

Leena Hellman

**Tutkasensorin ja syväoppimisen hyödyntäminen
liikkeen tunnistuksessa**

Tietotekniikan
pro gradu -tutkielma
5. kesäkuuta 2023

Jyväskylän yliopisto
Informaatioteknologian tiedekunta
Kokkolan yliopistokeskus Chydenius

Tekijä: Leena Hellman

Yhteystiedot: leena.m.hellman@student.jyu.fi

Puhelinnumero: -

Ohjaaja: Ismo Hakala

Työn nimi: Tutkasensorin ja syväoppimisen hyödyntäminen liikkeen tunnistuksessa

Title in English: Human activity recognition with radar sensor and deep learning

Työ: Tietotekniikan pro gradu -tutkielma

Sivumäärä: 80

Tiivistelmä: Ihmisten liikkeiden tunnistusta käytetään erilaisissa valvontaan ja älykoteihin liittyvissä ratkaisuissa. Liikkeen tunnistukseen voidaan käyttää erilaisia sensoreita. Viime aikoina erityisesti tutkasensorilla tapahtuva liikkeen tunnistus on ollut mielenkiinnon kohteena. Tutka toimii valaistusolosuhteista riippumatta, havaitsee liikkeitä seinien läpi eikä se loukkaa ihmisen yksityisyydensuojaa kuten kamera. Tutka lähettää elektromagneettista aaltoa, joka heijastuu kohteesta takaisin. Tästä palanneesta signaalista voidaan saada erilaista tietoa esimerkiksi kohteen nopeudesta, etäisyydestä ja kulmasta suhteessa tutkaan. Syväoppiminen on tuonut uusia mahdollisuuksia käsitellä suuria määriä dataa ja tehdä aineiston luokitusta ilman manuaalista aineiston läpikäyntiä. Tutkan tuottamasta datasta voidaan muodostaa kuvamuotoista dataa, joka sopii erityisen hyvin konvoluutioneuroverkkoa hyödyntäville syväoppimismalleille. Tässä työssä tutustutaan millimetriaalto-tutkan ja syväoppimisen mahdollisuuksiin liikkeiden tunnistuksessa käyttäen vapaasti ladattavissa olevaa PARrad-aineistoa, johon on koottuna yhteensä 14 eri liikettä 24 henkilön suorittamana ja useissa eri mittaustilanteissa kahden eri tutkan havainnoimana. Työssä implementoidaan kolme erilaista konvoluutioneuroverkkomallia ja vertaillaan näiden suorituskykyä. Lisäksi työ tarjoaa näkökulman hyödyntää TinyML-konseptia mallin optimoinnissa ja siirtämisessä langattomaan sensoriverkkoon mikrokontrollerin käyttöön. Malleista parhaimmaksi osoittautui malli, joka oli opetettu käyttäen data-augmentaatiolla lisättyä aineistoa. Virheettömyydeksi saatiin parhaimmalla mallilla 87%:n tulos.

Avainsanat: langaton sensoriverkko, tutka, syväoppiminen, liikkeentunnistus, konvoluutioneuroverkko

Abstract: Human activity recognition has many potential use cases, such as surveillance, healthcare and different smart home applications. There are many different sensors which can be used to detect movements and activity. Especially radar sen-

sors have been studied recently because they have many benefits compared to other types of sensors. For example, radar doesn't need any special light conditions, it can detect movements through walls and it doesn't have privacy issues. Deep learning is a way to analyse and label big amounts of data. Radar data can be processed as images and that is why convolutional neural network as a deep learning method suits well with it. This work presents activity recognition using millimeter wave radar and deep learning. The empirical part of the work has been done using the free access PARrad database, which consists of many data sets with overall 24 persons performing 14 different movements. Overall three convolutional models has been implemented and tested. Best results got the model which had been trained with the augmented data, accuracy being 87%. This work also represents the conversion of the best model to a smaller TinyML model which can be used directly in a wireless sensor network end device.

Keywords: wireless sensor network, radar, deep learning, convolutional neural network

Copyright © 2023 Leena Hellman

All rights reserved.

Sanasto

Adam	Adaptive Moment Estimation
ADC	Analog-to-Digital Converter
AE	Autoencoder
CNN	Convolutional Neural Network
DAC	Digital-to-Analog Converter
ELU	Exponential Linear Unit
FFT	Fast Fourier transform
FMCW	Frequency Modified Continuous Wave
GHz	Gigahertz
HRP	High Resolution Range Profile
LSTM	Long Short-Term Memory
ML	Machine Learning
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
TinyML	Tiny Machine Learning
WSN	Wireless Sensor Network

Sisällys

Sanasto	i
1 Johdanto	1
2 Liikkeiden tunnistus langattomissa sensoriverkoissa	3
2.1 Langaton sensoriverkko ja sensorit	3
2.2 Liikkeiden tunnistukseen käytettäviä sensoreita	5
2.2.1 Akustiset sensorit	6
2.2.2 Kapasitiiviset ja elektrostaattiset sensorit	6
2.2.3 Mekaaniset sensorit	7
2.2.4 Optiset sensorit	7
2.2.5 Elektromagneettiset sensorit	8
2.2.6 Hybridiratkaisut	9
2.3 Vertailua eri sensoreiden välillä	9
3 Tutka	11
3.1 Yleistä tutkasta	11
3.2 Tutkasta saatava data	12
3.3 Tutkan käyttämät taajuudet	13
3.4 Millimetriaaltotutka	14
3.4.1 Millimetriaaltotutkan käyttökohteet ja taajuudet	15
3.5 Tutkan signaalin prosessointi	17
4 Syväoppiminen	25
4.1 Yleistä syväoppimisestä	25
4.2 Konvoluutioneuroverkko	28
4.2.1 Konvoluutioverkon rakenne	28
4.2.2 Konvoluutioneuroverkon opettaminen ja optimointi	32
4.3 Syväoppimismallin suorituskyvyn analysoinnista	35

5	Liikkeiden tunnistus tutkan ja syväoppimisen avulla	38
5.1	Syväoppimismallit tutkan 3D-muotoisen datan kanssa	40
5.2	Syväoppimismallit tutkan 2D-muotoisen datan kanssa	41
5.3	Yksiulotteisen datan käyttö syväoppimisen keinoin	42
5.4	Hybridiratkaisuja liikkeen tunnistukseen	42
5.5	Yhteenveto liikkeiden tunnistuksesta tutkan ja syväoppimisen avulla	43
6	Tutkimus	45
6.1	Tutkimuksen taustaa	45
6.2	Tutkimuksen tavoitteet	48
6.3	Tutkimuksen toteutus	48
6.3.1	Datan esikäsittely	50
6.3.2	Datan luokittelu ja jakautuminen	53
6.3.3	Syväoppimismallin kehittäminen	54
6.3.4	Työssä käytetyt ohjelmistot ja laitteisto	57
7	Tutkimuksen tulokset ja analyysi	58
7.1	Mallin A testitulokset	59
7.2	Mallin B tulokset	62
7.3	Mallin C tulokset	64
7.4	Mallin siirtäminen langattomaan sensoriverkkoon	67
7.5	Tulosten vertailua alkuperäiseen tutkimukseen	70
7.6	Pohdintaa saaduista tuloksista ja jatkokehitysideoita	71
8	Yhteenveto	74
	Lähteet	75

1 Johdanto

Ihmisten liikkeiden ja aktiivisuuden tunnistus on kehittynyt paljon viime vuosina ja sitä on käytetty esimerkiksi valvontaan, älykotiratkaisuissa, videoiden analysoinnissa, autopiloteissa sekä ihmisen ja koneen välisessä vuorovaikutuksessa [31]. Liikkeiden tunnistukseen voidaan käyttää erilaisia sensoreita (*eng. sensors*). Fu et al. [14] jakavat nämä sensorit viiteen ryhmään: akustisiin, sähköisiin, optisiin, mekaanisiin ja elektromagneettisiin sekä näiden yhdistelmiin eli hybridisensoreihin. Sensorit toimivat linkkinä fyysikaalisen ja digitaalisen maailman välillä havaitsemalla fyysikaalisia ilmiöitä ja muokkaamalla näitä digitaaliseen muotoon, jotta ne voidaan tallentaa, säilyttää ja jotta niihin voidaan reagoida ja vaikuttaa [10].

