

Joni Aalto

# VÄKIVALTAISEN KÄYTTÄYTYMISEN TUNNISTAMINEN VIDEOANALYTII- KALLA

Diplomityö  
Informaatioteknologian ja viestinnän tiedekunta  
Joni Kämäräinen  
Esa Rahtu  
Huhtikuu 2022

# TIIVISTELMÄ

Joni Aalto: Väkivaltaisen käyttäytymisen tunnistaminen videoanalytiikalla  
Diplomityö  
Tampereen yliopisto  
Tietotekniikan DI-tutkinto-ohjelma  
Huhtikuu 2022

---

Videoanalytiikkaa voidaan hyödyntää muun muassa automatisoimaan väkivaltaisuuden tunnistamista videovalvontakameroista ja näin parantaa kaupunkiturvallisuutta tunnistamalla väkivaltata-pauksia. Tämän diplomityön tavoitteena oli tutkia, minkä tyyppinen koneoppimismalli soveltuisi tunnistamaan väkivaltaa videoista. Koneoppimismalliksi valikoitui edeltävien tutkimusten perusteella konvoluutioneuroverkko. Diplomityön tietoaineistona käytettiin UBI Fights tietoaineistoa, joka koostuu videoista, jotka sisältävät väkivaltaisia ja ei-väkivaltaisia tilanteita kaupunkikatukuvasta sekä sisätiloista. Tämän diplomityön koneoppimismallin toteuttaminen aloitettiin esiprosessoimalla käytettävää tietoaineistoa. Esiprosessoinnissa tietoaineiston videokehysten kuvakokoa muutettiin ja värikuvat muutettiin harmaakuviksi. Koneoppimismallia opetettiin käyttämällä ohjattua oppimista, jossa tietoaineiston videot on merkattu oikeiden luokkien perusteella. Koneoppimismallia varten tietoaineisto on jaettu opetus, validointi ja- testausaineistoksi. Tässä diplomityössä koneoppimismallin tarkkuudeksi saatiin opetusvaiheessa 92,1 %. Testiaineistolla koneoppimismallia testattaessa kokonaistarkkuudeksi saatiin 72,2 %. Koneoppimismallin raja-arvoa muuttamalla kokonaistarkkuus parani 78,3 %:in.

Avainsanat: Koneoppiminen, Videoanalytiikka, Konvoluutioneuroverkko

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

# ABSTRACT

Joni Aalto: Violent behaviour detection using video analytics  
Master's Thesis  
Tampere University  
Degree Programme in Information Technology, MSc (Tech)  
Master of Science  
April 2022

---

Video analytics can be used for example to automatize violence detection from video surveillance stream. This can improve urban security by detecting violent behaviour. The aim of this thesis was to implement machine learning model that detects violent actions from videos. Based on the previous studies, convolutional neural network was selected as the machine learning model for this thesis. In this thesis, UBI fights dataset was used. UBI fights dataset contains videos which include violent and non-violent actions from street view or indoors. Implementation of the machine learning model was started by preprocessing dataset videos. In the preprocessing step, the size of the video frames was modified, and colored frames were transformed to grayscale frames. Machine learning model was trained using supervised learning method and labeled dataset videos. For training purpose dataset was split into three different sets: training, validation, and test datasets. In this thesis, machine learning model training accuracy was 92.1%. With the test dataset, the accuracy of the machine learning model was 72.2%. By changing the threshold value of the machine learning model, the accuracy was increased to 78.3%.

Keywords: Machine Learning, Video Analytics, Convolutional Neural Network

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

# ALKUSANAT

Tämä diplomityö toteutettiin Euroopan Unionin rahoittamalle SURE (Smart Urban Security and Event Resilience) hankkeelle. Haluan kiittää kaikkia SURE-hankkeessa mukana olleita henkilöitä.

Suuri kiitos työni ohjaajalle professori Joni Kämäräiselle saamastani palautteesta ja korjausehdotuksista. Häneltä sain hyödyllisiä vinkkejä työni tekemiseen.

Erityiskiitokset perheelleni ja Julia Pietilälle kaikesta saamastani tuesta diplomityön aikana.

Tampereella, 27.4.2022

Joni Aalto

# SISÄLLYSLUETTELO

1. JOHDANTO .....	1
2. EDELTÄVÄ TUTKIMUS .....	3
3. TEORIA .....	5
3.1 Koneoppiminen .....	5
3.1.1 Ohjattu oppiminen .....	6
3.1.2 Ohjaamaton oppiminen .....	6
3.1.3 Vahvistusoppiminen .....	7
3.2 Neuroverkko .....	7
3.2.1 Neuroverkon rakenne .....	8
3.2.2 Monikerroksinen neuroverkko .....	10
3.2.3 Neuroverkon opettaminen .....	11
3.2.4 Ylioppiminen .....	11
3.3 Syväoppiminen .....	13
3.4 Konvoluutioneuroverkko .....	14
3.4.1 Konvoluutioneuroverkon rakenne .....	14
3.4.2 Konvoluutiokerros .....	15
3.4.3 Yhdistämiskerros .....	16
3.4.4 Täysin kytketty kerros .....	17
4. TIETOAINEISTO .....	18
4.1 Tietoaineisto .....	18
4.2 Opetusaineiston kokoaminen .....	23
4.2.1 Opetusaineistoon valittuja videoita .....	24
4.2.2 Opetusaineistosta hylättyjä videoita .....	27
5. MENETELMÄ .....	32
5.1 Tekniikoiden valinta .....	32
5.2 Mallin toteutuksen vaiheet .....	33
5.2.1 Videoiden esiprosessointi .....	33
5.2.2 Mallin kehitys .....	33
5.3 Mallin opettaminen .....	34
5.4 Arkkitehtuuri .....	34
5.5 Mallin iterointi .....	36
6. MALLIN ARVIOINTI .....	41
6.1 Arviointimenetelmät .....	41
6.2 Iteroitujen mallien tulokset .....	42
6.3 Valittu malli .....	43
6.4 Tulokset .....	45
6.5 Tulosten vertailu aiempiin tutkimuksiin .....	47
6.6 Opetusaineiston arviointi .....	48
7. YHTEENVETO .....	49
LÄHTEET .....	50

# KUVALUETTELO

<b>Kuva 1.</b>	<i>Videoanalytiikan vaiheet [6].....</i>	<i>2</i>
<b>Kuva 2.</b>	<i>Koneoppimisen oppimistapoja [15].....</i>	<i>6</i>
<b>Kuva 3.</b>	<i>Neuronin matemaattinen rakenne [19]. ....</i>	<i>8</i>
<b>Kuva 4.</b>	<i>ReLu-aktivointifunktion kuvaaja [22].....</i>	<i>9</i>
<b>Kuva 5.</b>	<i>Sigmoid-aktivointifunktion kuvaaja [23].....</i>	<i>10</i>
<b>Kuva 6.</b>	<i>Eteenpäin suuntautuva neuroverkko [19]. ....</i>	<i>11</i>
<b>Kuva 7.</b>	<i>Hyvin oppinut (vasen) ja ylioppinut regressio (oikea) [28].....</i>	<i>12</i>
<b>Kuva 8.</b>	<i>Pudotuskerroksen toiminta opetuskierrosten aikana [30]. ....</i>	<i>13</i>
<b>Kuva 9.</b>	<i>Syväoppivan ja perinteisen koneoppimismallin ero tarkkuudessa datamäärän suhteen [32]. ....</i>	<i>14</i>
<b>Kuva 10.</b>	<i>Tyypillinen konvoluutioverkon rakenne [35]. ....</i>	<i>15</i>
<b>Kuva 11.</b>	<i>Syötekuvan ja konvoluutiokerrosten välinen yhteys [35].....</i>	<i>16</i>
<b>Kuva 12.</b>	<i>Suurimman arvon valinta [35].....</i>	<i>17</i>
<b>Kuva 13.</b>	<i>Kuvasarja ei-väkivaltaisesta käyttäytymisestä sisätiloissa [37]. ....</i>	<i>19</i>
<b>Kuva 14.</b>	<i>Kuvasarja ei-väkivaltaisesta käyttäytymisestä ulkona [37].....</i>	<i>20</i>
<b>Kuva 15.</b>	<i>Kuvasarja kahden henkilön väkivaltaisesta käyttäytymisestä [37]. ....</i>	<i>21</i>
<b>Kuva 16.</b>	<i>Kuvasarja väkivaltaisesta joukkotappelusta [37].....</i>	<i>22</i>
<b>Kuva 17.</b>	<i>Kuvasarja ei-väkivaltaisesta käytöksestä sisätiloissa [37]. ....</i>	<i>25</i>
<b>Kuva 18.</b>	<i>Kuvasarja ei-väkivaltaisesta käytöksestä ulkotiloissa [37]. ....</i>	<i>26</i>
<b>Kuva 19.</b>	<i>Esimerkki kuvakehys valitusta videosta [37].....</i>	<i>27</i>
<b>Kuva 20.</b>	<i>Kuvasarja hylätystä peittävästä videosta [37].....</i>	<i>29</i>
	<i>30</i>	
<b>Kuva 21.</b>	<i>Kuvasarja hylätystä kaukaa kuvatusta videosta [37]. ....</i>	<i>30</i>
<b>Kuva 22.</b>	<i>Kuvasarja hylätystä epäselvästä videosta [37]. ....</i>	<i>31</i>
<b>Kuva 23.</b>	<i>Konvoluutioneuroverkon arkkitehtuuri. ....</i>	<i>36</i>
<b>Kuva 24.</b>	<i>Lopullisen mallin rakenne.....</i>	<i>44</i>

# LYHENTEET JA MERKINNÄT

LSTM	Long Short-Term Memory
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
ReLu	Rectified Linear Unit
UBI	University of Beira Interior

# 1. JOHDANTO

Isoissa kaupungeissa on käytössä suuria määriä videovalvontakameroita, joilla valvotaan kaupunkien yleistä turvallisuutta. Videokamerat keräävät huomattavan määrän dataa ollessaan päällä ympäri vuoden, mutta vain pieni osa kerätystä datasta sisältää turvallisuutta uhkaavaa käyttäytymistä. Perinteisesti videokameroiden kuvaamaa materiaalia käydään läpi ihmisen toimesta, jotta voidaan valvoa ja havaita turvallisuuspoikkeamia. Videokameroiden monitoreita valvovalle henkilölle on haastavaa, jos valvottavana on useampia kameroita. Voorthuijsen tutki vuonna 2005 ihmisen kykyä valvoa useampaa kameraa samanaikaisesti [1]. Keskimäärin ihmisen kyvykyys tarkkailla useampaa kameraa laski 19 %, kun samanaikaisesti tarkkailtavien kameroiden määrää kasvatettiin yhdestä neljään [1].

Nykypäivän videovalvontakameroissa onkin alettu käyttää tekoälyä useamman videovalvontakameran samanaikaisessa seurannassa, sillä tekoälyn käyttö helpottaa poikkeavien tilanteiden havaitsemista. Älykkäät valvontajärjestelmät (Intelligence Surveillance Systems) muodostavat automaattisesti hälytyksiä ylläpitäjille tai kolmannen osapuolen järjestelmille havaittaessa epänormaalia käyttäytymistä tai normaalista poikkeavia tapahtumia [2]. Tekoälyä hyödyntävän videoanalytiikan tulojen on arvioitu kasvavan vuoden 2018 1,1 miljardista dollarista vuoteen 2025 mennessä 4,5 miljardiin dollariin [3].

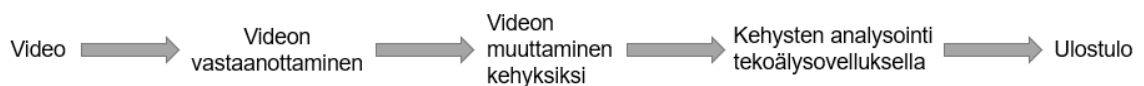
Videoanalytiikka on prosessi, jossa käsitellään videokuvamateriaalia koneoppimisen mallien avulla pyrkien löytämään datasta niiden välisiä yhteyksiä ja piilossa olevia säännönmukaisuuksia [4]. Löydettyjen säännönmukaisuuksien ansiosta videoanalytiikalla voidaan pystyä jopa ennustamaan tulevia tapahtumia ja käyttäytymistä, ja lisäksi se antaa tukea päätöksenteossa epävarmoissa tilanteissa [4]. Videoanalytiikkaa hyödynnetään muun muassa kaupunkien videovalvontakameroissa analysoimalla suuria määriä niiden tuottamaa dataa ja nostamalla niissä tapahtuneet poikkeamat tai muut kiinnostavat seikat ihmisen tarkkailtavaksi. Kaupunkien videovalvonnassa videoanalytiikka voidaan jakaa karkeasti kahteen eri tapaan: tapahtuman tunnistamiseen ja poikkeamien havaitsemiseen [5]. Tapahtumien tunnistamisessa keskitytään tunnistamaan ihmisten



erilaisia eleitä ja toimintoja, kun taas poikkeamien havaitsemisessa keskitytään tunnistamaan epätavallisia tai harvinaisia tapahtumia [5].

Videoanalytiikan avulla voidaan esimerkiksi vähentää väärin hälytysten määrää ja seurata automaattisesti epäilyttäviä tai normaalista poikkeavien henkilöiden käyttäytymistä. Reaaliaikainen videoanalytiikka myös parantaa järjestelmien suorituskykyä, sillä videokameroiden kuvaamaa materiaalia ei enää tarvitse varastoida pitkäksi aikaa tai siirtää toisaalle pitempiaikaista varastointia varten. Videoanalytiikan avulla videomateriaalista voidaan myös esimerkiksi lähettää kiinnostavat osat suoraan viranomaisille, jotka voivat niiden pohjalta tehdä päätöksiä [2]. Katukuvassa tapahtuvien väkivaltatilanteiden tunnistaminen videoanalytiikkaa hyödyntävien kaupunkikameroiden avulla voisi parantaa kaupunkiturvallisuutta. Aikaisessa vaiheessa tunnistettavia väkivaltatilanteita voidaan mahdollisesti pystyä estämään, jos niihin pystytään reagoimaan tarpeeksi nopeasti.

Videovalvonnassa videoanalytiikka-ketju koostuu useammasta vaiheesta, kuten esimerkiksi kuvassa 1 on esitetty [6]. Ensin vastaanotetaan video joko videovalvontakamerasta tai muusta lähteestä. Vastaanotettu video puretaan yksittäisiksi videokehyksiksi, joita on helpompi käsitellä. Videokehyksiä esikäsitellään halutuilla menetelmillä, minkä jälkeen ne syötetään koneoppimismallille. Koneoppimismalli antaa ulostuloksi luokitustuloksen.



**Kuva 1.** Videoanalytiikan vaiheet. [6]

Tämän diplomityön tutkimuskysymys on, miten videokuvasta voidaan tunnistaa ihmisten turvallisuutta uhkaavaa käyttäytymistä. Tässä diplomityössä keskitytään tunnistamaan ihmisten väkivaltaista käyttäytymistä kaupunkikatukuvasta hyödyntäen koneoppimismallia. Työssä käydään läpi aiheeseen liittyvää teoriaa ja tutkitaan edeltäviä tutkimuksia samasta aiheesta. Teorian ja edeltävien tutkimusten pohjalta toteutetaan koneoppimismalli, jonka tavoitteena on tunnistaa ihmisten väkivaltaista ja ei-väkivaltaista käyttäytymistä sille syötetyistä videoista. Tämän diplomityön tavoite on tutkia, millainen koneoppimismalli soveltuisi parhaiten tunnistamaan väkivaltaa videoista ja kehittää tähän käytötarkoitukseen soveltuva koneoppimismalli.

## 2. EDELTÄVÄ TUTKIMUS

Väkivaltaisuuden tunnistaminen videoista tekoälyn avulla on laajasti tutkittu alue. Aiemmin julkaistuissa tutkimuksissa on tyypillisesti tutkittu ja selvitetty sitä, millainen malli tunnistaa tehokkaammin väkivaltaa erilaisissatilanteissa. Näistä tutkimuksista muutamia esitellään tässä luvussa. Tutkimuksista käytiin läpi käytetyt menetelmät ja tietoaaineistot, joilla oli tutkittu väkivaltaisuuden tunnistamista videoista. Lisäksi tarkastellaan, millaisia tuloksia tutkimuksissa oli saatu käytetyillä menetelmillä. Tässä työssä läpikäytyissä tutkimuksissa malleja oli opetettu samoilla tietoaaineistoilla, joten tutkimusten vertailu keskenään oli helpompaa.

