisen välttelemiseksi | Kohtalainen, CNN auttaa vähentämään datatarvetta |
| Tarkkuus yksityiskohdissa | Hyvä | Erinomainen kun dataa on riittävästi | Usein paras molempien yhdistelmä |
| Tulkittavuus | Helppo: suodattimet ja piirteet selkeitä | Vaikeampi: huomioalueet monimutkaisia | Haastava |
| Kestävyys muunnoksissa | Korkea | Matala ilman erityiskoulutusta | Paranee CNN-taustalla |
| Siirtopiminen | Vahva: paljon esikoulutettuja malleja | Yleistyy, mutta vaatii usein suurta esikoulutusta | Hyvä, jos molemmat esikoulutettu |
| Viive (inferenssinopeus) | Matala | Kohtalainen tai korkea | Riippuu toteutuksesta |

Taulukko 5.1: CNN, ViT, Hybridi mallien vertailu

6 Yhteenveto

Tutkielmassa käytiin läpi konvoluutioneuroverkkojen perusterminologia, sen historiallinen kehitys kahdeksan eri mallin kautta, konvoluutioneuroverkkojen käyttökohteet sekä sen tulevaisuuteen liittyviä näkökulmia. Tutkielma käsitteli kolmea tutkimuskysymystä sen aikana:

- **Miten konvoluutioneuroverkot ovat kehittyneet vuosien aikana?** Konvoluutioneuroverkkojen kehitys on ollut hyvin nopeaa viimeisen kymmenen vuoden aikana. Konvoluutioneuroverkkojen kehitys räjähti 2010-luvulla, jonka jälkeen neuroverkkojen tutkimus on jatkunut useiden eri tahojen toimesta. Neuroverkkojen arkkitehtuuri on kehittynyt useiden eri vaiheiden kautta kohti monimutkaisempaa arkkitehtuuria yhä säilyttäen konvoluutioneuroverkkojen peruseräpäät. Arkkitehtuurin osalta kehitys on siirtynyt modernisoimaan vanhempia neuroverkkomalleja uusilla tekniikoilla ja kehittämään hybridiarkkitehtuureja ViT-mallien kanssa.
- **Mihin konvoluutioneuroverkkomalleja käytetään konenäön kannalta?** Tutkielman kautta on käynyt ilmi, että konvoluutioneuroverkoilla on paljon käyttötarkoituksia kuvan käsittelyn piirissä erityisesti asioiden tunnistamisessa ja kuvien analysoinnissa. Tämän lisäksi soveltamisalueita on useita esimerkiksi terveydenhuollosta muotiin.
- **Onko konvoluutioneuroverkoilla tulevaisuutta?** Kyllä on, sillä konvoluutioneuroverkkojen kehityksen kautta on tultu pisteeseen, jossa niiden merkitys

on edelleen suuri kuvankäsittelyn parissa vaikkakin asemasta on vahvaa kilpailua kuvamuuntimien kanssa. Konvoluutioneuroverkot toimivat paremmin kuin kuvamuuntimet tietyissä rajoitetuissa ympäristöissä, joissa ei voida hyödyntää suurta prosessivoimaa tai käytettävissä olevan koulutusdatan määrä on pieni. Moderni konvoluutioneuroverkkomalli, kuten ConvNext, pystyy parempiin tarkkuuksiin kuin ViT-mallit normaaleissa konenäkötehtävissä. Tämä tarkoittaa sitä, että konvoluutioneuroverkoilla on vieläkin hyvä mahdollisuus kehittyä. Konvoluutioneuroverkoilla on myös potentiaalia olla osana hybridiarkkitehtuuria ViT-mallien kanssa, sillä nykytutkimuksen mukaan näiden mallien suorituskyky on parempi kuin yksittäisten konvoluutioneuroverkkomallien tai ViT-mallien.