Erytisesti tutkalla tapahtuva liikkeiden seuranta on ollut tutkimusten mielenkiinnon kohteena viime aikoina. Tutka lähettää elektromagneettisen aallon ja vastaanottaa kohteesta takaisin heijastuneen signaalin, jolloin voidaan tehdä päätelmiä esimerkiksi kohteen etäisyydestä, kulmasta ja nopeudesta [44]. Tutkasensori toimii valaistuksesta ja ympäristön olosuhteista huolimatta, se ei loukkaa yksityisyyttä, se voi havaita seinänkin läpi liikkeitä eikä sitä tarvitse kiinnittää vartalolle liikkeiden havaitsemiseksi [31].

Syväoppiminen (*eng. deep learning*) on tuonut uusia ulottuvuuksia suurten datamäärien luokittelun ja tunnistukseen. Koneoppiminen (*eng. machine learning, ML*) tarkoittaa prosessia, jossa kone oppii saadusta aineistosta ilman, että sitä on erikseen ohjelmoitu [27]. Syväoppiminen (*eng. deep learning*) on yksi koneoppimisen muoto, jossa oppimista tapahtuu usealla eri tasolla ja josta voidaan todeta muodostuvan tietynlainen verkosto, niin kutsuttu neuroverkko [28]. Syväoppiminen pystyy tunnistamaan syviä yhteyksiä ja ominaisuuksia datassa automaattisesti ja lisäksi sillä pystytään käymään läpi hyvinkin suuria määriä dataa [31].

Konvoluutioneuroverkko (*eng. convolutional neural network*) on syväoppimismalli, joka sopii kuvamuotoisen aineiston käsittelyyn [9]. Tutkan tuottamasta aineistosta voidaan myös muodostaa kuvankaltaista dataa, joten konvoluutioneuroverkkoja on käytetty paljon liikkeen tunnistuksessa tutkan datasta. Esimerkiksi tutkimuksissa [30], [20] ja [6] toteutettiin liikkeen tunnistusta käyttäen konvoluutioneuroverkkoja syväoppimismallina. Syväoppimiseen kuuluu olennaisena osana mallin opetus

ja se tapahtuu usein hyvinkin suurilla datamäärillä ja vaatii paljon laskentatehoa ja muistia [13]. Näin ollen mallin saaminen todellisiin käyttöolosuhteisiin, kuten langattoman sensoriverkon (*eng. wireless sensor network*) käyttöön, vaatii malliin muutoksia. TinyML (*Tiny machine learning*) on konsepti, jolla syväoppimismalleja voidaan optimoida siten, että ne sopivat sensoriverkon mikrokontrollerin käyttöön [4]. Tutkimusta koskien liikkeiden tunnistusta syväoppimisen ja tutkan avulla on tehty paljon viime vuosina, mutta mallin muuntamisesta langattoman sensoriverkon käyttöön ei ole juurikaan tehty tutkimuksia.

Tässä työssä tavoitteena on luoda konvoluutioneuroverkkoa hyödyntävä syväoppimismalli, joka pystyy tunnistamaan tutkan tuottamasta datasta eri liikkeitä ja joka olisi mahdollista saada sellaiseen muotoon, että se mahtuu mikrokontrolleriin. Käytössä on PARrad-aineisto [5], jossa on kerättyä 24 henkilön suorittamia 14 liikettä 77 GHz ja 60 GHz taajuuksia käyttävien millimetriaaltotutkien havainnoimana. Työn empiirisessä osuudessa käydään läpi aineiston pilkkominen ja analysointi ennen varsinaisen syväoppimismallin kehittämistä. Tämän jälkeen kuvataan konvoluutioneuroverkkomallin luominen ja optimointi liikkeiden tunnistukseen. Työssä implementoidaan kolme erilaista konvoluutioneuroverkkomallia kummankin tutkan aineistosta sekä yhdestä data-augmentaatiolla muokatusta aineistosta. Näiden mallien tuloksia arvioidaan ja vertaillaan. Parhaimmalle mallille toteutetaan optimointi ja konversio TinyML -konseptin mukaan, jotta sitä pystytään käyttämään suoraan langattoman sensoriverkon sensorin mikrokontrollerissa. Työn teoriaosuudessa käydään läpi langattoman sensoriverkon perusteita, erilaisia sensoreita liikkeen tunnistukseen, tutkan toimintaa ja tutkan datan analysointia, syväoppimisen ja konvoluutioneuroverkon toimintaa sekä liikkeiden tunnistusta tutkan ja syväoppimisen avulla.

Luku 2 esittelee langattoman sensoriverkon käsitteen sekä liikkeiden tunnistusta erilaisten sensorien kautta. Luvussa 3 keskitytään esittelemään tutkaa ja sen toimintaa tarkemmin. Luku 4 puolestaan esittelee lyhyesti syväoppimisen käsitteenä ja tarkemmin konvoluutioneuroverkon rakennetta ja toimintaa sekä syväoppimismallin suorituskyvyn arviointia. Luvussa 5 tarkastellaan tutkan ja syväoppimisen käyttöä yhdessä liikkeiden tunnistukseen kirjallisuuskatsauksen muodossa. Työn empiirisen osuuden taustaa ja toteutusta käydään läpi luvussa 6. Luku 7 esittelee työn tuloksia, johtopäätelmiä tuloksista sekä sen, miten syväoppimismalli voidaan muuntaa mikrokontrollerin käyttöön. Viimeinen luku 8 tekee yhteenvedon työstä ja antaa suuntaviivoja, miten työstä opittua voidaan jatkossa hyödyntää.