Hanson kollegoineen [7] tutki vuonna 2018, miten kaksisuuntainen LSTM-konvoluutioneuroverkko soveltui väkivaltaisuuden tunnistamiseen videoista. He lähestyivät ongelmaa käyttämällä valmiiksi opetettua VGG13-konvoluutioneuroverkkoa piirteiden keräämiseen. Kerätyt piirteet he siirsivät kansisuuntaiselle LSTM-kerrokselle. Tämän jälkeen he siirsivät LSTM-kerroksen ulostulon luokittelijalle, jossa he toteuttivat varsinaisen luokittelun videon väkivaltaisesta sisällöstä. Näin muodostettua mallia he opettivat ja testasivat käyttämällä kolmea tietoaaineistoa, joita on yleisesti käytetty muissakin tutkimuksissa, joissa pyritään tunnistamaan väkivaltaisuutta videoista. Heidän käyttämät tietoaaineistot olivat Jääkiekko-aineisto (Hockey Fights dataset) [8], Elokuva-aineisto (Movie dataset) [8] ja Joukkoväkivalta-aineisto (Violent Flows dataset) [9]. Hanson kollegoineen [7] saivat Jääkiekko-aineistolla parhaan mallinsa tarkkuudeksi  $96,96\% \pm 1,08\%$ , Elokuva-aineistolla  $100\% \pm 0\%$  ja Joukkoväkivalta-aineistolla  $92,18\% \pm 3,29\%$ .

Ullah kollegoineen [10] tutki vuonna 2019, miten 3D-konvoluutioneuroverkkoa hyödyntävä tekoälymalli soveltuu väkivaltaisuuden tunnistamiseen videoista. He lähestyivät ongelmaa samalla tavalla kuin Hanson kollegoineen [7], sillä hekin käyttivät valmiiksi opetettua konvoluutioneuroverkkoa. Ullah kollegoineen [10] käytti valmiiksi opetettua MobileNet-konvoluutioneuroverkkoa tunnistamaan ensin videokuvakehyksissä esiintyvät ihmiset. Tällä tavalla he toteuttivat luokittelua vain niille videokuvakehyksille, joissa esiintyy ihmisiä, ja saivat minimoitua analysointiin tarvittavaa tietokoneiden laskutehoa. 3D-konvoluutioneuroverkolle siirrettiin ainoastaan ne kuvakehykset, joissa oltiin havaittu ihmisiä. 3D-konvoluutioverkon avulla he keräsivät videokuvakehyksistä piirteitä, jotka he siirsivät edelleen luokittelijalle. Luokittelijan avulla he toteuttivat lopullisen luokittelun sille, sisälsikö video väkivaltaa. Mallinsa opettamiseen ja testaamiseen Ullah kollegoineen

[10] käytti samoja tietoaaineistoja kuin Hanson kollegoineen [7]. He saivat Jääkiekko-aineistolla parhaan mallinsa tarkkuudeksi 96 %, Elokuva-aineistolla 99,9 % ja Joukkoväkivalta-aineistolla 98 %.

Asad kollegoineen [11] tutkivat vuonna 2020, miten spatiaalista ja ajallista tietoa keräämällä kuvakehyksistä konvoluutioneuroverkolla voidaan tunnistaa väkivaltaa videoista. He lähestyivät ongelmaa myös samalla tavalla kuin aikaisemmissa tässä luvussa esitetyissä tutkimuksissa eli käyttivät valmiiksi opetettua konvoluutioneuroverkkoa. Suurin ero muihin esiteltyihin tutkimuksiin on se, että heidän menetelmänsä keskittyy keräämään kuvakehyksistä spatiaaliset ja ajalliset tiedot, joiden avulla voidaan paremmin tunnistaa videoissa esiintyviä lyhytkestoisia väkivaltatilanteita. Heidän toteutuksensa koostui valmiiksi opetetusta VGG16-konvoluutioneuroverkosta, jolla he keräsivät videokuva kehyksistä piirteitä LSTM-kerrokselle, josta ne siirrettiin luokittelijalle. Luokittelijassa toteutettiin lopullinen luokittelu siitä, sisälsikö video väkivaltaa. He käyttivät malliensa opettamiseen ja testaamiseen samoja tietoaaineistoja kuin tässä luvussa aiemmin esitellyt tutkimukset. Lisäksi he käyttivät myös Käytös-tietoaaineistoa (Behave dataset) [12]. He saivat parhaan mallinsa tarkkuudeksi Jääkiekko-aineistoilla 98,8 %, Elokuva-aineistoilla 99,1 %, Joukkoväkivalta-aineistolla 97,1 % ja Käytös-tietoaaineistoilla 95,9 %.

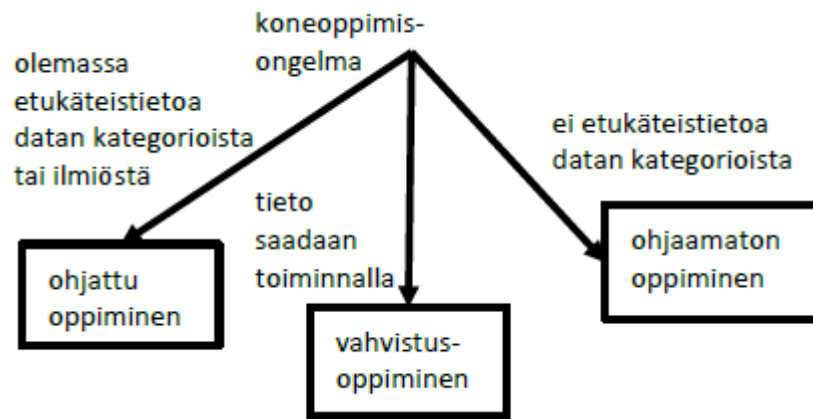
## 3. TEORIA

Teoriakappaleessa tehdään kirjallisuuskatsausta työhön liittyvästä teoriasta. Kappaleessa on tarkoitus perehdyttää lukijaa yleisellä tasolla koneoppimisen teoriaan ja sen osa-alueisiin, kuten ohjattuun oppimiseen, ohjaamattomaan oppimiseen ja vahvistusoppimiseen. Sen jälkeen käydään neuroverkkojen ja syväoppimisen perusteet läpi, minkä jälkeen tutkitaan konvoluutio neuroverkkoja hieman syvemmin.

### 3.1 Koneoppiminen

Koneoppiminen on yksi tekoälyn osa-alueista, jossa kone tai ohjelma oppii saamistaan syötteistä säännönmukaisuuksia [13]. Koneoppiminen on määritelty menetelmiksi, joilla pystytään havaitsemaan syötetiedoista niissä piileviä säännönmukaisuuksia [14]. Koneoppimismenetelmää voidaan esimerkiksi käyttää tunnistamaan tai luokittelemaan uusista tiedoista niitä säännönmukaisuuksia, joita se on oppinut [14]. Koneoppimista hyödynnetään esimerkiksi kuvan- ja puheentunnistuksen ongelmissa [14]. Kuvantunnistuksessa koneoppimismallilla voidaan syötekuvista tunnistaa esimerkiksi ihmisiä tai kasvoja [13]. Puheentunnistuksessa voidaan tunnistaa esimerkiksi äännteitä ja sanoja [13].

Koneoppiminen voidaan jakaa oppimistapojen perusteella kolmeen osa-alueeseen, joita ovat ohjattu oppiminen, ohjaamaton oppimisen sekä vahvistusoppiminen [15]. Oppimismenetelmän valintaan vaikuttaa saatavilla oleva tieto, kuten kuvassa 2 esitetään [15]. Jos syötetiedon luokat tunnetaan, käytetään ohjattua oppimista. Jos syötetiedon luokkia ei ole saatavilla, käytetään ohjaamatonta oppimista. [15] Vahvistusoppimista käytetään silloin, kun koneoppimismallin toiminnasta saadaan kerättyä tietoa [15].



Kuva 2. Koneoppimisen oppimistapoja [15].

### 3.1.1 Ohjattu oppiminen

Ohjatussa oppimisessa tavoitteena on oppia syötteen  $X$  avulla ulostulo  $Y$ . Syötteet annetaan syötepareina  $(x_i, y_i)$ , missä  $y_i \in Y$  on syöteparin ulostulo ja  $x_i \in X$  on syöteparin syötearvo [16]. Ohjatussa oppimisessa ohjelmiston opettaminen tapahtuu sille syötetyillä syötepareilla, joiden ulostulot tunnetaan [16]. Näitä syötetietoja voidaan syöttää esimerkiksi jonkun toisen ohjelmiston tai ihmisen toimesta [15]. Ohjatussa oppimisessa koneoppimismallin pitäisi tunnistaa uudet syötteet, ja luokitella ne tunnettujen luokkien perusteella [15].

Ohjatun oppimisen koneoppimisongelmat voidaan jakaa luokitteluun tai regressioon. Luokittelussa tavoitteena on luokitella saadut syötteet opettujen syöteparien avulla [14]. Luokittelussa käytetään syötepareja, joiden ulostulot ovat luokkakategorioita [14]. Luokat ovat muotoa  $y = \in \{1, \dots, C\}$ , jossa  $C$  on luokkien kokonaismäärä [14]. Luokkien määrä vaikuttaa siihen, mikä luokittelu on kyseessä [14]. Jos luokkia on vain kaksi, on kyseessä binääriluokittelusta, jossa  $y$  saa arvoja  $y \in \{0, 1\}$  [14]. Jos luokkien määrä on enemmän kuin kaksi, on kyseessä moniluokittelu [14]. Regressiossa ulostulon  $y$  arvot ovat jatkuvia [10].

### 3.1.2 Ohjaamaton oppiminen

Ohjaamattomassa oppimisessa ei muodosteta syötepareja kuten ohjatussa oppimisessa [14]. Sen sijaan ohjaamattomassa oppimisessa on tarkoitus selvittää syötteiden rakennetta sekä niissä piileviä samankaltaisuuksia [14]. Syötteiden rakennetta ja ominaisuuksia voidaan selvittää esimerkiksi havaitsemalla syötteissä esiintyviä ryhmiä klusteroinnin avulla [17]. Ohjaamattoman oppimisen käyttäminen ei vaadi syöteparien muodostamista

opetustiedoista [17]. Syöteparien muodostaminen voi olla kallis ja aikaa vievä prosessi, sillä yleensä se vaatii ihmisen muodostamaan opetustiedoista syöteparit [17].

Ohjaamaton oppiminen jaetaan klusteroinniksi ja dimensionaaliseksi pudotusmenetelmäksi [15]. Klusteroinnissa on tarkoitus löytää opetusdatasta erilaisia ryhmiä, jotka muodostuvat samankaltaisista syöteistä [17]. Klusteroinnissa jaetaan syötetiedot  $n$  ryhmisiin, että jokaisessa ryhmässä näytteet ovat keskenään samanlaisia, mutta ne eivät ole samankaltaisia muiden ryhmien kuuluvien tietojen kanssa [17]. Dimensionaalista pudotusmenetelmää käytetään, kun opetustiedon syötteet sisältävät monia dimensioita [15]. Dimensioiden määrän kasvu vaikeuttaa opetustietojen tulkintaa. Lisäksi jos syötetiedot koostuvat monista dimensioista, koneoppimismallien tarkkuus heikkenee [15] [17]. Syötteitä halutaan muuttaa selkeämmäksi dimensionaalilla pudotusmenetelmällä, jotta vältetään ”dimensionaalisuuden kirouksen” haitoilta [15].

### **3.1.3 Vahvistusoppiminen**

Vahvistusoppiminen eroaa ohjatusta oppimisesta siten, että siinä ei ole ollenkaan saatavilla opetustietoa, jonka avulla opettaminen toteutettaisiin [15]. Opetustiedon puuttumisen takia oppiminen tapahtuu vahvistusoppimisessa hitaasti, kun oppiminen perustuu yrityksistä saatavaan palautteeseen, eikä tarjottuun opetustietoon [18]. Yrityksistä saatava palaute on joko positiivista tai negatiivista, ja niiden avulla vahvistusoppimisessa opitaan, mitkä toiminnot ovat olleet koneoppimisongelman ratkaisun kannalta hyviä ja mitkä huonoja [18]. Vahvistusoppimisessa palautetta tarvitsee kerätä paljon toimintaympäristöstä, jotta oppiminen olisi mahdollista [18]. Vahvistusoppimista tapahtuu esimerkiksi silloin, kun ihminen opettelee pelaamaan shakkia: ihmisen on ensiksi pelattava useampia pelejä oppiakseen, mitkä siirrot ovat olleet hyviä tai huonoja [15] [18].

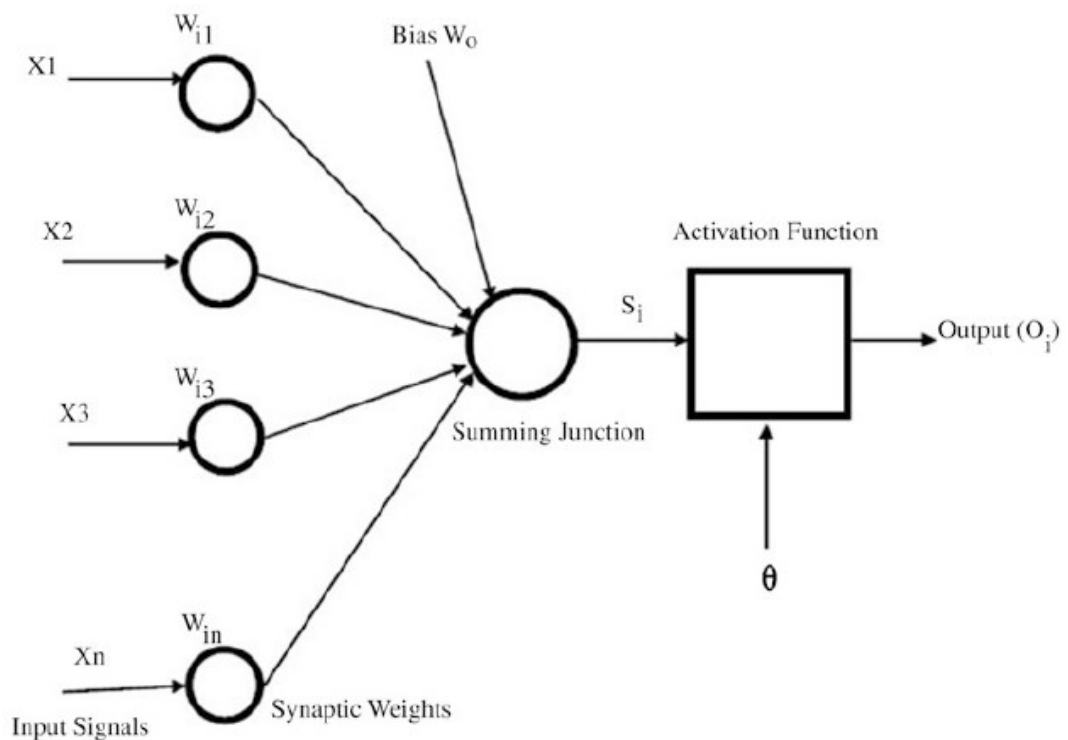
## **3.2 Neuroverkko**

Keinotekoinen neuroverkko (Artificial Neural Network) on matemaattinen malli ihmisen biologisista aivoista [19]. Jatkossa tässä diplomityössä neuroverkoilla tarkoitetaan keinotekoisia neuroverkkoja. Biologisissa aivoissa neuronit ovat aivohermoston perustason signalointiyksiköitä, joiden tehtävänä on käsitellä ja siirtää tietoa toisille neuroneille synapsien kautta sähköisillä impulsseilla [19] [20]. Mikäli impulssit ovat riittävän voimakkaita, ne pystyvät aktivoimaan synapsin ja siten siirtämään tiedon toiselle neuronille [20].

### 3.2.1 Neuroverkon rakenne

Neuroverkko koostuu kerroksista, jotka sisältävät laskentayksiköitä eli neuroneita. Kerrosten välissä on linkkejä eli synapseja, joilla neuronit yhdistetään seuraavan kerroksen neuroneihin [19]. Jokaiselle synapsille on määritelty oma painokerroin, joiden avulla lasketaan neuronin tuottama tulos [19]. Jokaisessa kerroksessa olevalla neuronilla on aktiivointifunktio, jossa määritellään, lähettääkö neuroni tulosta eteenpäin [19].