Seuraavina vuosina lisätutkimuksessa tulisi keskittyä erityisesti hybridimalleihin yhdessä kuvamuuntimien kanssa sekä tehokkaamman arkkitehtuurin kehittämiseen. Tutkimuksen tulisi myös keskittyä myös uusien konvoluutioneuroverkkomallien kehittämiseen ja uusien arkkitehtuurien testaamiseen. Tutkielman valossa konvoluutioneuroverkkojen tulevaisuus konenäössä on valoisa ja täynnä kehittämis- ja soveltamismahdollisuuksia.

Lähdeluettelo

- [1] D. H. Hubel ja T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex", *The Journal of Physiology*, vol. 148, nro 3, s. 574–591, lokakuu 1959, ISSN: 1469-7793. DOI: 10.1113/jphysiol.1959.sp006308. url: <http://dx.doi.org/10.1113/jphysiol.1959.sp006308>.
- [2] K. Fukushima, "Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements", *IEEE Transactions on Systems Science and Cybernetics*, vol. 5, nro 4, s. 322–333, 1969, ISSN: 0536-1567. DOI: 10.1109/tssc.1969.300225. url: <http://dx.doi.org/10.1109/TSSC.1969.300225>.
- [3] K. O. ja Ryan Nash, "An Introduction to Convolutional Neural Networks", joulukuu 2015, Accessed: 2024-10-24. url: <https://doi.org/10.48550/arXiv.1511.08458>.
- [4] S. A. e. a. Khan A., "A survey of the recent architectures of deep convolutional neural networks", huhtikuu 2020. url: <https://doi.org/10.1007/s10462-020-09825-6>.
- [5] N. E. e. a. Tabark A, "A Survey Study of the Deep Learning for Convolutional Neural Network Architecture", huhtikuu 2024. url: <https://doi.org/10.1051/bioconf/20249700083>.
- [6] B. Y. j. H. LeCun Y. Bottou L., "Gradient-based learning applied to document recognition", 1989. url: <https://doi.org/10.1109/5.726791>.

- [7] A. Krizhevsky ja G. E. Sutskever Ilya ja Hinton, ”ImageNet classification with deep convolutional neural networks”, *Commun. ACM*, vol. 60, nro 6, s. 84–90, toukokuu 2017, ISSN: 0001-0782. DOI: 10.1145/3065386. url: <https://doi-org.ezproxy.utu.fi/10.1145/3065386>.
- [8] I. S. ja Gulshan Goyal ja A. Chandel, ”AlexNet architecture based convolutional neural network for toxic comments classification”, *Journal of King Saud University - Computer and Information Sciences*, vol. 34, nro 9, s. 7547–7558, 2022, ISSN: 1319-1578. DOI: <https://doi.org/10.1016/j.jksuci.2022.06.007>. url: <https://www.sciencedirect.com/science/article/pii/S1319157822002026>.
- [9] M. D. Zeiler ja R. Fergus, ”Visualizing and Understanding Convolutional Networks”, teoksessa *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, s. 818–833, ISBN: 9783319105901. DOI: 10.1007/978-3-319-10590-1_53. url: http://dx.doi.org/10.1007/978-3-319-10590-1_53.
- [10] K. Simonyan ja A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, 2014. DOI: 10.48550/ARXIV.1409.1556. url: <https://arxiv.org/abs/1409.1556>.
- [11] C. Szegedy et al., *Going Deeper with Convolutions*, 2014. DOI: 10.48550/ARXIV.1409.4842. url: <https://arxiv.org/abs/1409.4842>.
- [12] K. He, X. Zhang, S. Ren ja J. Sun, ”Deep Residual Learning for Image Recognition”, teoksessa *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, kesäkuu 2016. DOI: 10.1109/cvpr.2016.90. url: <http://dx.doi.org/10.1109/CVPR.2016.90>.
- [13] G. Huang, Z. Liu, L. van der Maaten ja K. Q. Weinberger, *Densely Connected Convolutional Networks*, 2016. DOI: 10.48550/ARXIV.1608.06993. url: <https://arxiv.org/abs/1608.06993>.