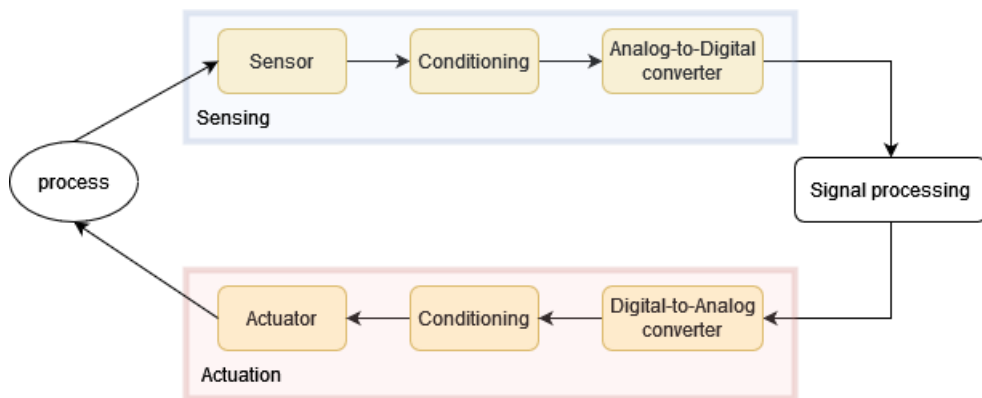
2 Liikkeiden tunnistus langattomissa sensoriverkoissa

Tässä luvussa tehdään katsaus liikkeiden tunnistukseen langattomissa sensoriverkoissa. Ensimmäinen aliluku 2.1 selittää lyhyesti sensoriverkon käsitteitä. Seuraavassa aliluvussa 2.2 käydään läpi, millaisia erilaisia sensoreita on olemassa liikkeiden ja aktiivisuuden tunnistukseen. Tämän jälkeen aliluvussa 2.3 vertaillaan eri sensoreiden ominaisuuksia liikkeiden tunnistuksessa.

2.1 Langaton sensoriverkko ja sensorit

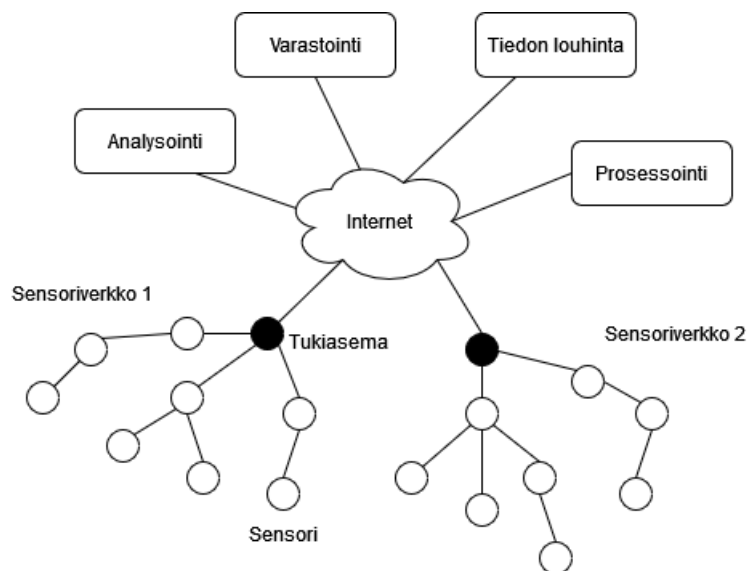
Ensimmäisessä aliluvussa tutustutaan langattomiin sensoriverkkoihin, jossa lähteenä on käytetty kirjaa [10]. Aistiminen (*eng. sensing*) tarkoittaa ympäristön informaation keräämistä joko fyysisistä objekteista tai jostakin tapahtumasta tai prosessista, kuten lämpötilan muutoksesta. Tällaista havainnointia suorittavaa toimijaa sanotaan sensoriksi tai anturiksi (*eng. sensor*). Sensoria, joka pystyy keräämään tietoa kohteesta ilman kosketusta, sanotaan etäsensoreiksi (*eng. remote sensor*). Sensorin tehtävänä on muuntaa aistimansa fyysikaalisen maailman energia sähköiseksi energiaksi, joka voidaan viedä tietokoneen tai jonkin laskevan yksikön käsiteltäväksi. Sensorin aistimaa fyysikaalista energiaa täytyy usein säädellä (*eng. conditioning*) ennen kuin signaali siirtyy eteenpäin. Joissain tapauksissa signaalia on voimistettava ja joskus taas vaimennettava ja usein tarvitaan myös kohinan poistoa. Lopulta analoginen signaali muunnetaan digitaalseksi signaaliksi ADC-muuntimen (*eng. analog-to-digital converter, ADC*) kautta. Aktuaattori (*eng. actuator*) on puolestaan sensori, joka voi suoraan vaikuttaa ympäristöönsä. Esimerkiksi lämpimän veden virtausta valvova tai oven aukaisua tai sulkemista ohjaava moottori on aktuaattori. Aktuaattorin tuottamat digitaaliset signaalit kulkevat DAC-muuntimen (*eng. digital-to-analog converter, DAC*) kautta, jolloin analogisilla signaaleilla vaikutetaan ympäristöön. Kuvassa 2.1 on havainnollistettu sensorin ja aktuaattorin toimintaa suhteessa ympäristöön.

Toisinaan sensorit saattavat olla suoraan yhteydessä prosessoivaan yksikköön, mutta useimmiten sensorit siirtävät keräämänsä datan langattoman verkon kautta keskusyksikköön. Näin ollen sensorissa on paitsi aistiva osa, sillä on myös jon-



Kuva 2.1: Kuva sensorin ja aktuaattorin toiminnasta Dargie et al.[10, s.25] mukaan.

kinlainen prosessoiva, varastoiva ja kommunikaatioon tarkoitettu osa. Kun useat sensorit tekevät yhteistyötä ja valvovat yhdessä isoa fysikaalista ympäristöä, niiden sanotaan muodostavan langattoman sensoriverkon (*wireless sensor network, WSN*). Sensorit kommunikoivat langattomien radioiden kautta toistensa kanssa sekä myös tukiaseman (*eng. base station, BS*) kanssa. Tukiaseman kautta on puolestaan yhteys Internetiin. Varsinainen datan käsittely, prosessointi, säilytys ja analysointi tapahtuvat yleensä siis muualla kuin sensoreissa. Kuvassa 2.2 on esitettyä havainnekuva langattoman sensoriverkon osista.



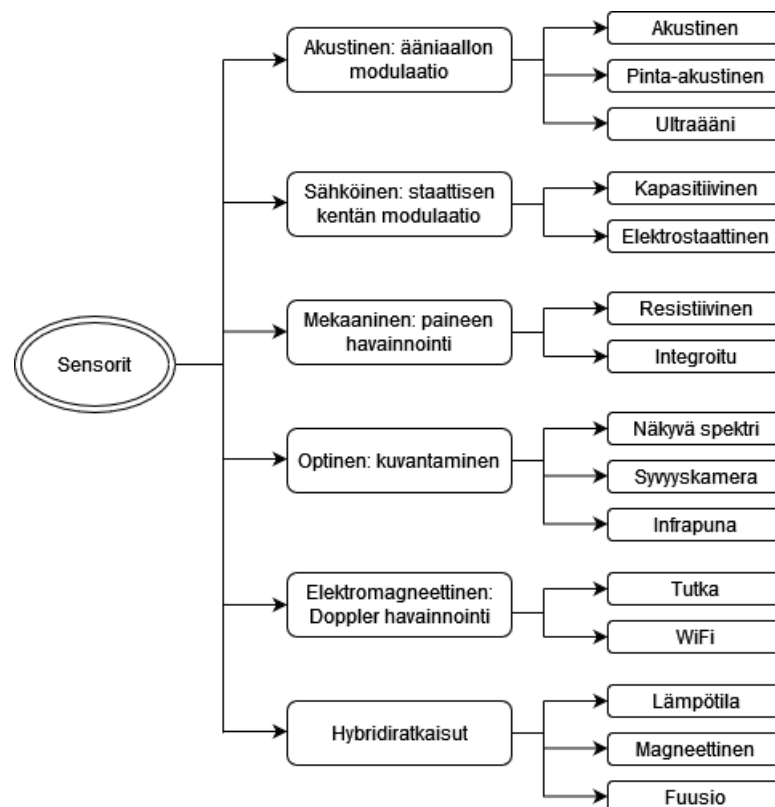
Kuva 2.2: Langattoman sensoriverkon osat ja toiminta Dargie et al.[10, s.28] mukaan.