Kuvassa 3 on esitetty tarkemmin neuronin matemaattista rakennetta. Neuronin vastaanottaa syötteitä ( $X_i$ )  $n$  kappaleen verran ja jokaisella on oma syötepaino ( $W_i$ ) [19]. Syötteiden painoilla määritellään eri yhteyksien merkitystä [19]. Suuret painokertoimet viittaavat siihen, että kyseessä on tärkeä yhteys [21]. Sen sijaan pienet kertoimet viittaavat siihen, että yhteys ei ole tärkeä [21]. Syötteet ja niitä vastaavat syötepainot siirretään summausfunktioon [19]. Summausfunktiossa (Summing Junction) syötteistä ja niiden painokertoimista (Synaptic Weights) muodostetaan tulo kertomalla ensin jokainen syöte ja sen painokerroin keskenään ja lopuksi summaamalla saadut tulot keskenään kuten kaavassa 2.1 on esitetty [19].



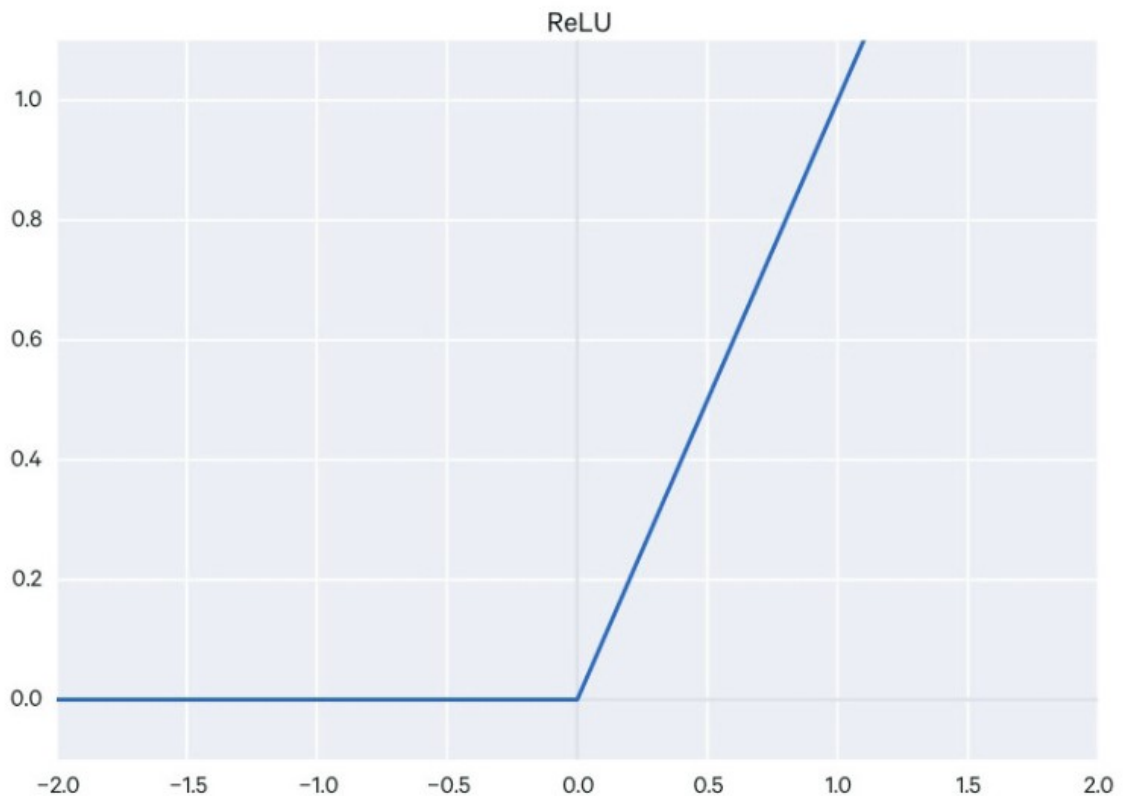
**Kuva 3.** Neuronin matemaattinen rakenne [19].

$$S_i = w_0 + \sum_{j=1}^n w_{ij}x_j \quad (2.1)$$

Summausfunktiossa kaavan 2.1 mukaan muodostettu tulos siirretään aktivointifunktiolle (Activation Function), jossa muodostetaan neuronin tulos [19]. Eri aktivointifunktioissa neuronin tuottama tulos lasketaan eri tavoilla [19]. Esimerkiksi kuvien luokittelussa voidaan käyttää ReLu (Rectified Linear Unit), Sigmoid ja Softmax-aktivointifunktioita [19].

ReLu on yksi käytetyimmistä aktivointifunktioista. ReLu kerroksia on tyypillisesti konvoluutiokerrosten jälkeen. Kuvasta 4 huomaa, että ReLu antaa negatiivisilla arvoilla nollan ja positiivisilla arvoilla ReLu antaa kyseisen arvon [22] [23]. ReLu:n funktio on esitetty kaavassa 2.2. ReLu on yksinkertaisempi aktivointifunktio muihin aktivointifunktioihin verrattuna. Neuroverkon opettaminen ReLu:lla on nopeampaa verrattaessa muihin aktivointifunktioihin. [23]

$$f(x) = \max(0, x) \quad (2.2)$$

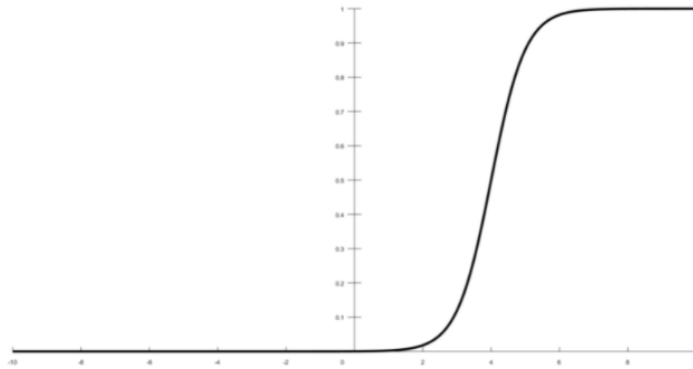


**Kuva 4.** ReLu-aktivointifunktion kuvaaja [22].

Sigmoid on matemaattinen aktivointifunktio, joka tuottaa ulostuloksi arvoja, jotka ovat 0 ja 1:n välillä, kuten kuvasta 5 nähdään. Sigmoidin funktio on esitetty kaavassa 2.3. [23]

$$y = f(x) = \frac{1}{1 + e^{-x}} \quad (2.3)$$





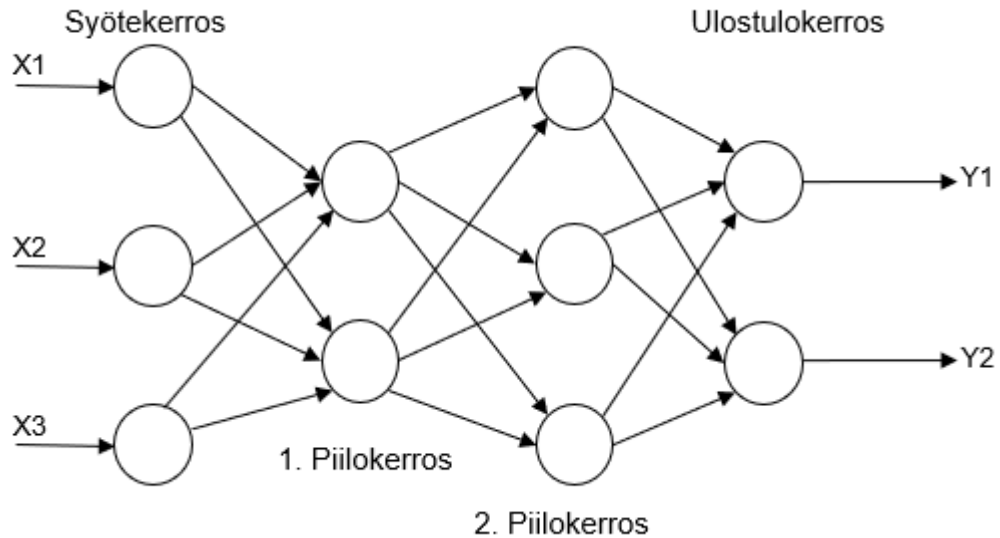
**Kuva 5.** Sigmoid-aktivointifunktion kuvaaja [23].

Softmax-aktivointifunktiota käytetään, kun luokiteltavien luokkien määrä on vähintään 2 [24]. Softmax tuottaa jokaiselle ulostuloluokalle todennäköisyyden ja kaikkien ulostuloluokkien todennäköisyyksien summa on 100 % [24]. Kaavassa 2.4 on Softmax:n funktio, joka on monimutkaisempi verrattaessa aikaisemmin esiteltyihin aktivointifunktioihin. [24]

$$y_i = \frac{\exp(y_i)}{\sum_{j=0}^n \exp(y_j)} \quad (2.4)$$

### 3.2.2 Monikerroksinen neuroverkko

Monikerroksinen neuroverkko (Multilayer Perceptron) on eteenpäin suuntautuva neuroverkko, joka koostuu yhdestä tai useammasta kerroksesta syöte- ja tulokerroksen välillä kuten kuvasta 6 nähdään [19]. Syöte- ja tulokerrosten välisiä kerroksia kutsutaan piilokerroksiksi [19]. Nämä piilokerrokset on linkitetty toisiinsa siten, että tieto kulkee verkossa vain yhteen suuntaan [19]. Neuroverkossa neuronit välittävät tietoa ainoastaan eteenpäin siten, että tieto siirtyy syötekerroksesta ensimmäiseen piilokerrokseen, ja piilokerrosten kautta tulokerrokseen [19]. Eri kerrosten neuronien välille ei ole toteutettu takaisinkytkentää [19].



**Kuva 6.** Eteenpäin suuntautuva neuroverkko [19].

### 3.2.3 Neuroverkon opettaminen

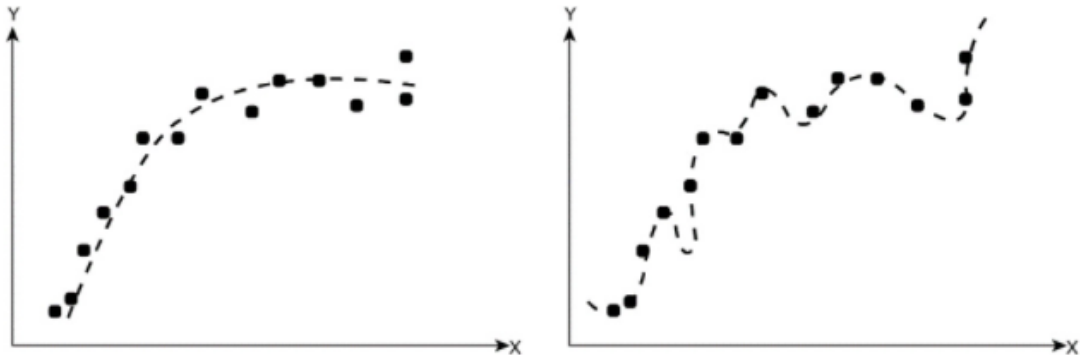
Neuroverkon opettamisessa on tarkoitus etsiä neuronien painokertoimille ja muille parametreille sellaiset painot, joilla neuroverkko tuottaisi halutun lopputuloksen [25]. Neuroverkon opettamisen voidaan ajatella olevan epälineaarinen optimointiongelma, jossa pyritään löytämään sopivat neuroverkon painot [25]. Sopivia painojen arvoja etsitään minimoimalla kustannusfunktioita, ja näin pyritään saavuttamaan haluttu lopputulos [25].

Neuroverkkoa opettaessa ohjatulla opettamisella syötetietoja tarvitsee usein esikäsitellä [26]. Tiedon esivalmistelussa muodostetaan syötepareja  $(x_i, y_i)$ , joiden avulla neuroverkkoa opetetaan [26]. Syöteparien muodostamisen jälkeen tiedot jaetaan opetus- ja testausaineistoihin [26]. Opetusaineiston kooksi asetetaan tyypillisesti 70 – 80 % koko aineistosta, ja loput jätetään testausta varten [26]. On myös mahdollista jakaa tietoa kolmeen osaan, jotka ovat opetus-, testaus- ja validointiaineistot. Validointiaineistoa käytetään opetuksen aikana arvioimaan neuroverkon tarkkuutta, ja voidaan tarkastella paraneeko neuroverkon tarkkuus jokaisen opetuskierroksen (epoch) jälkeen [26]. Opettamisessa ongelmaksi voi muodostua jos tietoa on liian pieni eikä siitä riitä riittäviä määriä neuroverkon opetukseen, testaukseen ja validointiin [26].

### 3.2.4 Ylioppiminen

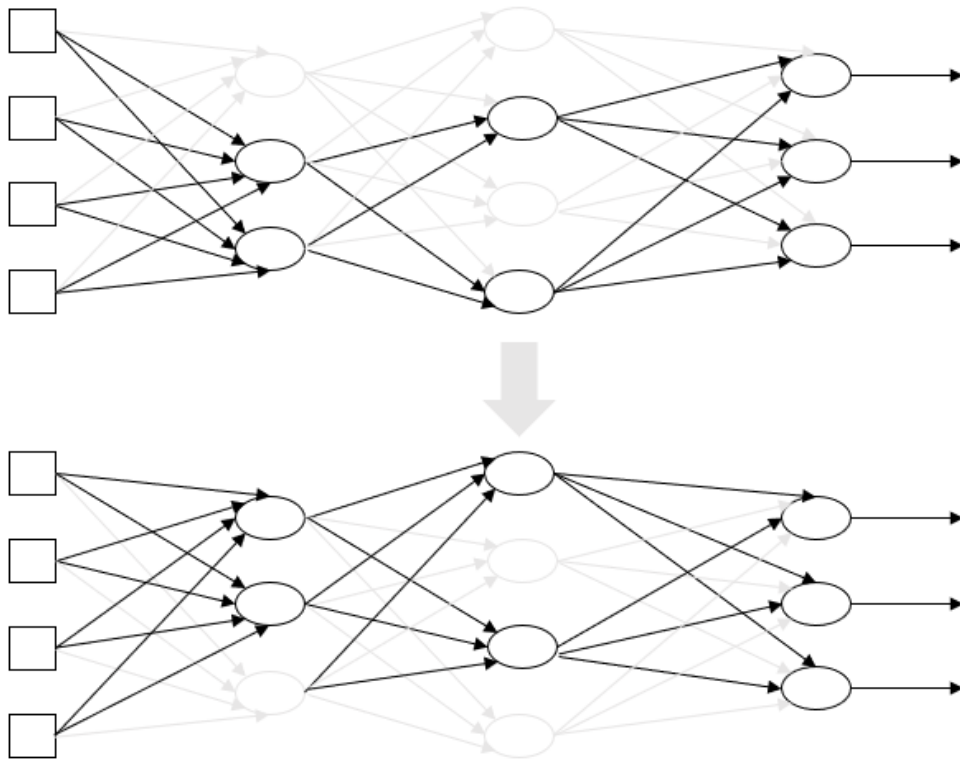
Neuroverkkoa opettaessa voi tulla tilanteita, joissa neuroverkko on liian joustava tai opetustietoa on liian vähän mallin opettamiseen. Tällöin neuroverkko tunnistaa ainoastaan opetusvaiheessa esiintyviä syötteitä, mutta ei kykene tunnistamaan sille opetusvaiheen

jälkeen annettuja syötteitä [27]. Kuvassa 7 esitetään esimerkit hyvin oppineesta ja ylioppineesta regressiosta. Ylioppinut malli ei opi vain opetustietojen syötteiden ja ulostulojen välisiä suhteita, vaan näiden lisäksi se oppii aineistossa esiintyvät kohinat ja poikkeamat [28] [29]. Usein ylioppivissa malleissa parametrien lukumäärä on liian suuri suhteessa opetettaviin tietopisteisiin tai malli on liian monimutkainen [28] [29].