- [14] H. Yunusa, S. Qin, A. H. A. Chukkol, A. A. Yusuf, I. Bello ja A. Lawan, *Exploring the Synergies of Hybrid CNNs and ViTs Architectures for Computer Vision: A survey*, 2024. DOI: 10.48550/ARXIV.2402.02941. url: <https://arxiv.org/abs/2402.02941>.
- [15] S. Bbouzidi, G. Hcini, I. Jdey ja F. Drira, *Convolutional Neural Networks and Vision Transformers for Fashion MNIST Classification: A Literature Review*, 2024. DOI: 10.48550/ARXIV.2406.03478. url: <https://arxiv.org/abs/2406.03478>.
- [16] M. Tan ja Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", 2019. DOI: 10.48550/ARXIV.1905.11946. url: <https://arxiv.org/abs/1905.11946>.
- [17] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell ja S. Xie, *A ConvNet for the 2020s*, 2022. DOI: 10.48550/ARXIV.2201.03545. url: <https://arxiv.org/abs/2201.03545>.
- [18] Y. Dai et al., "Multi-Focus Image Fusion Based on Convolution Neural Network for Parkinson's Disease Image Classification", *Diagnostics*, vol. 11, nro 12, s. 2379, joulukuu 2021, ISSN: 2075-4418. DOI: 10.3390/diagnostics11122379. url: <http://dx.doi.org/10.3390/diagnostics11122379>.
- [19] T. Y. Wang, J. Cui ja Y. Fan, "A wearable-based sports health monitoring system using CNN and LSTM with self-attentions", *PLOS ONE*, vol. 18, nro 10, S. Veerappampalayam Easwaramoorthy, toim., e0292012, lokakuu 2023, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0292012. url: <http://dx.doi.org/10.1371/journal.pone.0292012>.
- [20] M. S. Saranya ja P. Geetha, "Fashion Image Classification Using Deep Convolution Neural Network", teoksessa *Computer, Communication, and Signal Processing*. Springer International Publishing, 2022, s. 116–127, ISBN: 9783031116339.

- DOI: 10.1007/978-3-031-11633-9_10. url: http://dx.doi.org/10.1007/978-3-031-11633-9_10.
- [21] J. Wang, W. Zhang, H. Wu ja J. Zhou, "Improved bilayer convolution transfer learning neural network for industrial fault detection", *The Canadian Journal of Chemical Engineering*, vol. 100, nro 8, s. 1814–1825, elokuu 2021, ISSN: 1939-019X. DOI: 10.1002/cjce.24281. url: <http://dx.doi.org/10.1002/cjce.24281>.
- [22] P. L. j. C. S. Mohamed Chetouani Virginia Dignum, "Human-Centered Artificial Intelligence", huhtikuu 2023, chapter Artificial Neural Networks. url: <https://doi.org/10.1007/978-3-031-24349-3>.
- [23] G. I. ja Bengio Yoshua ja Courville Aaron, *Deep Learning*. MIT Press, 2016. url: <https://www.deeplearningbook.org/>.
- [24] T. Szandała, *Bio-inspired Neurocomputing*. Springer Singapore, 2021, ISBN: 9789811554957. DOI: 10.1007/978-981-15-5495-7. url: <http://dx.doi.org/10.1007/978-981-15-5495-7>.
- [25] Y. j. H. G. LeCun Y.ja Bengio, "Deep learning", *Nature*, vol. 521, nro 7553, s. 436–444, 2015. DOI: <https://doi.org/10.1038/nature14539>.
- [26] S. Nitish, H. Geoffrey, K. Alex, S. Ilya ja S. Ruslan, "Dropout: a simple way to prevent neural networks from overfitting", vol. 15, nro 1, s. 1929–1958, tammikuu 2014, ISSN: 1532-4435.
- [27] J. Ravi, "Handwritten alphabet classification in Tamil language using convolution neural network", *International Journal of Cognitive Computing in Engineering*, vol. 5, s. 132–139, 2024, ISSN: 2666-3074. DOI: 10.1016/j.ijcce.2024.03.001. url: <http://dx.doi.org/10.1016/j.ijcce.2024.03.001>.