Langattomat sensoriverkot tuovat kuitenkin omat haasteensa tietoverkon suunnittelulle. Useimmiten isoimman haasteen tuo se, että sensorit toimivat rajoitetulla

energiämäärällä, esimerkiksi pattereiden tai akun varassa, ja siksi ne tulisi suunnitella energiatehokkaiksi. Sensorit sijaitsevat usein etäällä ja karuissa olosuhteissa, jolloin niiden on selviydyttävä itsenäisesti erilaisista ongelmatilanteista ja yhteyksistä toisiin sensoreihin. Lisäksi sensorit tarvitsevat usein langattoman yhteyden, jonka toteuttaminen voi olla hankalaa, jos välimatkat tukiasemaan ovat pitkiä. Usein myös sensorit keräävät arkaluontoista tietoa ja lähettävät sitä langattomalla yhteydellä eteenpäin. Ne altistuvat siten erilaisille tietoturva-uhkille ja vaativat erityistä huomiota tämänkin osalta.

2.2 Liikkeiden tunnistukseen käytettäviä sensoreita

Ihmisen liikkeitä voidaan havainnoida monella eri tavalla käyttäen hyväksi erilaisia sensoreita. Kuvassa 2.3 on esitettyinä artikkelin [14] mukainen jako sensoreihin. Seuraavissa aliluvuissa käydään lyhyesti läpi, kuinka nämä eri sensorit toimivat ja millaisia tutkimuksia niistä on tehty.



Kuva 2.3: Sensoreiden luokittelu liikkeiden tunnistuksessa artikkelin [14, s.2] mukaan, muokattu.

2.2.1 Akustiset sensorit

Akustiset sensorit voidaan jakaa kolmeen ryhmään Fu et al. mukaan [14]: aktiivisiin akustisiin sensoreihin, pinta-akustisiin sensoreihin sekä ultraäänisensoreihin. Akustiset sensorit aistivat mekaanisia tai akustisia aaltoja, jotka kulkevat eri materiaalien läpi. Esimerkiksi seismografi on pinta-akustinen sensori, joka toimii passiivisesti havainnoiden maanpinnan värinää. Aktiiviset akustiset sensorit puolestaan pystyvät havainnoimaan ilmassa kulkevia aaltoja. Myös mikrofoni on akustinen sensori, ja sillä voidaan esimerkiksi tunnistaa yskimistä, huonekalujen siirtelyä tai oven paukuttamista, mitkä voivat auttaa myös liikkeiden tunnistuksessa. Ultraäänisensorit puolestaan ovat sensoreita, jotka aktiivisesti lähettävät ja vastaanottavat ultraäänisignaalia 20 kHz ja 200 MHz väliltä havainnoidakseen etäällä olevaa kohdetta.

Tutkimuksessa [3] tutkijat kehittivät systeemin, jossa lattiaan asennettiin passiivisesti toimivia pinta-akustisia sensoreita, jotka havaitsevat tärinää. Näiden sensoreiden avulla pystyttiin tunnistamaan kaatumisen lattialle, joka aiheutti muista liikkeistä poikkeavan värinän muodon, jossa esimerkiksi värinän taajuus, kesto ja amplitudi erosivat muissa liikkeissä. Ghosh et al. [16] toteuttivat tutkimuksessaan systeemin, jossa ultraääntä käyttävä sensori pystyi havaitsemaan ihmisen liikkeistä istumisen, seisomisen ja kaatumisen.

Fu et al. [14] toteavat kaikilla akustisilla sensoreilla olevan omat hyvät ja huonot puolensa. Äänten tarkkailu mikrofonilla voi olla hyödyllistä ajatellen jokapäiväisiä toimintoja ja askareita, mutta ne tuovat oman haasteensa, koska niistä voidaan tulkita puhetta ja siten yksityisyydensuoja on uhattuna. Kaatumista ja kävelyä on helpompi puolestaan havaita lattiaan asennetuilla sensoreilla, mutta tämä yleensä vaatii useita sensoreita, jotta saataisiin edes yksi tila katettua. Ultraääni taas voi olla hyödyllinen pienempien liikkeiden tarkkailuun, sillä niissä vaaditaan lähempää aistimista.

2.2.2 Kapasitiiviset ja elektrostaattiset sensorit

Fu et al. [14] jakavat sähkökentän ominaisuuksia hyödyntävät sensorit kahteen ryhmään: kapasitiivisiin ja elektrostaattisiin sensoreihin. Kapasitiivinen sensori käyttää hyväkseen sähkökentän voimakkuuden ja sen läheisyyden havaitsemista. Sensorin läheisyydessä olevan sähköä johtavan kohteen, kuten ihmisen, liike tai läheisyys tekee häiriön sensorin sähkökentässä. Häiriön määrä on suoraan suhteessa kohteen läheisyyteen. Toinen sähköisen sensorin muoto on elektrostaattinen sensori, joka

toimii paikallisen varauksen avulla toisin kuin kapasitiivinen sensori, joka pitää yllä jatkuvaa sähkökenttää. Elektrostaattinen havainnointi tarkoittaa sähköisen varauksen muodostamista kohteen pinnalle, jonka kosketus ja kontakti kohteen ja sensorin välillä aiheuttaa. Tämä indusoi vastakkaisen varauksen vastapuolen pintaan.

Tutkimuksessaan Valtonen et al. [55] toteuttivat lattiaan sijoitettavan systeemin, jossa lattialaattojen välistä kapasitanssia mitattiin ja pystyttiin toteuttamaan kohteen 2-ulotteinen paikannus. Systemi toimi 10 Hz:n taajuudella ja sillä saavutettiin ihmisen paikannus 15 cm tarkkuudella ja kävelevän henkilön paikannus 41 cm tarkkuudella. Usein sähköistä varausta aistivia sensoreita käytetäänkin lähellä olevan kohteen asennon tai paikalla tapahtuvan liikkeen seurantaan tai esimerkiksi sormenjäljen tunnistukseen [14].

2.2.3 Mekaaniset sensorit

Artikkelin [14] mukaan mekaaninen signaali sensorista tarkoittaa voimaa, joka kohdistuu pintaan. Pinnan muodonmuutos tuottaa sähköisen signaalin, josta saadaan näyte ADC-muuntimen kautta ja voidaan tehdä määrällinen mittaus paineesta. Artikkelissa jaetaan mekaaniset sensorit kahteen ryhmään: resistiiviset paineentunnistussensorit sekä integroidut sensorit.