**Kuva 7.** Hyvin oppinut (vasen) ja ylioppinut regressio (oikea) [28].

Neuroverkot ovat herkkiä ylioppimaan, jos niissä on suuri määrä piilokerroksia [30]. Yksi tehokas keino vähentää neuroverkon ylioppimista on lisätä siihen pudotuskerroksia (dropout) [30]. Pudotuskerroksissa valitaan satunnaisesti, mitä neuroverkon neuroneita opetetaan opetuskierröksellä (epoch) [30]. Kuva 8 esittää pudotuskerrosten toimintaa eri opetuskierröksillä: ylempi kuva esittää aiempaa opetuskierröksestä ja alempi kuva sen jälkeistä opetuskierröksestä [30]. Yhdessä opetuskierröksessä opetetaan vain tiettyjä neuroneita ja muita neuroneita ei käytetä kuten kuvassa 8 esitetään [30]. Seuraavalla opetuskierröksellä puolestaan opetetaan niitä neuroneita, joita edellisellä opetuskierröksellä ei käytetty [30]. Pudotuskerroksen arvo voidaan valita ja se määrää opetuskierrösten aikana käytettävien neuronien lukumäärän. Esimerkiksi pudotuskerroksen arvo 0,5 tarkoittaa, että 50 % neuroneista ei opeteta opetuskierrösten aikana [30].



*Kuva 8. Pudotuskerroksen toiminta opetuskerrosten aikana [30].*

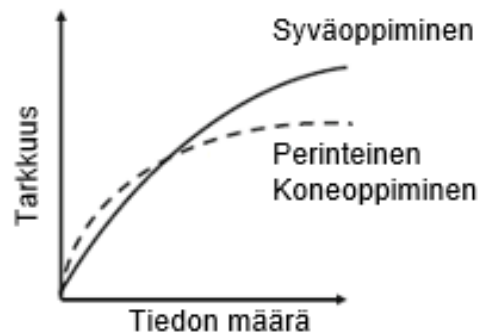
### 3.3 Syväoppiminen

Syväoppiminen on koneoppimisen osa-alue, jossa neuroverkko koostuu useammista kerroksista [31]. Jokainen lisätty kerros neuroverkossa kasvattaa sen syvyyttä [31]. Syväoppimisessa neuroverkon alemmat kerrokset muodostavat käsiteltävästä datasta yksinkertaisia piirteitä, kun taas ylemmät kerrokset muodostavat alempien kerrosten keräämistä ylemmän tason piirteitä. [31] Mitä alempi kerros on, sitä yksinkertaisempia piirteitä se kerää [18]. Esimerkiksi kuvantunnistuksessa syväoppivan neuroverkon alemmat kerrokset tunnistavat kuvista niiden reunoja tai hahmojen rajaviivoja, kun taas ylemmät kerrokset tunnistavat paljon monimutkaisempia piirteitä kuten kasvoja. [18]

Syväoppivat neuroverkot ovat tehokkaita esimerkiksi kuvantunnistuksessa, puheentunnistuksessa ja luokitteluongelmissa, erityisesti silloin kun datamäärä on suuri [18]. Syväoppivien menetelmien käyttö on yleistynyt viime vuosina, koska dataa on ollut enemmän saatavilla, ja tietokoneiden laskentateho on kasvanut [32].

Alla olevassa kuvassa 9 havainnollistetaan, että syvemmät neuroverkot ovat parempia vaihtoehtoja kuin perinteiset koneoppimismallit, kun käsitellään suuria datamääriä. Syvät

neuroverkot tuottavat tarkempia tuloksia kuin perinteiset koneoppimismallit suurilla datamäärillä [32]. Sen sijaan perinteiset koneoppimismallit soveltuvat paremmin tilanteisiin, joissa dataa on vähemmän [32].



*Kuva 9. Syväoppivan ja perinteisen koneoppimismallin ero tarkkuudessa datamäärän suhteen [32].*

### 3.4 Konvoluutioneuroverkko

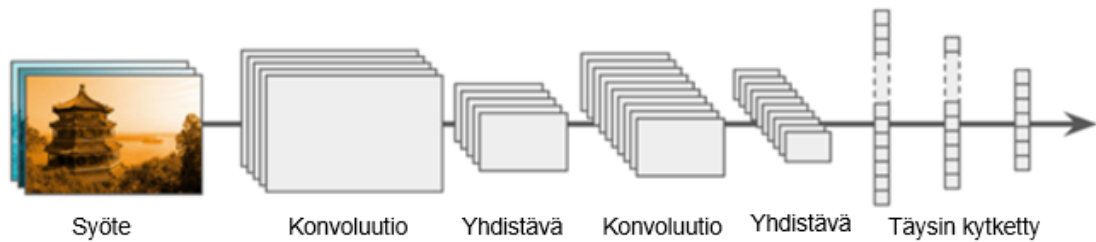
Konvoluutioneuroverkko (Convolutional Neural Network) on ohjatulla oppimisella toteutettu syväoppiva neuroverkko, jossa on yksi tai useampia konvoluutiokerroksia. Konvoluutioneuroverkot sopivat hyvin luokitteluun [33] [31]. Konvoluutioneuroverkkoa käytetään oppimaan kuvajoukosta piirteitä toteuttamalla konvoluutio-operaatio [33]. Konvoluutiota käytetään keräämään erilaisia piirteitä kuvista toteuttamalla miljoonia kertolaskuja ja summauksia suotimien (filter) ja kerättyjen piirteiden välillä. [34]

#### 3.4.1 Konvoluutioneuroverkon rakenne

Konvoluutioneuroverkko koostuu muutamista erityyppisistä kerroksista, jotka ovat erilaisia toiminnoiltaan. Perinteinen konvoluutioneuroverkko koostuu konvoluutiokerroksista, joissa aktivointifunktiona on usein ReLu, ja niiden jälkeisistä yhdistävistä (pooling) kerroksista [35]. Tällaisia kerrospareja on peräkkäin  $N$  kappaletta, riippuen siitä kuinka syvän konvoluutioverkon haluaa toteuttaa. Konvoluutioneuroverkossa käsiteltävä kuva pienenee ilman, että siitä katoaa kuvan ydintietoa [33] [35].

Kuvassa 10 on tyypillinen konvoluutioneuroverkko, joka koostuu konvoluutio- ja yhdistämiskerrospareista, joiden jälkeen on toteutettu täysin kytkettyjä kerroksia (dense), joiden aktivointifunktio on myös ReLu [35]. Näiden kerrosten jälkeen tulee viimeinen kerros, jossa voidaan esimerkiksi toteuttaa luokittelua. [35] Tyypillisesti viimeisen kerroksen ak-

tivointifunktio on Softmax, joka antaa tulokseksi jokaiselle luokalle oman todennäköisyyden [36]. Esimerkiksi jos halutaan tunnistaa oliko kuvassa kissa vaiko koira, niin silloin viimeisen kerroksen aktivointifunktiona on Softmax [36]. Tällä tavoin ulostuloksi saadaan kaksi todennäköisyyttä, joista toinen on kissan todennäköisyys ja toinen on koiran todennäköisyys ja näiden yhteenlaskettu summa on 100 % [36].



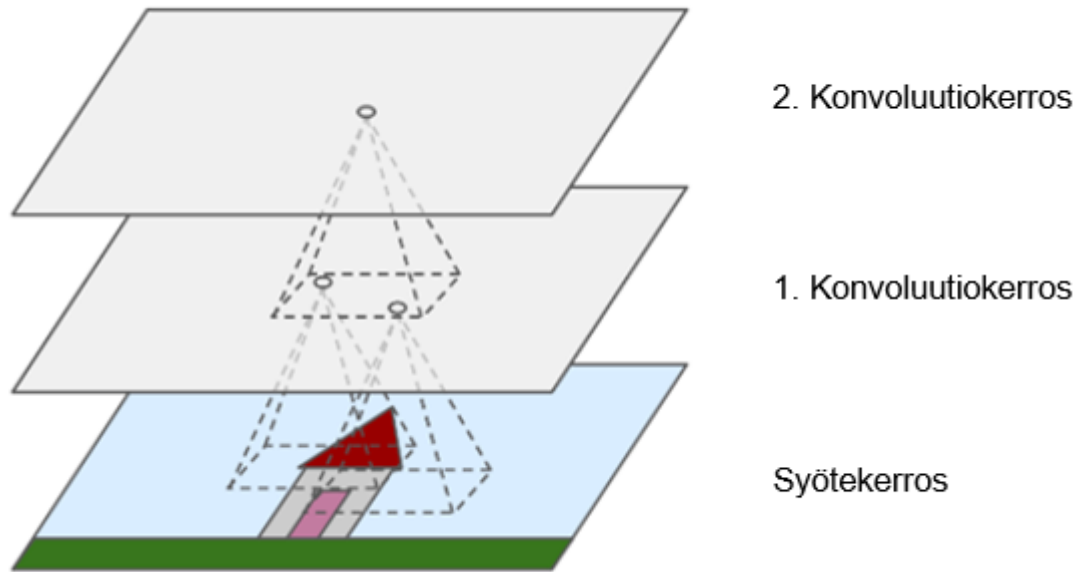
*Kuva 10. Tyypillinen konvoluutioverkon rakenne [35].*

### 3.4.2 Konvoluutiokerros

Konvoluutiokerros on konvoluutioneuroverkon tärkein osa, jossa tunnistetaan kuvissa esiintyviä piirteitä [35]. Kuvassa 11 on esitetty konvoluutiokerrosten toimintaa. Syötekuvaa käydään läpi konvoluutioikkunalla, joka kerää kuvasta piirteitä ensimmäiselle konvoluutiokerrokselle [35]. Ensimmäisen konvoluutiokerroksen neuroneita on vähemmän kuin syötekuvan pikseleitä [35]. Toisessa konvoluutiokerroksessa käydään läpi ensimmäisen konvoluutiokerroksen keräämiä piirteitä syötekuvasta [35].

Konvoluutioneuroverkon ensimmäisessä konvoluutiokerroksessa keskitytään tunnistamaan alatasen ominaisuuksia [35]. Alemman tason ominaisuuksista muodostetaan konvoluutiokerroksella ylemmän tason ominaisuuksia, jotka edelleen siirretään seuraavalle konvoluutiokerrokselle [35]. Tätä toistetaan konvoluutiokerrosten välillä [35].

Konvoluutiokerrosten sekä konvoluutioikkunan koon valinta vaikuttavat konvoluutioneuroverkon tarvitsemaan laskentatehoon [35]. Yleinen virhe valittaessa konvoluutioikkunan kokoa on valita kerralla liian suuri ikkunan koko [35]. Käyttämällä useampaa kerrosta, joissa on pienempi ikkuna, saadaan vähennettyä siirrettävien parametrien määrää ja säästetään tarvittavaa laskentatehoa verrattaessa suuremman konvoluutioikkunan käyttöön [35].

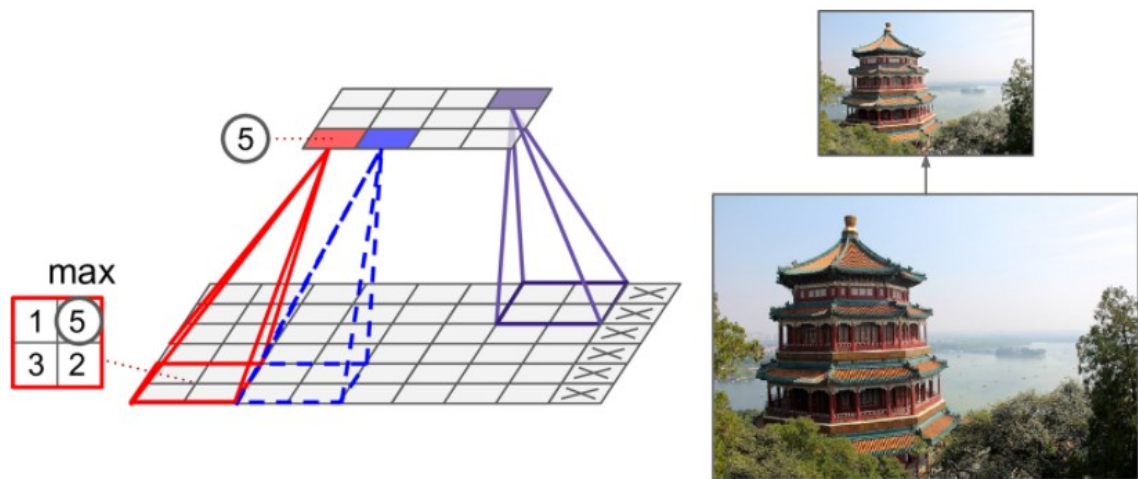


**Kuva 11.** Syötekuvan ja konvoluutiokerrosten välinen yhteys [35].

### 3.4.3 Yhdistämiskerros

Konvoluutiokerrosten jälkeen konvoluutioverkossa tulee yleensä yhdistämiskerros, jonka tarkoituksena on vähentää siirrettävien parametrien määrää. Tämä tehostaa oppimista, sillä tällä tavoin saadaan vähennettyä tarvittavaa laskentakapasiteettia ja vähennetään ylioppimisen riskiä [22] [35]. Yhdistämiskerroksessa pienennetään konvoluutiokerroksesta saatujen syötteiden dimensionaalisuutta siten, että ydintieto ei häviä [22] [35].

Neuronin tullessa yhdistämiskerrokseen konvoluutiokerroksesta sitä aletaan käydä läpi määritellyn askelien (striden) mukaan. Yhdistämiskerroksen neuroneilla ei ole omia syötepainoja, vaan arvot saadaan käyttämällä yhdistämisfunktiota, joka laskee esimerkiksi suurimman arvon tai keskiarvon. Kuvassa 12 on esimerkki suurimman arvon ottamisesta talteen (max pooling), joka on yleisin käytetty yhdistämiskerros. Kuvassa 12 suurimman arvon ottamiseen käytetään 2x2 kokoista aluetta eli kerneliä, jossa liikutaan 2 ruudun verran. Kernel-alueen suurin arvo otetaan talteen, ja muut syötearvot tiputetaan pois [35].



*Kuva 12. Suurimman arvon valinta [35].*

#### 3.4.4 Täysin kytketty kerros

Täysin kytkettyjä kerroksia käytetään konvoluutioneuroverkoissa konvoluutio- ja yhdistämiskerrosten jälkeen [33]. Kerros käyttäytyy kuten perinteinen neuroverkko, missä jokainen neuroni käsitellään yksittäisinä [33]. Viimeisestä täysin kytketystä kerroksesta saadaan tulokseksi luokittelun tulos [33]. Tähän viimeisen kerroksen kokoon vaikuttaa luokiteltavien luokkien määrä. [33]



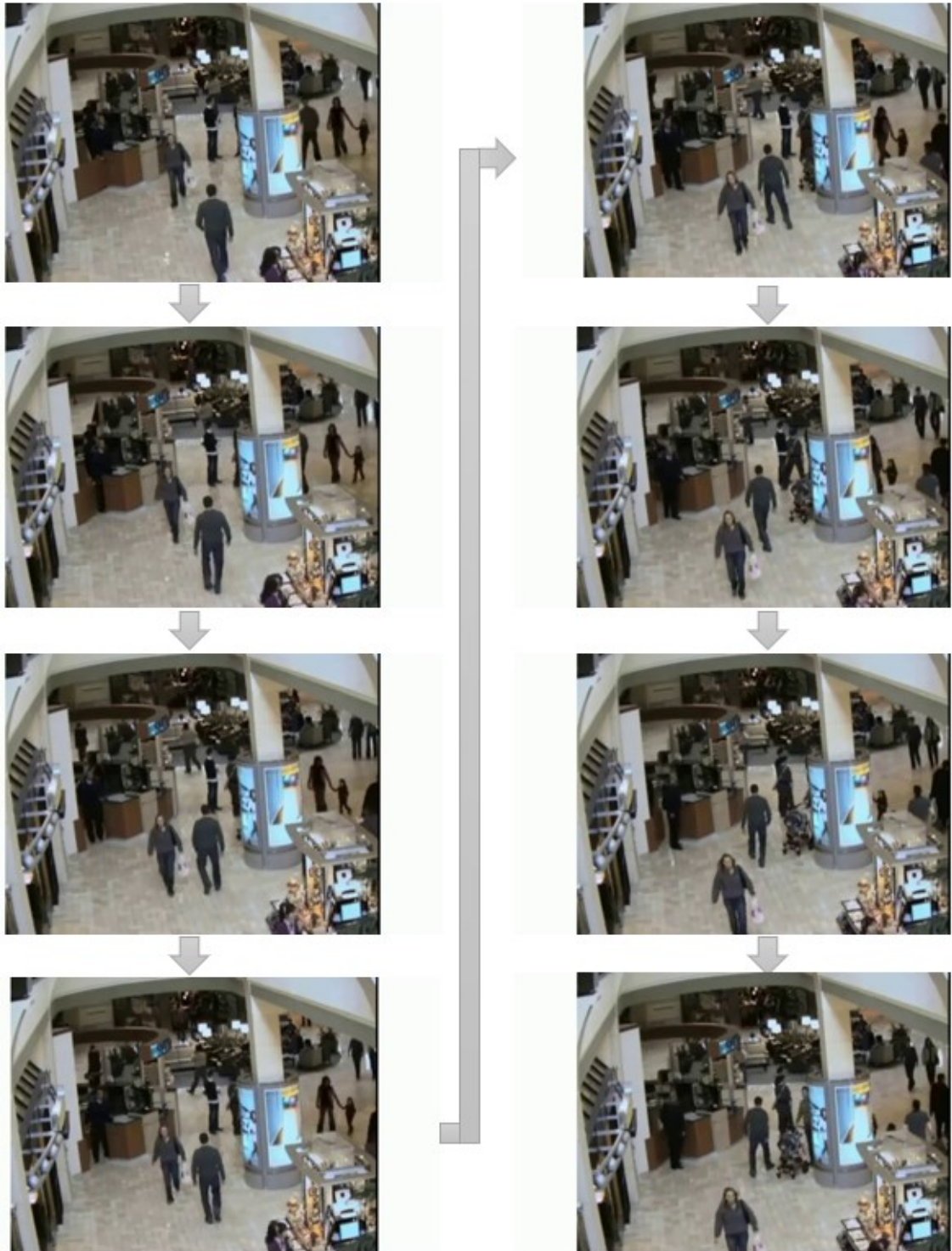
## 4. TIETOAINEISTO

Neljännessä luvussa kerrotaan, mitä tietoaaineistoa työssä on käytetty sekä miten koneoppimismallille on toteutettu opetus-, validointi- ja testaus-tietoaaineistot.