Sundholm et al. [48] toteuttivat kuntosalinympäristöön systeemin, jossa mattoon sijoitetuilla painesensoreilla havaittiin erilaisia voimisteluliikkeitä, kuten vatsalihaspunnerruksia ja käsivarsipunnerruksia, ja mitattiin liikkeiden toistojen määrää. Joustavan kankaan sisään asennettiin ohut kerros sähköä johtavaa polymeerilevyä, jossa paineanturit muodostivat painetta tunnistavan verkon. Lim et al. [33] puolestaan tutkivat huonekaluihin ja lattioihin sijoitetuilla painesensoreilla erilaisia arkipäivän toimintojen tunnistamista. Tutkijat onnistuivat tunnistamaan muun muassa ruokailun, nukkumisen, fyysisen ponnistelun, ulos lähtemisen ja lepäämisen sen perusteella, kuinka paljon henkilö käyttää huonekaluja, joihin sensorit on asennettu. Fu et al. [14] mukaan painesensorit sopivat parhaiten tilanteisiin, joissa tapahtuu staattista paikallaan olevaa liikettä tai erilaisten asentojen tunnistukseen.

2.2.4 Optiset sensorit

Artikkeli [14] kertoo optisten sensorien toiminnan perustuvan valon intensiteetin mittaamiseen eri optisen spektrumien taajuuksilta, aina ultraviolettivalon (280 nm - 360 nm) ja näkyvän valon (380 nm - 750 nm) taajuuksista infrapunasäteilyyn as-

ti (800 nm - 1000 nm). Optiset sensorit voidaan jakaa kolmeen luokkaan: näkyvä kuvantaminen, syvyyskuvantaminen ja lämpökamerat [14]. Näkyvä kuvantaminen tarkoittaa kamerapohjaista sensoria, jolla voidaan kuvata ihmisen liikkeitä. Syvyyskameralla tarkoitetaan puolestaan sellaista kameraa, jolla voidaan saada ihmisten liikkeistä kolmiulotteisia koordinaatteja ja nivelten välisten kulmien suuruuksia mitattua. Tällöin saadaan tavalliseen kaksiulotteiseen kameraan verrattuna tietoa toimintojen sijoittumisesta etu- ja taka-alalle. Lämpökamerat toimivat passiivisesti ja mittaavat ympäristön lämpimistä kohteista, kuten ihmiskehosta, säteilevää infrapunasäteilyä. Näkyvän valon havaitsemiseen perustuviin kameroihin verrattuna lämpökamera ei tarvitse valaistusolosuhteiden huomioimista.

Esimerkiksi artikkelissa [24] tehtiin GymCam-nimeä kantava sovellus, jossa syvyyskameran avulla pystyttiin tunnistamaan, seuraamaan ja laskemaan kuntosaliympäristössä useamman eri henkilön suorittamia liikkeitä. Infrapunasäteilyn mittaamiseen perustuvia sensoreita käyttivät puolestaan Shelke et al. [41], jotka tutkimuksessaan kahdella lämpösensorilla pystyivät tunnistamaan seisomista, istumista tuolilla ja lattialla sekä lattialla makaamista. Kamerapohjaisten ratkaisujen suurimpana ongelmana ovat kuitenkin yksityisyyteen liittyvät asiat, joskin infrapunakameralla ei tätä ongelmaa ole [14].

2.2.5 Elektromagneettiset sensorit

Tutkimus [14] kertoo elektromagneettisista sensoreista seuraavasti. Elektromagneettisilla sensoreilla tarkoitetaan sensoreita, jotka lähettävät ja vastaanottavat korkeataajuuksista elektromagneettista säteilyä. Tyypillisiä elektromagneettisia sensoreita ovat tutkasensorit ja WiFi-sensorit. Tutkan toimintaa käsitellään tarkemmin luvussa 3 ja tutkan käyttöä liikkeiden tunnistuksessa luvussa 5. WiFi-sensorin käytössä hyödynnetään sen WiFi-verkolle tyypillistä kanavan tilan tietämystä (*eng. channel state information*), joka kuvaa kanavan ominaisuutta vastaanottajan ja lähettäjän välillä. Esimerkiksi suosittu sovellus on käyttää WiFi-sormenjälkeä, jolloin henkilön tullessa WiFi-lähettimen ja vastaanottajan välille, muuttuu vastaanotetun signaalin (*eng. received signal strength, RSS*) voimakkuus, jota voidaan hyödyntää aktiivisuuden mittauksessa.

2.2.6 Hybridiratkaisut

Hybridiratkaisuja ovat nimensä mukaisesti sellaiset systeemit, jotka yhdistävät useampia erilaisia sensoreita ja niiden keräämää tietoa [14]. Esimerkiksi lämpötilaa, kemiallista koostumusta tai magneettista kenttää aistivaa sensoria pystytään harvoin yksinään käyttämään aktiivisuuden tai liikkeiden tunnistuksessa. Mutta yhdistettynä muihin sensoreihin niitä voidaan käyttää. Esimerkiksi magneettisia sensoreita voidaan käyttää siten, että niitä sijoitetaan huonekaluihin tai oviin, ja kun tarkkailtava kohde on suoraan kosketuksissa näihin, voidaan saada lisätietoa. Lämpötilaa, valoa, kosteutta ja hiilidioksidipitoisuutta ilmassa käytetään yleisesti älykotiratkaisuissa (*eng. smart home*) osana kokonaisvaltaista havainnointia kodin tilasta ja henkilöistä.

2.3 Vertailua eri sensoreiden välillä

Edellä kuvattiin ja jaoteltiin sensoreita sen mukaan, millaisia ominaisuuksia ja ilmiöitä ne mittaavat. Toinen tapa lähestyä asiaa on luokitella sensorit sen mukaan, missä niitä pidetään. Singh et al. [43] ovat koonneet katsauksessaan sensorit kolmen eri kategorian alle. Ensimmäisenä ryhmänä ovat puettavat sensorit, joista esimerkkeinä mainitaan kiihtyvyydsmittari (*eng. accelerometer*) ja gyroskooppi sekä älypuhelin. Toisena ryhmänä ovat ympäristöä tarkkailevat sensorit, joita ovat esimerkiksi kamera, ultraääni, tutka, akustiset sensorit ja lattian paine. Viimeisen ryhmän muodostavat hybridiratkaisut eli sellaiset sovellukset, jotka yhdistävät useampia erilaisia sensoreita.

Han et al. [19] tuovat esille tutkimuksessaan erilaisten sensoreiden hyviä ja huonoja puolia. Nämä ovat tiivistettynä taulukossa 2.1. Puettavien sensoreiden etuna voidaan pitää sitä, että niillä säilyy ihmisen yksityisyyden suoja, sillä ne eivät kuvaa ketään tai tallenna ääntä. Toisekseen puettavat sensorit on helppo asentaa ja ottaa käyttöön. Myös dataa saadaan kerättyä helposti niistä. Kustannustehokkuudeltaan ne ovat myös hyvä valinta, sillä ne eivät maksa kovin paljoa. Kuitenkin haittapuolena puettavissa sensoreissa on niiden epämukavuus ja sensoreiden pitäminen jatkuvasti päällä. Lisäksi akkua on ladattava tai patterit vaihdettava, joten käyttöaika on rajoitettu. Kamerapohjaisten ratkaisujen etuna on niiden toimintavarmuus ja tarkkuus verrattuna muihin tekniikoihin. Lisäksi verrattuna puettaviin laitteisiin, kamera on käyttäjäystävällisempi eikä se häiritse muuta elämää. Kuitenkin merkit-

tävä haittapuoli kameroiden suhteen on yksityisyyden suojan rikkominen. Kameroiden asentaminen makuu- ja kylpyhuoneisiin nostaa esiin eettisiä ja laillisia kysymyksiä. Haittana kameroiden suhteen on myös se, että ne tarvitsevat kohtuullisen valaistuksen toimiakseen hyvin. Ympäristön sensoreiden etuna voidaan pitää niiden helppoutta käyttäjälle sekä sitä, etteivät ne loukkaa ihmisen yksityisyyden suojaa. Toisaalta haittana on niiden kohtalaisen suuri väärin hälytysten määrä sekä se, että niitä voidaan käyttää vain sisätiloissa.

Sensori	Etu	Haitta
Puettava	<ul style="list-style-type: none"> - Yksityisyyden säilyttäminen - Kustannustehokas - Helppo asentaa ja kerätä dataa - Voidaan mitata tarkasti myös sydämen sykettä 	<ul style="list-style-type: none"> - Epämukava pitää ja asentaa vartalolle - Hankala pitää yllä jatkuvasti - Patterin/akun käyttöaika rajoitettu - Akun lataaminen kolmannen osapuolen toimesta
Kamera-pohjainen sensori	<ul style="list-style-type: none"> - Korkea havaitsemistarkkuus - Toimintavarma - Käyttäjystävällinen, häiritsee vähän elämisen toimintoja 	<ul style="list-style-type: none"> - Yksityisyydensuojan rikkominen - Pimeissä olosuhteissa toiminta heikkoa - Havaitseminen on hankalaa sensorin alueen ulkopuolella
Ympäristön sensori	<ul style="list-style-type: none"> - Yksityisyyden säilyttäminen - Käyttäjystävällinen, häiritsee vähän elämisen toimintoja 	<ul style="list-style-type: none"> - Altistuu helposti melulle - Vääriä hälytyksiä tapahtuu usein - Vain sisätilojen tarkkailu

Taulukko 2.1: Sensoreiden edut ja haitat Han et al. mukaan [19, s.4,taulukko 1]

3 Tutka

Tässä luvussa tehdään lyhyt katsaus tutkan toimintaan ja tarkastellaan siitä saatavaa dataa. Ensimmäinen aliluku 3.1 kertoo tutkan toiminnan periaatteista, aliluku 3.2 kuvailee tutkan tuottamaa informaatiota ja aliluku 3.3 esittää tutkan toiminta-alueet taajuuksien mukaan. Seuraavassa aliluvussa 3.4 tutkitaan tarkemmin millimetriaaltotutkaa ja sen käyttökohteita. Viimeinen aliluku 3.5 kuvaa, miten tutkan antamaa tietoa prosessoidaan.

3.1 Yleistä tutkasta

Tutka on elektromagneettinen systeemi, joka lähettää ympäristöönsä signaalin ja vastaanottaa kohteesta heijastuneen signaalin [44]. Tällä tavalla voidaan ympäristössä olevia ihmisiä, esineitä, kulkuneuvoja jne. havaita ja mitata etäisyys niihin. Tämän lisäksi johtopäätöksiä voidaan tehdä myös tutkimalla lähetetyn ja vastaanotetun signaalin eroja. Tutka pystyy toimimaan niin lyhyillä kuin pitkillä välimatkoilla eikä sen toimintaa häiritse pimeys, sade tai muut ympäristön olosuhteet. Tutkan muodostavat lähetin (*eng. transmitter*) ja vastaanotin (*eng. receiver*). Lähetin muodostaa elektromagneettisen signaalin, joka antennin kautta leviää ympäristöön. Signaali etenee ympäristössä ja osa siitä osuu kohteeseen ja heijastuu takaisin. Tutkan antenni kerää tätä takaisin tullutta säteilyä ja sen puolestaan ottaa vastaan vastaanotin. Useimmiten sama antenni toimii sekä lähettimenä että vastaanottimena. Kohteen etäisyys voidaan laskea mittaamalla se aika, mikä lähetetyllä signaalilla kestää kohteeseen ja takaisin vastaanottimelle.

Tutkan toiminnan takana on Doppler-ilmiö. Chen et al. [7] kuvaavat ilmiötä seuraavasti. Doppler-ilmiöllä tarkoitetaan sitä fysikaalista ilmiötä, jossa tutkan lähettämä elektromagneettinen signaali törmää liikkuvaan kohteeseen, jolloin tapahtuu muutoksia kantaja-aallon taajuudessa. Kun tähän lisätään vielä kohteen ns. mikroliikkeet, kuten värinä tai pyöriminen, taajuudessa tapahtuu sivuttaista muutosta, jolloin voidaan saada tarkempaa tietoa kohteen ominaisuuksista. Tätä kutsutaan mikro-Doppler ilmiöksi.

Kirjassa [45] jaotellaan tutkia eri kategorioihin seuraavasti. Pulssitutka (*eng. pul-*

se radar) lähettää sarjana pulsseja ja sen voidaan ajatella olevan eräänlainen perusmuoto tutkasta. Korkea resoluutio (*eng. high-resolution radar*) tutka viittaa sellaiseen tutkaan, jolla saadaan korkea resoluutio eli tarkkuus joko etäisyydessä, kulmassa tai nopeudessa. Jatkuva-aaltainen tutka (*eng. continuous wave, CW*) puolestaan tarkoittaa sellaista tutkaa, joka lähettää jatkuvaa sinimuotoista aaltoa ja sillä pystytään Doppler-taajuus vaihe-eron avulla havainnoimaan liikkuvia kohteita ja kohteen suhteellista nopeutta. FMCW-tutka (*eng. Frequency Modulated Continuous Wave*) eli taajuusmoduloitu tutka tarjoaa lisäksi mahdollisuuden etäisyyden mittaukseen. Valvontatutka (*eng. surveillance radar*) puolestaan havaitsee kohteen ja määrittää sen sijainnin etäisyytenä ja suuntana. Se voi myös valvoa kohdetta tietyn ajanjakson kuluessa. Tällaisia tutkia ovat esimerkiksi lentokoneen tai laivan havaitsemiseen käytetyt tutkat. Pulssi-Doppler tutka käyttää nimensä mukaisesti Doppler-taajuusmuutosta havaitakseen liikkuvia kohteita. Seurantatutka (*eng. tracking radar*) tarkoittaa sellaista tutkaa, jolla on tarkoitus tarkkailla yhden tai useamman kohteen rataa tai lentorataa. Muitakin tapoja luokitella ja tyypitellä tutkia on olemassa kuten esimerkiksi niiden sovelluskohteiden mukaan.