### 4.1 Tietoaaineisto

Tässä diplomityössä käytetään valmiiksi kerättyä tietoaaineistoa koneoppimismallin opettamiseen, validointiin ja testaamiseen. Käytettävä tietoaaineisto on UBI Fights dataset [37], joka on portugalilaisen yliopiston (University of Beira Interior, UBI) kokoama tietoaaineisto. Tietoaaineisto koostuu tuhannesta videosta, joista 215 sisältää erityyppistä väkivaltaa ja 785 videota sisältävät ihmisten tavallista kanssakäymistä. Videoita on kerätty erilaisista lähteistä kuten valvontakameroista, kauppakeskusten kameroista, liikennekameroista ja ihmisten kuvaamista videoista. Väkivaltaiset videot koostuvat videoista, joissa tapahtuu väkivaltaa sisätiloissa tai ulkona kahden tai useamman ihmisen välillä. Ei-väkivaltaiset videot sisältävät ihmisten tavallista kanssakäymistä kaupunkikatukuvassa kuten kävelyä, juttelua ja kokoontumista pienissä tai suuremmissa ryhmissä. Tielikennekameroiden kuvaamat videot sisältävät jalankulkijoiden ja autoilijoiden toimintaa keskustassa ja suuremmilla tieliikenne-osuuksilla.

UBI Fights tietoaaineisto on tämän diplomityön tutkimuskysymyksen ratkaisuun soveltuva, sillä UBI Fights tietoaaineistoon kuuluvat videot sisältävät niitä tilanteita ja tapahtumia, joita kehitettävän koneoppimismallin halutaan oppivan tunnistamaan. Tässä luvussa esitetyt kuvasarjat on otettu UBI tietoaaineistosta. Tietoaaineistosta muodostetuissa kuvasarjoissa kuvia on otettua 2 kuvaa per sekunti. Jokaisessa kuvasarjassa kuvataan 4 sekunnin pätkä yhdestä videopätkästä. Kuvissa 13 ja 14 on esitetty tietoaaineiston ei-väkivaltaista sisältöä kuvasarjoina. Kuvassa 13 ihmisiä kävelee sisätiloissa. Kuvassa 14 ihmisiä kävelee ulkotiloissa. Kuvissa 15 ja 16 on esitetty tietoaaineiston väkivaltaista sisältöä kuvasarjoina. Kuvassa 15 on heikko kuvanlaatu kahden ihmisen tappelusta. Kuvassa 16 on ihmisten kadulla tapahtuva joukkotappelu. Kuvista huomaa, että tietoaaineiston videoiden kuvanlaatu vaihtelee huomattavasti. Tämä voi tuottaa ongelmia koneoppimismallin opetuksessa.



*Kuva 13. Kuvasarja ei-väkivaltaisesta käyttäytymisestä sisätiloissa [37].*



**Kuva 14.** Kuvasarja ei-väkivaltaisesta käyttäytymisestä ulkona [37].



**Kuva 15.** Kuvasarja kahden henkilön väkivaltaisesta käyttäytymisestä [37].



*Kuva 16. Kuvasarja väkivaltaisesta joukkotappelusta [37].*

## 4.2 Opetusaineiston kokoaminen

Opetusaineiston kokoaminen aloitettiin läpikäymällä kaikki UBI Fights tietoaaineiston videot yksi kerrallaan. Läpikäynnin yhteydessä tarkasteltiin sitä, sisälsikö video tilanteita, joita koneoppimismallin haluttiin oppivan. Mikäli video sisälsi tällaisia tilanteita, niin se otettiin talteen tarkempaa läpikäyntiä varten. Seuraavissa aliluvuissa käydään tarkemmin läpi, millä perusteilla videoita valittiin ja hylättiin tietoaaineistosta.

Videoiden valinnan jälkeen videoiden määrä oli huomattavasti pienempi, eikä tietoaaineisto enää riittänyt sellaisenaan koneoppimismallin opettamiseen. Videoaineistoa tarvitsi kasvattaa joko lisäämällä uusia videoita, tai suorittamalla keinotekoisia tiedonlisäämistä (data augmentation) olemassa olevilla videoilla. Tässä työssä päädyttiin kasvattamaan tietoaaineistoa keinotekoisella tiedonlisäämisellä. Olemassa olevien videoiden pituus vaihteli hyvinkin paljon ja videoiden kestoajat ovat väliltä 30 s – 120 min. Jokaisesta valitusta videosta valittiin kolmesta kuuteen 5 s pituisia pätkää. Jokaisesta videosta valittiin vain tämän verran lyhyempiä pätkiä, sillä pidemmistä videoista olisi saatu muodostettua hyvin suuri määrä 5 s pätkiä. Tämä olisi voinut johtaa siihen, että tekoälymalli oppisi opetusvaiheessa vain joidenkin videoiden tapauksia. Tässä työssä tarkoituksena on muodostaa mahdollisimman kattava tietoaaineisto, jonka avulla voidaan opettaa tekoälymalli, joka soveltuu erottamaan väkivaltaisen ja ei-väkivaltaisen käyttäytymisen useammasta paikasta ja tilanteesta. Kaikki lyhyemmät 5 s kestoiset videopätkät sisältävät vain joko väkivaltaista tai ei-väkivaltaista käyttäytymistä. Kun kaikista videoista oli saatu tehtyä pienempiä pätkiä ja valittua jokaisesta videosta kolmesta kuuteen 5 s pituisia videopätkää, niin tietoaaineiston kokonaismäärä oli 450 videopätkää, joista 225 sisältää väkivaltaa ja 225 ei sisältänyt väkivaltaa.

Koneoppimismallin opetusta varten tietoaaineisto jaettiin kolmeen osaan: opetus-, validointi- ja testausaineisto. Ennen varsinaisen opetusaineiston muodostamista tietoaaineistosta otettiin 10 % videoita, joista muodostetaan koneoppimismallin testausaineisto. Testausaineistoa käytettiin valmiin mallin tarkkuuden ja toimivuuden mittaamiseen. Opetusvaiheessa käytettäväksi jäi 90 % tietoaaineiston videoista. Nämä videot jaettiin opetusaineistoon, joka oli 80 % videoista, ja validointiaineistoon, joka on 10 % videoista. Opetusaineistolla opetetaan itse koneoppimismallia. Validointiaineistolla optimoidaan käytettäviä meta-parametreja. Seuraavissa aliluvuissa kerrotaan, millä perusteilla videoita valittiin tai hylättiin.

### **4.2.1 Opetusaineistoon valittuja videoita**

Opetusaineistoon valittiin videopätkiä, jotka katsottiin sisältönsä puolesta soveltuvan parhaiten luokittelumallin opettamiseen. Valintaan vaikuttavia piirteitä olivat videon kuvanlaatu, kuvauskulma, henkilöiden määrä, videoiden sisältö ja henkilöiden etäisyys kamerasta. Kuvissa 17, 18 ja 19 on opetusaineistoon valituista videoista muodostettuja kuvasarjoja. Kuvassa 17 ihmisiä kävelee ulkotiloissa. Kuvassa 18 ihmisiä kävelee sisätiloissa. Kuvassa 19 on ulkona kuvattu joukkotappelu. Kuvissa olevat kuvasarjat täyttivät vaaditut kriteerit, sillä kuvauskulma oli hyvä ja videoiden sisältö soveltui luokittelumallin opetukseen. Kuvauskulman ollessa henkilöiden yläpuolella ihmisten liikkeet eivät peity kuvassa. Esimerkiksi jos kuvauskorkeus oli ihmisten korkeudella, niin taaimmaisten ihmisten liikkeitä ei näkynyt, sillä etummaisiet ihmiset peittäisivät kuvan. Kriteerit täyttävät videot otettiin talteen, ja muita tietoaineistoon kuuluvia videoita ei tässä diplomityössä enempää hyödynnetty. Kuvista voitiin havaita ei-väkivaltaisten ja väkivaltaisten videoiden välisiä eroja. Väkivaltaisissa videoissa tapahtui huomattavasti enemmän nopeita muutoksia kuin ei-väkivaltaisessa videossa.



*Kuva 17. Kuvasarja ei-väkivaltaisesta käytöksestä sisätiloissa [37].*





*Kuva 18. Kuvasarja ei-väkivaltaisesta käytöksestä ulkotiloissa [37].*

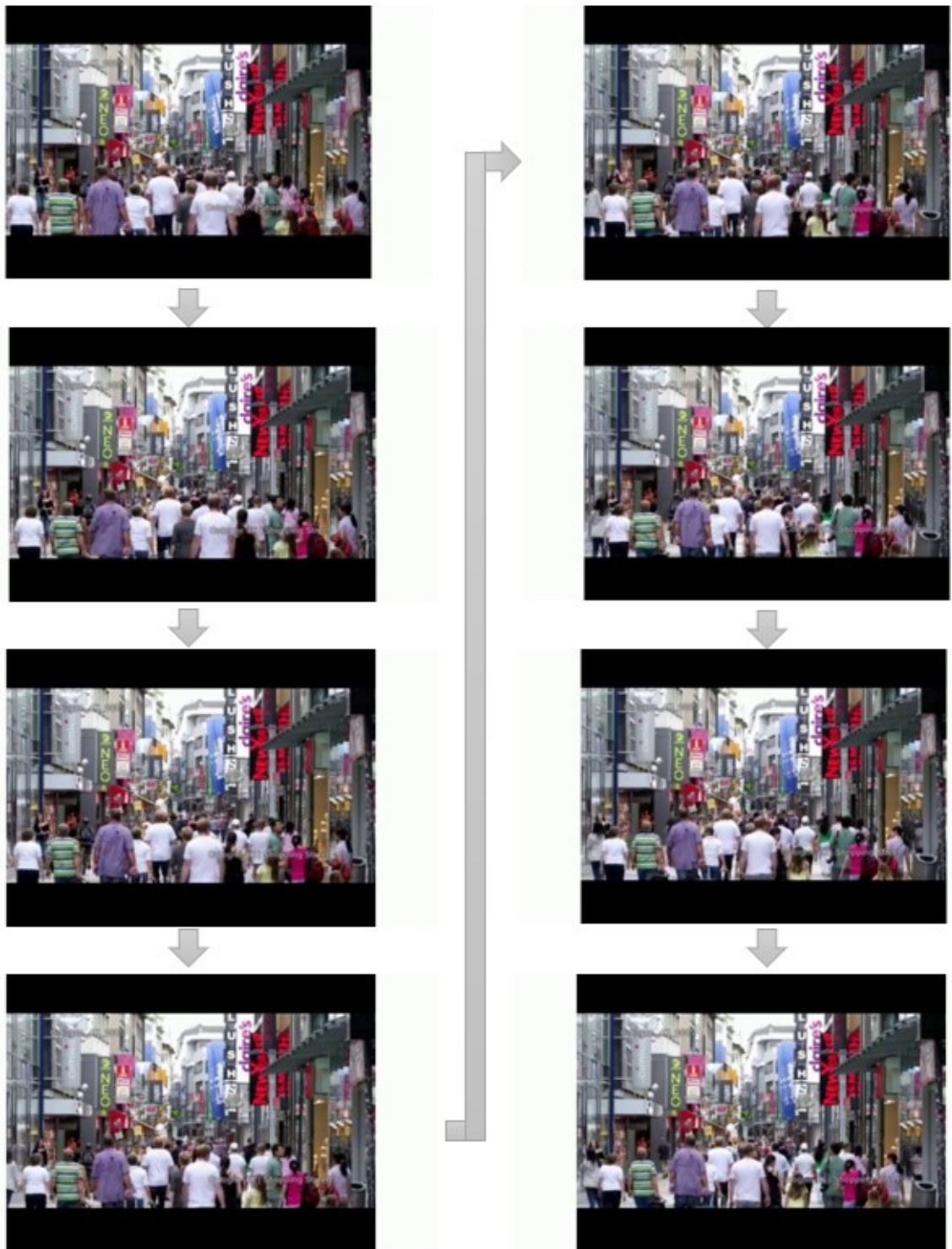


*Kuva 19. Kuvasarja väkivaltaisesta videosta [37].*

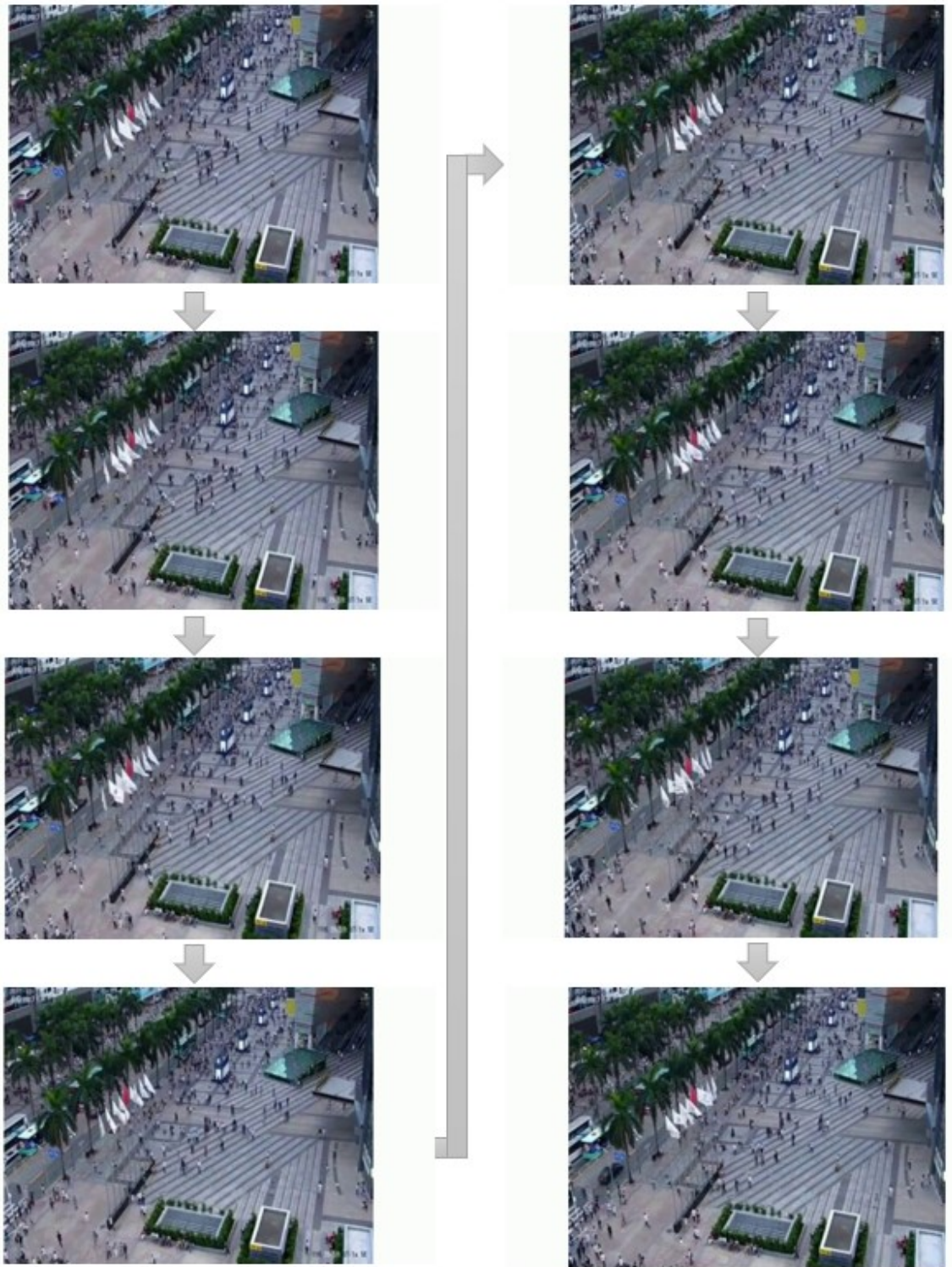
#### 4.2.2 Opetusaineistosta hylätyjä videoita

Opetusaineistosta hylättiin ne videot, jotka eivät vastanneet edellisessä aliluvussa esitellyjä vaatimuksia. Suurinta osaa tietoaaineistoon kuuluvista videoista ei pystytty hyödyntämään koneoppimismallin opetuksessa. Hylätyt videot sisälsivät tilanteita, joita ei tässä työssä katsottu tarpeelliseksi opettaa luokittelumallille. Tällaisia tilanteita olivat muun muassa joukkotappeluita ravintoloissa, joissa hakattiin toisia tuoleilla tai kännykällä kuvattuja kahden ihmisen välisiä tappeluita, joissa kuva tärisi kokoajan. Myös videot, joiden

tapahtumat eivät olleet kaupunkikatukuvassa tapahtuvia tilanteita hylättiin opetusaineistosta. Kuvissa 20, 21 ja 22 on muodostettu kuvasarjoja hylätyistä videoista. Kuvan 20 kuvasarja hylättiin tietoaineistosta kuvauskulman takia. Videossa kuvauskulma on ihmisten tasolla ja kuvasarjan kuvista pystyttiin huomaamaan, mitenkä etummaisiet ihmiset peittävät kuvan taaimmaisiet ihmiset. Kuvan 21 kuvasarja hylättiin tietoaineistosta kuvausetäisyyden takia. Kuvasarjan kuvista voitiin havaita, että kuvausetäisyyden ollessa liian suuri kuvissa esiintyvät henkilöt ovat liian pieniä tunnistettavaksi. Kuvan 22 kuvasarja hylättiin tietoaineistosta epäselvyyden takia. Kuvasarjan kuvista voitiin huomata, että siinä tapahtuu väkivaltaa, mutta ei riittävän selkeästi, että se olisi valittu tietoaineistoon.



*Kuva 20. Kuvasarja hylätystä peittävästä videosta [37].*



**Kuva 21.** Kuvasarja hylätystä kaukaa kuvatusta videosta [37].



**Kuva 22.** Kuvasarja hylätystä epäselvästä videosta [37].

## 5. MENETELMÄ

Viidennessä luvussa kuvataan koneoppimismallin toteutus ohjelmistotuotannon periaatteiden mukaisesti.

### 5.1 Tekniikoiden valinta

Koneoppimismallin eri vaiheiden toteutuksessa hyödynnettiin useita kirjastoja ja ohjelmistokehystä. Alla olevassa taulukossa 1 on listattu tekniikat, joita käytettiin tässä diplomityössä.

**Taulukko 1 Käytettävät tekniikat**

<b>Tekniikka</b>	<b>Versio</b>	<b>Käyttötarkoitus</b>
Visual Studio Code	1.66.2	Ilmainen ohjelmointiympäristö, jossa ohjelmiston teko suoritettiin.
Anaconda	4.9.2	Anacondalla luotiin virtuaalinen käännösympäristö, jonne ladattiin kaikki tarvittavat kirjastot.
Python	3.6.13	Pythonilla toteutettiin ohjelman runko sekä tiedostojen lukeminen kansioista.
OpenCV	3.4.1	OpenCV:llä toteutettiin videoiden muuntaminen kuvakehyksiksi ja muuttamalla kuvakehysten kokoa sekä väriä.
Numpy	1.19.2	Numpy:lla muunnetaan saadut videokehysten pikselitiedot numpy-tilukoksi, joka voidaan syöttää mallille opetusdataksi.
Keras	2.3.1	Keras:lla toteutettiin koneoppimismalli.
Tensorflow	2.0.0	Avoimenlähdekoodin koneoppimisen kirjasto ja ympäristö.

## 5.2 Mallin toteutuksen vaiheet

Koneoppimismallin toteutus jaettiin kahteen vaiheeseen. Ensimmäinen vaihe koostui videoiden esiprosessoinnista ja toinen mallin toteutuksesta. Mallin toteutus oli iteroitava prosessi, jossa mallia kehitettiin ja testattiin. Testauksesta saatavan palautteen perusteella mallille tehtiin muutoksia, jonka jälkeen sitä testattiin uudelleen. Tätä iterointia jatkettiin, kunnes mallin tarkkuus oli riittävällä tasolla.

### 5.2.1 Videoiden esiprosessointi

Videoiden esiprosessoinnissa opetusaineiston videoita muokattiin koneoppimismallin opetukseen sopivaan muotoon. Tässä diplomityössä jokainen opetusaineiston video käsiteltiin yksi kerrallaan. Jokaisen opetusvideon kesto oli 5 sekuntia, ja jokaisesta videosta otettiin talteen kaksi kuvakehystä sekunnissa, eli yhteensä 10 kuvakehystä. Käsiteltävien kuvakehysten koko muutettiin alkuperäisestä 640x840 pikseliä kokoon 224x224 pikseliä. Kuvakoon muuttamisen jälkeen jokainen kuva muutettiin harmaakuvaksi. Kuvakoon pienentämisellä sekä muuttamalla värikuva harmaakuvaksi saatiin vähennettyä havainnointipikselien määrää, mikä pienensi kuvien käsittelyyn tarvittavaa tietokoneen laskentatehoa.

Videon kuvakehysten esiprosessoinnin jälkeen jokaisen videon kuvakehykset jaettiin videon nimen perusteella joko väkivaltaiseksi tai ei-väkivaltaiseksi. Kuvakehykset ja niitä vastaavat luokkatiedot tallennettiin tietorakenteisiin. Kun kaikki videot oli käyty läpi ja kaikkien videoiden kuvakehykset ja luokkatiedot oli tallennettu tietorakenteisiin, tietorakenteet muutettiin vielä koneoppimismallin opetukseen sopivaan muotoon.

### 5.2.2 Mallin kehitys

Tässä työssä koneoppimismalliksi päädyttiin toteuttamaan konvoluutioneuroverkko, joka on samankaltainen kuin Hansonin ja kollegoidensa [7] koneoppimismalli ilman LSTM-kerrosta. Tässä työssä toteutettava konvoluutioneuroverkko muistutti paljon VGG13 konvoluutioneuroverkkoa. Konvoluutioneuroverkossa konvoluutiokerrosten jälkeen malliin lisättiin täysin kytkettyjä (dense) kerroksia, joissa suoritettiin varsinainen luokittelu sisällön väkivaltaisuudesta. Tämän tyyppiseen koneoppimismalliin päädyttiin, koska konvoluutioneuroverkoilla on viime aikoina päästy erittäin hyviin tarkkuuksiin kuvantunnistuksessa.

Aluksi toteutettiin VGG13:sta tapainen konvoluutioneuroverkko, johon valittiin iterointikierrosten lopuksi sopivat painot ja hyperparametrit. Edeltävissä tutkimuksissa käytettiin valmiiksi opetettua VGG13-konvoluutioneuroverkkoa, mutta tässä työssä päädyttiin itse



toteuttamaan konvoluutioneuroverkko valmiiksi opetetun koneoppimismallin käyttämisen sijaan. Konvoluutiokerrosten jälkeen lisättiin täysin kytkettyjä kerroksia saman tavalla kuin Hanson ja kollegat [7] olivat omaan malliinsa lisänneet. Koneoppimismallin toteutuksen jälkeen sitä alettiin opettaa ja testata, miten hyvin opetettu koneoppimismalli tunnisti testiaineiston videoista väkivaltaa.

### 5.3 Mallin opettaminen

Mallin opettamisessa haluttiin löytää sopiva määrä konvoluutiokerroksia, ja sopivat suodintimet ja kernel-alueet kyseisille kerroksille. Koneoppimismallin opetuksessa opetustiedon erien (batch) koko oli 8 ja oppimisnopeudeksi (learning rate) valittiin 0,001. Oppimisfunktioiksi valittiin ADAM. Yhden opetuskierron aikana koko opetusaineisto käydään läpi

Mallin opetuksen aikana tarkkailtiin, että opetettavasta mallista ei tullut ylioppivaa. Mallin ylioppimista pystyttiin havaitsemaan jo opettamisen aikana, jos mallin tarkkuus oli muutamana opetuskierron (epoch) jälkeen 100 %. Tämä viittasi siihen, että opetettava malli oli ylioppinut opetusaineistoonsa eikä se pysty tunnistamaan muita kuin opetusaineistossa esiintyviä videoita. Koneoppimismallin ylioppimisen hallitsemiseksi päädyttiin kasvattamaan pudotuskerrosten (dropout layer) kokoa ja säätämään konvoluutiokerrosten suodinten ja kernel-alueiden määrää. Kun koneoppimismalli ei ollut enää ylioppiva muutosten jälkeen, niin koneoppimismallin tarkkuutta alettiin parantaa iterointikierroksilla.

### 5.4 Arkkitehtuuri

Kuten aiemmissa kappaleissa on kerrottu, konvoluutioneuroverkon konvoluutiokerrokset ovat samankaltaiset kuin VGG13 konvoluutioneuroverkossa on. Konvoluutioverkon kerrosten ja parametrien optimointiin tarvittiin useampia iterointikierroksia. Konvoluutioneuroverkon arkkitehtuuria suunniteltaessa täytyi pohtia, kuinka monta piilokerrosta tarvittiin ja kuinka paljon tarvittiin opetettavia parametreja. Liian suuret neuronimäärät kasvattivat mallin opetukseen tarvittavaa laskentatehoa. Esimerkiksi pienet konvoluutioneuroverkot ovat herkempiä virheille luokiteltaessa uusia syötteitä, kun taas suuret konvoluutioverkot ovat herkkiä ylioppimaan eivätkä enää tunnistaisi uusia syötteitä [10]. Koneoppimismallia suunniteltaessa pyrittiin optimoimaan konvoluutioneuroverkon kokoa tähän työhön sopivaksi. Konvoluutioneuroverkko pyrittiin toteuttamaan niin, että ala- ja ylätasen konvoluutiokerrokset oppisivat videoista haluttuja piirteitä.

Kuvassa 23 esitetään tässä työssä toteutettua konvoluutioneuroverkon arkkitehtuuria. Konvoluutioneuroverkko koostui konvoluutiokerroksista, joiden jälkeen tuli yhdistämiskerrokset (max pooling). Näiden jälkeen lisättiin litistyskerros (flatten). Lopuksi lisättiin pudotuskerroksia ja täysin kytkettyjä kerroksia. Viimeisen täysin kytketyn kerroksen parametrit määräytyivät tunnistettavien luokkien lukumäärän perusteella. Mallin iteroinnissa oli tarkoitus tutkia, millä kerrosmäärillä ja hyperparametreilla saatiin toteutettua tähän soveltuva luokittelijamalli.

Ensimmäisissä konvoluutiokerroksissa kerätään videokuvakehyksistä alatasen piirteitä kuten värejä, kuvien reunoja ja tunnistettavien objektien rajaviivoja. Viimeiset konvoluutiokerrokset keräävät ylätasonpiirteitä. Litistyskerroksessa muutetaan konvoluutiokerroksista saatu tulosmatriisi yhdeksi pitkäksi riviksi. Pudotuskerroksissa jätetään valitun arvon perusteella  $x$  määrä neuroneja kytkemättä jokaisella epokilla. Täysin kytketyissä kerroksissa kaikki tuloneuronit kytketään lähtö neuroneihin [38].



**Kuva 23.** Konvoluutioneuroverkon arkkitehtuuri.

## 5.5 Mallin iterointi

Tässä diplomityössä iteroinnin tavoitteena oli etsiä parasta mahdollista luokittelumallia väkivaltaisuuksien tunnistamiseen. Iterointikierrosten välissä mallin hyperparametreja muutettiin ja tarkasteltiin niiden vaikutusta mallin kokonaistarkkuuteen. Hyperparametrit ovat parametreja, joita käytetään koneoppimismallin oppimisen hallintaan [39]. Koneoppimismallia toteuttaessa hyperparametreja optimoidaan, jotta koneoppimismallin tarkkuus saadaan mahdollisimman hyväksi [39]. Tässä diplomityössä sopivia hyperparametri arvoja etsittiin iterointikierrosten aikana. Tässä työssä käytettyjä hyperparametreja olivat oppimisaste, piilokerrosten määrä, aktivointifunktiot, opetuskierrosten määrä ja suotimen sekä kernel-alueen koko.

Iterointikierrosten aikana koneoppimismallia opetettiin samalla tietoaaineistolla käyttäen sklearn-kirjaston `train_test_split`-metodia, jolla opetusaineisto jaettiin opetus- ja validointiaineistoksi. Jokaisella iterointikierroksella koneoppimismallia opetettiin useampaan kertaan, koska jokaisella uudella opetuksella metodi jakoi eri videoita opetus- ja validointiaineistoiksi. Näin pystyttiin havaitsemaan, vaikuttaako opetus- ja validointivideoiden vaihtuvuus mallin tarkkuuteen. Jokaista opetettua koneoppimismallia testattiin samalla testiaineistolla.

Iterointikierrosten välillä havaitun koneoppimismallin kokonaistarkkuuden muutoksen perusteella hyperparametreja muutettiin seuraavaa iterointikierrosta varten. Alla olevassa taulukossa 2 on esitetty, minkälaisia hyperparametrimuutoksia malliin tehtiin iterointikierrosten välissä. Koneoppimismallin kehitys iteroimalla osoittautui hyväksi menetelmäksi, sillä jokaisen iterointikierroksen jälkeen voitiin todeta, paraniko mallin kokonaistarkkuus suhteessa aikaisempaan malliin. Lisäksi iterointikierrosten aikana saatettiin havaita koneoppimismallin ylioppimista, jos konvoluutiokerroksissa oli liian suuret suotimet ja kernel-alueet. Iterointikierrosten aikana havaittiin ylioppimista, jos koneoppimismallin opetettavien parametrien määrä oli suurempi kuin opetusaineiston havainnointi pikselien kokonaismäärä.