3.2 Tutkasta saatava data

Tutka tarjoaa tyypistään riippuen mahdollisuuden saada havaitsemastaan kohteesta erilaista dataa, jota Skolnik [45] kuvailee seuraavasti. Tärkeimpänä tietona usein on kohteen etäisyys (*eng. range*). Tämä saadaan selville mittaamalla aika, joka tutkan lähettämällä signaalilla kuluu kohteeseen ja takaisin. Se, kuinka tarkasti etäisyys voidaan määrittää tutkalla, riippuu kaistanleveydestä. Mitä suurempi on käytetty kaistanleveys, sitä suuremmaksi saadaan tarkkuus. Toinen tutkasta saatava tieto on kohteen radiaalinen nopeus eli etäisyyden muutos tietyn ajan kuluessa. Tämä voidaan esimerkiksi saada Doppler-taajuusmuutoksesta. Kolmas tärkeä tieto on kohteen kulmasuunta (*eng. angular direction*) eli se missä suunnassa kohde sijaitsee tutkaan nähden. Tämä voidaan laskea mittaamalla kulma tai suunta, josta palaava signaali tulee maksimissaan. Tarkkuus suunnan määrittämisessä riippuu antennin käyttämästä aallonpituudesta. Lisäksi tietoa voidaan saada kohteen koosta ja muodosta, mikäli tutkan tarkkuus riittää.

Tutkan tarkkuuteen ja sen luotettavaan kykyyn havaita kohteita vaikuttaa eniten signaali-kohinasuhde (*eng. Signal-to-Noise Ratio*), joka määritellään yhtälöllä

$$E/N_0, \quad (3.1)$$

missä E on tutkan vastaanottaman signaalin kokonaisenergia ja N_0 on kohinan suuruus kaistanleveyden suhteen vastaanottajalla [45].

Toinen tärkeä mitattava ominaisuus on Doppler-taajuusmuutos, joka mahdollistaa liikkuvan kohteen havaitsemisen muusta ympäristön kohinasta [45]. Se voidaan määritellä seuraavalla yhtälöllä

$$f_d = 2v_r/\lambda = (2v \cos \theta)/\lambda, \quad (3.2)$$

missä v_r on kohteen suhteellinen nopeus, v on kohteen absoluuttinen nopeus, λ on aallonpituus ja θ on kohteen suunnan ja tutkan säteen välinen kulma [45].

Tutkan yhtälö yhdistää etäisyyden tutkan ominaisuuksiin, kuten lähettimeen, vastaanottimeen sekä kohteeseen ja ympäristöön ja sillä saadaan selville maksimietäisyys, jolla tutka voi havaita kohteen, mutta sillä voidaan myös saada parempi ymmärrys tutkan suorituskyvystä [44]. Tutkan yhtälö kirjoitetaan yleensä seuraavassa muodossa [39]

$$P_r = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R^4}, \quad (3.3)$$

missä P_r on tutkan vastaanottama teho, P_t on lähetetyn signaalin teho watteina, G_t on lähettävän antennin hyöty, G_r on vastaanottavan antennin hyöty, λ on kantaja-aallon aallonpituus metreinä, σ on RCS-arvon (*eng. radar cross section*, kohteen heijastusvoima) keskiarvo neliömetreissä ja R on kohteen etäisyys tutkasta metreissä [39].

3.3 Tutkan käyttämät taajuudet

Kuten edellä todettiin, tutkia on erilaisia riippuen käyttötarkoituksista, ja näin ollen erilaiset tutkat käyttävät myös erilaisia taajuuksia toimiessaan. Usein tutkat käyttävät taajuuksia 5 MHz:n ja 95 GHz:n väliltä ja eri taajuudet tuovat mukanaan omat erityispiirteensä tutkille [45]. Matalammilla taajuuksilla toimivat tutkat sopivat paremmin pidempien etäisyyksien mittaamiseen, koska tehokkaita ja suurempia antennejä on helpompi saada matalammille taajuuksille. Toisaalta korkeammilla taajuuksilla saavutetaan tarkempi etäisyyden ja paikan määrittäminen, koska kaistanleveys saadaan leveämmäksi ja antenni saadaan pienemmäksi. Taulukossa 3.1 on esitettyä

ITU:n (*The International Telecommunication Union*) määrittämät nimelliset taajuusalueet. Rajat eivät ole tarkkoja, vaan eri taajuusalueet menevät päällekkäin. Lisäksi puhutaan millimetriaalloista, jotka sijoittuvat taajuusspektrissä ITU:n mukaan 30-300 GHz:n alueelle. Mikroaalloista taas puhutaan, kun taajuus on noin 300 MHz ja 40 GHz välillä.

Taajuusalueen nimitys	Nimellinen taajuusalue
HF	3 - 30 MHz
VHF	30 - 300 MHz
UHF	300 - 1000 MHz
L	1.0 - 2.0 GHz
S	2.0 - 4.0 GHz
C	4.0 - 8.0 GHz
X	8.0 - 12.0 GHz
K _u	12.0 - 18.0 GHz
K	18.0 - 27.0 GHz
K _a	27.0 - 40.0 GHz
V	40.0 - 75 GHz
W	75.0 - 110 GHz

Taulukko 3.1: Taajuusalueet Skolnik mukaan [45, s.35,taulukko 1.1]

3.4 Millimetriaaltotutka

Tutkaa, joka käyttää lyhyempää elektromagneettista aaltoa, kutsutaan millimetriaaltotutkaksi (*eng. millimeter wave, mmWave*) [21]. Nimensä mukaisesti tällainen tutka lähettää aaltoa, jonka aallonpituus on millimetrien luokkaa. Taajudella 76-81 GHz aallonpituus on noin 4 millimetriä. Lyhyemmän aallonpituuden johdosta tutkat voivat olla pienempiä kooltaan, koska esimerkiksi antennien koot saadaan pienemmiksi. Toinen etu on korkea tarkkuus, jolloin pientenkin liikkeiden huomaaminen on mahdollista.

Millimetriaaltotutkan mahdollisuudet ovat monenlaiset ja niiden käyttömahdollisuuksia tutkitaan enenevässä määrin. Esimerkiksi Alizadeh et al [2] tutkivat, kuinka ihmisen vitaalitoimintoja kuten hengitystiheyttä ja sydämen sykettä, voidaan seurata millimetriaaltotutkalla. Tutkijat toteavat millimetriaaltoa käyttävällä FMCW-tutkalla olevan monia etuja verrattuna muihin tutkiin. Heidän mukaansa millimet-

riaalto tarjoaa hyvin tarkan erotuskyvyn. Sillä pystytään erottamaan jopa niin pientä liikettä kuin rintakehän liikettä, joka on millimetriluokan liikettä. Toiseksi, jos käytetään nimenomaan FMCW eli taajuusmodulointua tutkaa, sillä voidaan myös paikantaa, mistä suunnasta signaalit palaavat ja näin ollen voidaan erottaa eri henkilöiden liikkeitä.

Wang et al. [57] puolestaan kuvaavat millimetriaaltotutkan eduksi erittäin laajan kaistanleveyden, joka on tulosta korkeiden taajuuksien (30-300 GHz) käyttämisestä kantaja-aaltolina ja siten laajemman resurssien tarjoamisen. Lisäksi he toteavat lyhyemmän aallonpituuden tarjoavan mahdollisuuden pakata antennijoukon pienen fyysiseen laitteeseen ja toisaalta taas verrattuna mikrometriaalloilla toimiviin tutkiin, millimetriaaltotutkilla on kapeampi säde ja sitä kautta voidaan myös pakata enemmän antenneja samaan laitteeseen.