**Taulukko 2 Mallien Iterointikierrosten muutoksia.**

<b>Mallin nro:</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>1. Konvoluutio-kerros 2D</b>	64 <sup>a</sup> , (1x1) <sup>b</sup>	32 <sup>a</sup> , (1x1) <sup>b</sup>	64 <sup>a</sup> , (1x1) <sup>b</sup>	64 <sup>a</sup> , (1x1)	32 <sup>a</sup> , (1x1) <sup>b</sup>	32 <sup>a</sup> , (1x1) <sup>b</sup>	32 <sup>a</sup> , (1x1) <sup>b</sup>	32 <sup>a</sup> , (1x1) <sup>b</sup>	32 <sup>a</sup> , (1x1) <sup>b</sup>	32 <sup>a</sup> , (1x1) <sup>b</sup>
<b>1. Yhdistämiskerros (Max pooling)</b>	2x2	2x2	2x2	2x2	2x2	2x2	2x2	2x2	2x2	2x2
<b>2. Konvoluutio-kerros 2D</b>	128 <sup>a</sup> , (1x1) <sup>b</sup>	64 <sup>a</sup> , (1x1) <sup>b</sup>	128 <sup>a</sup> , (1x1) <sup>b</sup>	128 <sup>a</sup> , (1x1) <sup>b</sup>	64 <sup>a</sup> , (1x1) <sup>b</sup>	64 <sup>a</sup> , (1x1) <sup>b</sup>	64 <sup>a</sup> , (1x1) <sup>b</sup>	64 <sup>a</sup> , (1x1) <sup>b</sup>	64 <sup>a</sup> , (1x1) <sup>b</sup>	64 <sup>a</sup> , (1x1) <sup>b</sup>
<b>2. Yhdistämiskerros (Max pooling)</b>	2x2	2x2	2x2	2x2	2x2	2x2	2x2	2x2	2x2	2x2
<b>3. Konvoluutio-kerros 2D</b>	256 <sup>a</sup> , (3x3) <sup>b</sup>	128 <sup>a</sup> , (3x3) <sup>b</sup>	256 <sup>a</sup> , (3x3) <sup>b</sup>	256 <sup>a</sup> , (3x3) <sup>b</sup>	128 <sup>a</sup> , (3x3) <sup>b</sup>	128 <sup>a</sup> , (3x3) <sup>b</sup>	128 <sup>a</sup> , (3x3) <sup>b</sup>	128 <sup>a</sup> , (3x3) <sup>b</sup>	128 <sup>a</sup> , (3x3) <sup>b</sup>	128 <sup>a</sup> , (3x3) <sup>b</sup>
<b>3. Yhdistämiskerros (Max pooling)</b>	2x2	2x2	2x2	2x2	2x2	2x2	2x2	2x2	2x2	2x2
<b>4. Konvoluutio-kerros 2D</b>	512 <sup>a</sup> , (5x5) <sup>b</sup>	256 <sup>a</sup> , (3x3) <sup>b</sup>	512 <sup>a</sup> , (3x3) <sup>b</sup>	512 <sup>a</sup> , (5x5) <sup>b</sup>	-	256 <sup>a</sup> , (3x3) <sup>b</sup>	256 <sup>a</sup> , (3x3) <sup>b</sup>	256 <sup>a</sup> , (3x3) <sup>b</sup>	256 <sup>a</sup> , (3x3) <sup>b</sup>	256 <sup>a</sup> , (3x3) <sup>b</sup>
<b>4. Yhdistämiskerros (Max pooling)</b>	2x2	2x2	2x2	2x2	-	2x2	2x2	2x2	2x2	2x2

<b>Mallin nro:</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>5. Konvoluutio-kerros 2D</b>	512 <sup>a</sup> , (5x5) <sup>b</sup>	512 <sup>a</sup> , (3x3) <sup>b</sup>	512 <sup>a</sup> , (3x3) <sup>b</sup>	512 <sup>a</sup> , (5x5) <sup>b</sup>	-	512 <sup>a</sup> , (3x3) <sup>b</sup>	512 <sup>a</sup> , (3x3) <sup>b</sup>	512 <sup>a</sup> , (3x3) <sup>b</sup>	512 <sup>a</sup> , (3x3) <sup>b</sup>	512 <sup>a</sup> , (3x3) <sup>b</sup>
<b>5. Yhdistämiskerros (Max pooling)</b>	2x2	2x2	2x2	2x2	-	2x2	2x2	2x2	2x2	2x2
<b>6. Konvoluutio-kerros 2D</b>	512 <sup>a</sup> , (5x5) <sup>b</sup>	512 <sup>a</sup> , (3x3) <sup>b</sup>	-	-	-	-	512 <sup>a</sup> , (3x3) <sup>b</sup>	512 <sup>a</sup> , (3x3) <sup>b</sup>	512 <sup>a</sup> , (3x3) <sup>b</sup>	512 <sup>a</sup> , (3x3) <sup>b</sup>
<b>6. Yhdistämiskerros (Max pooling)</b>	2x2	2x2	-	-	-	-	2x2	2x2	2x2	2x2
<b>Litistyskerros</b>	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä
<b>1. Pudotuskerros</b>	0.7	0.7	0.8	0.7	0.7	0.7	0.7	0.7	0.7	0.7
<b>1. Täysin kytketty-kerros</b>	4096	10 000	4096	4096	256	512	512	4096	8192	6000
<b>2. Pudotuskerros</b>	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
<b>2. Täysin kytketty-kerros</b>	4096	2	4096	4096	2	2	2	2	2	10 000
<b>3. Pudotuskerros</b>	0.5	-	0.8	0.5	-	-	-	-	-	0.7

Mallin nro:	1	2	3	4	5	6	7	8	9	10
<b>3. Täysin kytketty-kerros</b>	2	-	2	2	-	-	-	-	-	2
<sup>a</sup> suotimen koko										
<sup>b</sup> kernel-alueen koko										

## 6. MALLIN ARVIOINTI

Kuudennessa luvussa tarkastellaan ja arvioidaan toteutetun luokittelu toimivuutta ja analysoidaan sitä, kuinka hyvin luokittelumalli tunnistaa väkivaltaisia tilanteita.

### 6.1 Arviointimenetelmät

Parhaiten luokittelumalleja voidaan arvioida käyttämällä sekavuusmatriisia (confusion matrix). Matriisin avulla voidaan laskea luokittelun tarkkuus laskemalla, kuinka monta oikeaa ja väärää tunnistusta luokittelija on tunnistanut. Luokittelijan tarkkuutta voidaan mitata neljällä eri tavalla, joita ovat todelliset positiiviset TP (True Positive), väärät positiiviset FP (False Positive), todelliset negatiiviset TN (True Negative) ja väärät negatiiviset FN (False Negative) [40]. Tässä työssä positiiviset tapaukset ovat väkivaltaista sisältöä ja negatiiviset tapaukset ovat ei-väkivaltaista käyttäytymistä. Tässä työssä raja-arvoksi (threshold) valittiin 0.5, ja sen perusteella sekavuusmatriisin arvot määriteltiin seuraavasti:

- Todelliset positiiviset ovat tapauksia, kun malli on luokitellut väkivaltaisen sisällön väkivaltaiseksi.
- Väärät positiiviset ovat tapauksia, kun malli on luokitellut ei-väkivaltaisen sisällön väkivaltaiseksi
- Todelliset negatiiviset ovat niitä tapauksia, kun malli on luokitellut ei-väkivaltaisen sisällön ei-väkivaltaiseksi.
- Väärät negatiiviset ovat tapauksia, kun malli on luokitellut väkivaltaisen sisällön ei-väkivaltaiseksi.

Sekavuusmatriisi koostuu edellä mainituista tavoista laskea luokittelijan tarkkuus. Matriisi on yleensä 2x2 kokoinen kuten taulukosta 3 nähdään.



**Taulukko 3. Sekavuusmatriisi.**

		Todellinen luokka	
		Positiivinen	Negatiivinen
Ennustettu luokka	Positiivinen	Todelliset positiiviset	Väärät positiiviset
	Negatiivinen	Väärät negatiiviset	Todelliset negatiiviset

Luokittelijan kokonaistarkkuus saadaan laskettua seuraavasti:

$$Kokonaistarkkuus = \frac{TP + TN}{TN + FP + FN + TP} \quad (5.1)$$

Kokonaistarkkuudella mitataan sitä, kuinka hyvin malli tunnistaa saamistaan syötteistä luokat oikein. Kokonaistarkkuuden lisäksi mallin tarkkuutta voidaan mitata myös muilla metriikoilla kuten saanti (recall) ja tarkkuus (precision). Tässä työssä saannilla (recall) kuvataan kuinka monta väkivaltaista tilannetta on tunnistettu oikein kaikista väkivaltaisista tilanteista [41].

$$Saanti = \frac{TP}{TP + FN} \quad (5.2)$$

Tarkkuus (precision) kuvaa sitä, miten suuri osuus väkivaltaiseksi luokitellut videot sisältävät todellisuudessa väkivaltaa [41].

$$Tarkkuus = \frac{TP}{TP + FP} \quad (5.3)$$

## 6.2 Iteroitujen mallien tulokset

Taulukossa 4 on esitetty iteroitujen mallien luokittelutuloksia testausaineistolla, tarkkuus validointiaineistolla sekä opetettujen parametrien määrä. Mallien tarkkuus validointiaineistolla oli pääasiassa yli 90 %, paitsi mallilla 1, jonka tarkkuus oli vain 45,2 %. Mallin 1 tulosten tarkempi tarkastelu osoitti, että se tunnisti ainoastaan väkivaltaiset videot oikein. Mallin 1 testinaineiston kokonaistarkkuus oli 50 %, koska se luokitteli kaikki testinaineiston videot väkivaltaisiksi. Muiden mallien kokonaistarkkuudet testiaineistolla vaihtelivat 63,9 % – 72,5 % välillä. Useimmat mallit saivat tarkkuudeksi (precision) 100 %, eli kaikki väkivaltaiseksi luokitellut videot sisälsivät todellisuudessa väkivaltaa. Useimmissa mal-leissa saanti (recall) oli alle 50 %, eli alle puolet kaikista väkivalta tilanteista tunnistettiin oikein.

**Taulukko 4. Iteroitujen mallien tulokset**

Mallin nro	Opetettujen parametrien määrä	Tarkkuus validointiaineistolla	Kokonais-tarkkuus testiaineistolla	Tarkkuus	Saanti
1	67 037 186	45,2 %	50,0 %	50,0 %	100,0 %
2	85 861 170	95,2 %	70,8 %	100,0 %	41,7 %
3	123 397 634	97,6 %	63,9 %	85,7 %	33,3 %
4	129 689 090	92,1 %	72,2 %	100,0 %	44,4 %
5	25 766 914	92,9 %	69,4 %	100,0 %	38,9 %
6	14 397 954	97,6 %	66,4 %	89,3 %	37,2 %
7	8 107 010	97,6 %	67,2 %	100,0 %	34,4 %
8	37 477 890	95,2 %	65,8 %	100,0 %	31,7 %
9	71 044 610	97,6 %	65,8 %	100,0 %	31,7 %
10	113 099 170	95,2 %	69,4 %	100,0 %	38,9 %

### 6.3 Valittu malli

Kuvassa 24 on esitetty lopullisen mallin kerrosten määrä ja tyypit. Konvoluutioneuroverkko koostuu viidestä konvoluutiokerroksesta ja jokaisen konvoluutiokerroksen jälkeen tulee yhdistämiskerros. Yhdistämiskerroksen jälkeen on litistyskerros (flatten), jonka jälkeen on pudotuskerros. Lopussa on kaksi täysin kytkettyä kerrosta, joiden jälkeen on lisätty pudotuskerrokset. Koneoppimismallin viimeisenä kerroksena on täysin kytketty kerros. Viimeisen täysin kytketyn kerroksen parametrit määräytyvät tunnistettavien luokkien lukumäärän perusteella. Taulukosta 4 näkee, että parhaimpaan tarkkuuteen pääsi malli nro 4.

Käytetyt yhdistämiskerrokset olivat maxpooling, joiden koot olivat 2x2. Täysin kytkettyjen kerrosten koko oli 4096, ja viimeisen täysin kytketyn kerroksen koko on 2. Käytettyjen pudotuskerrosten koot olivat 0,5 ja 0,7.



**Kuva 24.** Lopullisen mallin rakenne.

## 6.4 Tulokset

Lopullisen mallin valinnan ja opettamisen jälkeen se testattiin opetusaineistosta jaetulla validointiaineistolla. Validointiaineistolla mallin tarkkuudeksi saatiin 92,1 %. Mallin tarkempaa arviointia varten lopullista mallia käytettiin tunnistamaan testiaineiston videoista väkivaltaiset ja ei-väkivaltaiset videot. Testiaineiston videot syötettiin opetelulle mallille ja mallin antama luokittelu jokaiselle videon kehykselle otettiin talteen. Näin testiaineiston avulla voitiin tarkemmin tutkia, kuinka hyvin malli tunnisti oikein väkivaltaisia ja ei-väkivaltaisia videoita.

Testiaineiston tulosten perusteella mallia voitiin tarkemmin arvioida metriikoiden avulla. Alla olevassa taulukossa 5 on esitetty mallin tuottama sekavuusmatriisi testiaineistosta.

**Taulukko 5. Mallin sekavuusmatriisi testiaineistolle**

		Todellinen luokka	
		Väkivaltainen sisältö (N=180)	Ei-väkivaltainen sisältö (N = 180)
Ennustettu luokka	Väkivaltainen sisältö (N =80)	80	0
	Ei-väkivaltainen sisältö (N=280)	100	180

Taulukosta huomaa että lopullinen malli tunnisti testiaineiston videoista väkivaltaisia kehyksiä 80/180 oikein. Loput 100 väkivaltaista kehystä malli oli luokitellut väärin ei-väkivaltaisiksi kehyksiksi. Malli tunnisti oikein kaikki ei-väkivaltaisia tapahtumia sisältävät kehykset.

Koneoppimismallin kokonaistarkkuus, tarkkuus ja saanti saadaan laskettua opettamisen jälkeen mallille syötetään sellaisia videoita, joita se ei ole vielä aikaisemmin nähnyt. Taulukossa 6 on esitetty tulosmetriikoilla lasketut mallin tulokset saaduista testiaineiston tuloksista.

**Taulukko 6 Mallin tulokset testiaineistolle**

Kokonaistarkkuus	Tarkkuus	Saanti
72,2 %	100 %	44,4 %

Kerätyistä tuloksista saatiin laskettua mallin kokonaistarkkuudeksi 72,2 %, saanniksi 44,4 % ja tarkkuudeksi 100 %. Todellisten positiivisten osuus väkivaltaisista videoista oli 44,4 % ja väärin negatiivisten osuus väkivaltaisissa videoissa oli 55,6 %. Tämä tarkoittaa sitä, että malli tunnisti testiaineiston väkivaltaisista videoista noin puolet oikein. Todellisten negatiivisten osuus ei-väkivaltaisista videoista oli 100 % ja väärin positiivisten osuus oli 0 %. Tämä tarkoittaa sitä, että malli tunnisti tietoaineiston ei-väkivaltaiset videot täysin oikein. Saatujen tulosten perusteella voidaan huomata, että kun väkivaltaisuuden tunnistamisen raja-arvoksi valittiin 0,5 niin väkivaltaistapauksia tunnistettiin alle puolet. Ei-väkivaltaisia tapauksia tunnistettiin 100 %.

Lopuksi haluttiin vielä tutkia raja-arvon vaikutusta luokittelijamallin tarkkuuteen. Raja-arvo määrittelee sen arvon, jonka perusteella malli luokittelee sisältääkö kehys väkivaltaa. Raja-arvon muutos vaikuttaa koneoppimismallin luokitteluherkkyteen. Alhaisella raja-arvolla luokittelumalli luokittelee kaikki tapaukset väkivaltaisiksi, joissa tapahtuu luokittelumallin mielestä väkivaltaa. Vastaavasti korkealla raja-arvolla luokittelumalli luokittelee väkivaltaisiksi vain ne tilanteet, joissa tapahtuu luokittelijamallin mielestä selvästi väkivaltaa. Valitun mallin alkuperäiset tulokset on tehty raja-arvolla 0,5. Alla olevassa taulukossa 7, on esitetty valitun mallin luokittelutuloksia.

**Taulukko 7 Mallin tulokset testiaineistolla eri raja-arvoilla.**

Raja-arvo	0,2	0,3	0,4	0,5	0,6	0,7	0,8
<b>Kokonais-tarkkuus</b>	50,0 %	48,9 %	78,3 %	72,2 %	69,4 %	66,4 %	51,1 %
<b>Tarkkuus</b>	50,0 %	49,4 %	84,5 %	100 %	100 %	100 %	100 %
<b>Saanti</b>	100 %	97,8 %	69,4 %	44,4 %	38,9 %	32,8 %	2,2 %
<b>Todelliset positiiviset (TP)</b>	180	176	125	80	70	59	4
<b>Väärät positiiviset (FP)</b>	180	180	23	0	0	0	0
<b>Väärät negatiiviset (FN)</b>	0	4	55	100	110	121	176
<b>Todelliset negatiiviset (TN)</b>	0	0	157	180	180	180	180

Taulukosta 7 nähdään, että raja-arvolla 0,4 saatiin paras kokonaistarkkuus. Tuloksista nähdään, että alemmilla raja-arvoilla 0,2 ja 0,3 malli on luokitellut kaikki tapaukset väkivaltaisiksi. Se on vaikuttanut siihen, että malli on luokitellut kaikki ei väkivaltaiset kehykset väärin. Suuremmilla raja-arvoilla (0,6, 0,7 ja 0,8) mallin luokittelija kokonaistarkkuus laskee lähemmäksi 50 %. Tämä johtuu siitä, että korkealla raja-arvolla malli luokittelee oikein vain ne tilanteet, joista se tunnistaa selvästi väkivaltaa sekä ei-väkivaltaa. Laske-malla raja-arvon 0,5:stä 0,4:n kokonaistarkkuus parani 6,1 %. Raja-arvon alentaminen vaikutti siihen, että osa ei-väkivaltaisista kehyksistä luokiteltiin väärin väkivaltaisiksi.

## 6.5 Tulosten vertailu aiempiin tutkimuksiin

Alla olevassa taulukossa 8 on listattu lopullinen malli, sekä toisessa kappaleessa esiteltyjen tutkimuksien mallien tuloksia, kun niitä on testattu opetusaineistosta jaettavalla validointiaineistolla.

**Taulukko 8. Mallien tulosten vertailu**

Malli	Mallin rakenne	Tietoaineisto	Tarkkuus opetusaineistolla
Diplomityön malli	Konvoluutioneuroverkko	UBI Fights:sta muodostettu aineisto.	92,1 %
Hanson kollegooneen [7]	Konvoluutioneuroverkko + kaksisuuntainen LSTM	Jääkiekko-aineisto Elokuva-aineisto Väkivaltaiset kehykset	96,9 % 100 % 92,2 %
Ullah kollegooneen [10]	3D konvoluutioneuroverkko	Jääkiekko-aineisto Elokuva-aineisto Väkivaltaiset kehykset	96 % 99,9 % 98,0 %
Asad kollegooneen [11]	Spatiaalisten tietojen kerääminen + konvoluutioneuroverkko	Jääkiekko-aineisto Elokuva-aineisto Väkivaltaiset kehykset Behave aineisto	98,8 % 99,1 % 97,1 % 95,9 %

Taulukosta huomaa, että jokaisessa mallissa on käytetty konvoluutioneuroverkkoa. Kuten kappaleessa 2 kuvatuissa tutkimuksissa käytettiin valmiiksi opetettua konvoluutioneuroverkkoa, kun taas tässä diplomityössä kehitettiin verkosta löytyvän esimerkkilähdekoodin perusteella itse konvoluutioneuroverkko. Tuloksista huomaa, että aiempien tutkimuksien mallit eivät tulosten osalta merkittävästi eroa toisistaan. Tässä työssä tehty mallin tarkkuus eroaa hieman muista malleista. Tässä työssä ei käytettyä valmiiksi opetettua konvoluutioneuroverkkoa toisin kuin edeltävissä tutkimuksissa käytettiin. Lisäksi muut tietoaaineistot saattoivat sisältää korkealaatuisempaa kuvanlaatua. Edeltävissä tutkimuksissa käytetyt tietoaaineistot sisälsivät yksinkertaisempia tapauksia kuin tässä työssä käytetty UBI Fights tietoaaineisto. Tietoaaineisto sisälsi hyvin monipuolisia tapauksia niin väkivallasta kuin ei-väkivaltaisista tilanteista, joita haluttiin mallin oppivan. Huonona puolena oli, että UBI Fights tietoaaineiston kaikkien videoiden kuvanlaatu ei ollut korkea. Tässä työssä toteutettu malli osuu tarkkuudeltaan samaan suuruusluokkaan edeltävien tutkimusten kanssa.

## 6.6 Opetusaineiston arviointi

Diplomityössä käytetty UBI Fights tietoaaineisto oli verkosta löydettyistä tietoaaineistoista monipuolisin sisältäen videoita, joissa tapahtui väkivaltaa kaduilla ja sisätiloissa. Opetusaineisto oli itsessään kohtalaisen laaja, se sisälsi yli tuhat videota eri tilanteista ja paikoista. Opetusaineiston hyviä puolia olivat monipuoliset videot. Videoita oli kuvattu paljon erilaisista tilanteista, mikä auttoi mallia oppimaan enemmän erilaisia tilanteita.

Opetusaineiston huonoja puolia olivat huonolaatuisten videoiden määrä suhteessa hyvälaatuisiin videoihin. Valitettavasti tämän vuoksi hyvin suurta osaa videoista ei voitu käyttää koneoppimismallin opetuksessa ja testauksessa. Hylätyt videot olivat joko liian epäselviä, tai niissä tapahtui sellaisia tilanteita, jotka eivät kuuluneet toteutetun koneoppimismallin tutkimuspiiriin. Mikäli opetusaineistossa olevien videoiden tarkkuus olisi ollut parempi, niin koneoppimismallin tulokset olisivat voineet olleet paremmat, sillä huonolaatuisista kuvista on koneoppimismallin vaikeampaa tehdä luokittelua kuin korkealaatuisista videoista. Mallin opetuksen aikana havaittiin, että korkealaatuisia videoita oli selvästi liian vähän. Tarkempaa koneoppimismallia varten olisi tarvittu enemmän opetusaineistoa.

Lisäksi videoiden kuvauskulma oli huono tämän työn tarkoitukseen, sillä suurin osa videoista oli kuvattu ihmisten korkeudelta noin 1,5 metristä. Valvontakamerat ovat usein korkeammalla, jolloin niiden kuvausalue on laajempi. 1,5 metrin korkeudelta kuvatuissa videoissa yksi ihminen voi peittää hyvin paljon kuvasta, sillä osa videoista oli kuvattu hyvin läheltä.

## 7. YHTEENVETO

Tässä diplomityössä suunniteltiin ja kehitettiin koneoppimismalli, joka pystyi tunnistamaan sisälsikö, video väkivaltaa vai ei-väkivaltaista sisältöä. Aluksi diplomityössä esiteltiin muutama edeltävä tutkimus, joissa oltiin tutkittu erityyppisten koneoppimismallien soveltuvuutta väkivallan tunnistamiseen videoista. Tämän jälkeen työssä käsiteltiin koneoppimisen, neuroverkkojen ja konvoluutioneuroverkkojen teoriaa. Teoriaosuuden jälkeen esiteltiin työssä käytettävä tietoaaineisto, ja miksi juuri kyseinen tietoaaineisto valittiin työn opetus ja- testausaineistoksi. Tietoaaineiston esittelyn jälkeen esiteltiin toteutetun koneoppimismallin toteutusvaiheet, ja miten tekoälymalli toteutettiin. Lopuksi esiteltiin lopullisen mallin tulokset testausaineistolla testattaessa. Lopullisen mallin tarkkuutta saatiin parannettua muuttamalla mallin raja-arvoa.

Diplomityössä saaduista tuloksista voidaan huomata, että toteutetun koneoppimismallin tarkkuus opetusaineistolla on samalla tasolla kuin edeltävissä tutkimuksissa esitetyjen mallien. Koneoppimismallin tarkkuus oli selvästi huonompi mallia testattaessa sellaisilla videoilla, jotka eivät kuuluneet sen opetusaineistoon. Tämä johtunee siitä, että koneoppimismallia opettaessa ohjatulla oppimisella koneoppimismalli oppii vain tunnistamaan sellaisia tilanteita ja tapauksia, joita esiintyy käytetyssä opetusaineistossa. Ohjatulla opettamisella opetetun koneoppimismallin ei voida olettaa tunnistavan sellaisia tapauksia, joita ei kuulu käytettyyn opetusaineistoon.

Malli osasi tunnistaa käytetystä testiaineistosta kaikki ei-väkivaltaiset kehykset oikein, mutta väkivaltaisia kehyksiä tunnistettiin vain noin puolet oikein. Raja-arvoa muuttamalla noin 70 % väkivaltaisia kehyksiä tunnistettiin oikein. Varmuudella ei voida todeta, kuinka paljon hyötyä koneoppimismallista on automatisoimaan väkivaltaisuuden tunnistamista.

Jatkokehityksenä olisi kerätä paljon enemmän korkealaatuisia videoita. Ihanteellinen tilanne olisi saada suoraan videovalvontakameralla kuvattuja korkealaatuisia videoita väkivaltaisista ja ei-väkivaltaisista tapauksista. Tällöin voitaisiin käyttää videokameran kuvaamaa materiaalia koneoppimismallin opetuksessa. Toteutetun koneoppimismallin tarkkuus kärsi tietoaaineiston puutteellisuuden takia. Lisäksi koneoppimismallista voisi toteuttaa syvemmän, tai suuremmalla tietoaaineistolla opettaessa lisätä LSTM-kerros.

Internetistä löytyvät tietoaaineistot näyttivät soveltuvan hyvin tutkimukseen. Pohdittavaksi jää, kuinka hyvin tällaiset tietoaaineistot soveltuvat sellaisten koneoppimismallien toteutuksiin, joiden tulisi toimia oikeassa ympäristössä. Ainakin tässä diplomityössä havaittiin, että tietoaaineiston heikkolaatuisuuden ja videoiden vähäisyys vaikutti mallin tarkkuuteen.



## LÄHTEET

- [1] G. Van Voorthuijsen, H. Van Hoof, M. Klima, K. Roubik, M. Bernas, and P. Pata, "CCTV effectiveness study," *Proc. 39th Annu. 2005 Int. Carnahan Conf. Secur. Technol.*, pp. 105–108, 2005.
- [2] M. T. Nguyen, L. H. Truong, T. T. Tran, and C.-F. Chien, "Artificial intelligence based data processing algorithm for video surveillance to empower industry 3.5," *Comput. Ind. Eng.*, vol. 148, p. 106671, 2020.
- [3] M. Smolaks, "Research: AI-powered video analytics market set to reach \$4.5 billion by 2025 - AI Business," 2020. [https://aibusiness.com/document.asp?doc\\_id=761172](https://aibusiness.com/document.asp?doc_id=761172) (accessed Jan. 20, 2021).
- [4] A. Choudhary and S. Chaudhury, "Video analytics revisited," *IET Comput. Vis.*, 2016, doi: 10.1049/iet-cvi.2015.0321.
- [5] A. Ramchandran and A. K. Sangaiah, "Unsupervised deep learning system for local anomaly event detection in crowded scenes," *Multimed. Tools Appl.*, 2020, doi: 10.1007/s11042-019-7702-5.
- [6] A. Balasundaram and C. Chellappan, "An intelligent video analytics model for abnormal event detection in online surveillance video," *J. real-time image Process.*, vol. 17, no. 4, pp. 915–930, 2018.
- [7] A. Hanson, K. PNVR, S. Krishnagopal, and L. Davis, "Bidirectional Convolutional LSTM for the Detection of Violence in Videos," *Comput. Vis. – ECCV 2018 Work. 2019-01-29, p.280-295*, pp. 1–17, 2018.
- [8] E. Bermejo Nievas, O. Deniz, G. Bueno, and R. Sukthankar, "Violence detection in video using computer vision techniques," 2011.
- [9] T. Hassner, Y. Itcher, and O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behavior," *Proc. 2012 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 1–6, 2012.
- [10] F. U. M. Ullah, A. Ullah, K. Muhammad, I. U. Haq, and S. W. Baik, "Violence Detection Using Spatiotemporal Features with 3D Convolutional Neural Network," *Sensors (Basel, Switzerland)*, 2019-05-30, Vol.19 (11), p.2472, pp. 1–15, 2019.
- [11] A. Mujtaba, Y. Jie, H. Jiang, S. Pourya, and H. Xiangjian, "Multi-frame feature-fusion-based model for violence detection," *Vis. Comput.* 371415–1431, pp. 1–17,

2020.

- [12] S. . Blunsden and R. . Fisher, "The BEHAVE video dataset: ground truthed video for multi-person behavior classification," *Ann. BMVA*, vol. 4, pp. 1–12, 2010.
- [13] E. Alpaydin, "Introduction to Machine Learning," Third edit., MIT Press, 2014, p. 3.
- [14] K. P. Murphy, "Machine learning a probabilistic perspective," MIT Press, 2012, pp. 2–11.
- [15] M. Pietikäinen and O. Silvén, "Tekoälyn haasteet - koneoppimisesta ja konenäöstä tunnetekoälyyn," 2019, pp. 60, 69, 91–92.
- [16] O. Chapelle, B. Schölkopf, and A. Zien, "Semi-Supervised Learning," MIT Press, 2006, p. 1.
- [17] M. Kubat, "An Introduction to Machine Learning," Springer, 2017, pp. 273–275, 104–105.
- [18] W. Ertel, "Introduction to Artificial Intelligence (Undergraduate Topics in Computer Science)," Springer, 2017, pp. 289–292.
- [19] N. Yadav, A. Yadav, and M. Kumar, "An Introduction to Neural Network Methods for Differential Equations," Springer, 2015, pp. 19–21.
- [20] J. Patterson and A. Gibson, *Deep learning: A Practionar Approach*. O'Reilly, 2017.
- [21] S. Skansi, "Introduction to Deep Learning," Springer, 2018, pp. 79–81.
- [22] A. Menshawy, "Deep Learning By Example," Packt, 2018, p. 129.
- [23] C. Giuseppe, "Keras 2.x Projects," Packt, 2018.
- [24] M. Osvaldo, "Bayesian Analysis with Python," Packt, 2016.
- [25] K.-L. Du and M. N. S. Swamy, "Neural Networks and Statistical Learning," Springer, 2014, p. 7,10.
- [26] A. Kattan, R. Abdullah, and Z. W. Geem, "Artificial Neural Network Training and Software Implementation Techniques," Nova Science Publishers, 2011, pp. 17–19.
- [27] S. Gerrish and K. Scott, *How smart machines think*. The MIT Press, 2018.
- [28] B. Bateman, A. R. Jha, B. Johnston, and I. Mathur, "The supervised learning workshop: a new interactive approach to understanding supervised learning algorithms," Packt Publishing, 2020, pp. 140–142.

- [29] J. Lever, M. Krzywinski, and N. Altman, "Points of Significance: Model selection and overfitting," *Nat. methods*, 2016-09-01, Vol.13 (9), p.703, vol. 13, pp. 1–2, 2016.
- [30] P. Kim, "MATLAB Deep Learning With Machine Learning, Neural Networks and Artificial Intelligence," Apress, 2017.
- [31] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," *MIT Press*, 2016, [Online]. Available: <https://www.deeplearningbook.org/>.
- [32] C. C. Aggarwal, "Neural Networks and Deep Learning," Springer, 2018, pp. 5, 53–54.
- [33] M. Vatsa, R. Singh, and A. Majumdar, "Deep Learning in Biometrics," CRC Press, 2018, pp. 14–16.
- [34] H. Xianghong, C. Taosheng, H. Huang, L. Zihao, L. Xueming, and X. Xiaoming, "Efficient field-programmable gate array-based reconfigurable accelerator for deep convolution neural network," *Electron. Lett.*, vol. 57, p. 1, 2021.
- [35] G. Aurélien, "Neural networks and deep learning," O'Reilly Media, 2018.
- [36] J. P. Mueller and L. Massaron, "Deep learning," 1st ed., For Dummies, 2019.
- [37] B. Degardin and H. Proença, "Human Activity Analysis: Iterative Weak/Self-Supervised Learning Frameworks for Detecting Abnormal Events," *2020 IEEE Int. Jt. Conf. Biometrics*, pp. 1–7, 2020.
- [38] "Keras Documentation." <https://keras.io/>.
- [39] K. Nyuytiybiy, "Parameters and Hyperparameters in Machine Learning and Deep Learning," *Towards Data Science*, 2020. <https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac>.
- [40] V. Kotu and B. Deshpande, "Predictive analytics and data mining : concepts and practice with rapidminer," 1st editio., Elsevier Inc, 2015, pp. 258–259.
- [41] R. Eberhart and Y. Shi, "Computational intelligence concepts to implementations," 1st editio., Elsevier/Morgan Kaufmann Publishers, 2007, pp. 407–410.