Toisaalta millimetriaaltotutkan käyttö ei ole täysin ongelmaton. Artikkelissa [57] todetaan millimetriaaltotutkan lyhyen aallonpituuden tuovan ongelmia esimerkiksi signaalin voimakkaan vaimenemisen ilmakehän hapen vaikutuksesta, esineiden aiheuttavan helposti esteitä signaalin etenemiselle ja lisäksi millimetriaaltoa käytävällä tutkalla on lyhyt kantama. Taulukossa 3.2 on koottuna millimetriaaltotutkan hyötyjä ja haasteita.

Hyöty	Haaste
tarkka erotuskyky	voimakas vaimeneminen ilmakehässä
pienempi antennin koko	esineet aiheuttavat esteitä signaalin etenemiselle
useita antenneja mahtuu pienen laitteeseen	lyhyt kantama

Taulukko 3.2: Millimetriaalto tutkan hyötyjä ja haasteita

3.4.1 Millimetriaaltotutkan käyttökohteet ja taajuudet

Millimetriaaltotutkaa käytetään tyypillisesti kolmella eri taajuudella, 24 GHz, 60 GHz ja 77 GHz, riippuen sovelluksesta. Artikkelissa [36] kerrotaan millimetriaaltotutkan käyttökohteita olevan erilaiset kuluttajalaitteet, kuten oven avaajat ja automaattiset vesihanat, liikenteen nopeusvalvonta, autoissa käytetyt pysäköinnin avustajat ja muut ajamista avustavat sovellukset, helikoptereiden laskeutumisen avusta-

jat, teollisuudessa esimerkiksi tasojen mittaukset ja värinän tunnistus, turvallisuusalalla tunkeutumisen havaitseminen ja aseiden tunnistus sekä lisäksi erilaiset tieteelliset sovellukset kuten raketin telakoitumiset tai tulivuorten tutkimukset.

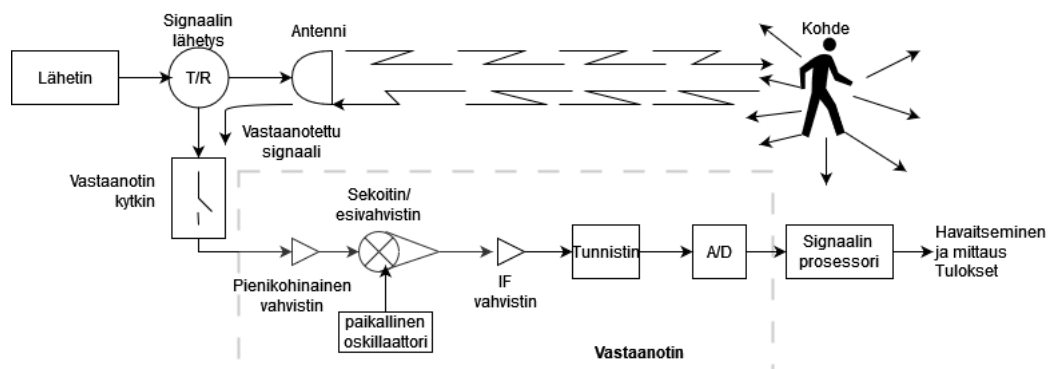
Lähteiden [37] ja [22] mukaan ISM-taajuuksista kerrotaan seuraavasti. Taajuudella 24 GHz tarkoitetaan ISM (industrial, scientific, medical) -kaistaa, joka toimii taajuudella 24.0-24.25 GHz ja tätä kutsutaan kapeakaistaiseksi (*narrow band, NB*), sillä se kattaa vain 250 MHz:n alueen. Tämä alue on lisensoimaton ja siten vapaasti käytettävissä. Lisäksi 24 GHz:n alueeseen kuuluu vielä 5 GHz:n UWB (ultrawide band) alue, joka on tosin poistunut vuoden 2022 alusta alkaen. 24 GHz:n taajuusalue on siis kapea ja ruuhkainen käyttää, joten tilanne on vaatinut muidenkin taajuusalueiden käytön ottamista mukaan. Siksi 24 GHz:n alue on jäämässä vähemmälle käytölle uusien teknologioiden tullessa tilalle.

Jacobi et al.[22] kertoo seuraavasti 60 GHz:n taajuudesta. Taajuusalue 60 GHz kattaa laajemman alueen verrattuna 24 GHz:n taajuusalueeseen, sillä se ulottuu 60 GHz:stä 64 GHz:iin, mikä tarkoittaa 4 GHz:n kaistaa. Se on myös ISM-taajuus eli sitä voidaan käyttää vapaasti. Usein millimetriaaltotutkaa käytetään kohteiden havaitsemiseen ja laskemiseen sekä myös kohteiden liikkeiden, etäisyyksien ja nopeuksien tarkasteluun. 60 GHz:n taajuudella toimivat tutkat tarjoavat 24 GHz:n tutkiin verrattuna paremman etäisyyden erotuskyvyn johtuen laajemmasta kaistasta, sekä paremman nopeuden erotuskyvyn, sillä keskitaajuus on korkeampi 60 GHz:n tutkissa.

Kolmantena tyyppillisenä millimetritutkan taajuusalueena on 77 GHz:n alue, joka käytännössä kattaa alueen 77 ja 81 GHz välillä. Tämä ei ole varsinaisesti ISM-taajuus, mutta Suomen viestintäviraston (Traficom) määräyksen [53] mukaan tämä alue on kuitenkin vapaasti käytettävissä. Artikkelin [37] mukaan taajuusalueesta käytettävän myös nimeä SRR (*eng. short range radar*) ja monet autoteollisuuden sovellukset, kuten ajoa avustavat tutkat sekä teollisuuden sovellukset, käyttävät tätä taajuusaluetta. Lisäksi tällä taajuudella toimivat tutkat tuovat paremman erotuskyvyn ja tarkkuuden niin nopeuden kuin etäisyydenkin suhteen verrattuna 24 GHz:n tutkiin. Lisäksi mitä korkeampaa taajuutta käytetään, sitä pienemmäksi saadaan tutkan koko, sillä antenni on pienempi.

3.5 Tutkan signaalin prosessointi

Tämä alaluku pohjautuu kirjan [39] tietoihin ja antaa yleiskäsityksen tutkan tuomasta tiedosta ja sen käsittelystä. Tutka lähettää radiotaajuudella (*eng. radiofrequency, RF*) elektromagneettisia aaltoja haluttuun suuntaan ja vastaanottaa heijastuneita aaltoja kohteista siltä alueelta. Tutkan perusrakenne, signaalin eteneminen ja heijastuminen kohteesta sekä signaalin vastaanottaminen on esitettyä kuvassa 3.1. Pääosan muodostavat lähetin, antenni, vastaanotin ja signaalin prosessori. Elektromagneettisten aaltojen muodostus tapahtuu lähettimessä. Antenni ottaa vastaan syötteenä nämä signaalit ja vapauttaa ne ympäröivään ilmakehään. Lähetin on kytkettyä antenniin lähetin/vastaanotin laitteeseen (*eng. transmit/receive device, TR*), joka on tyypillisesti kytkin tai kierrätin (*eng. circulator*). T/R-laitteen tarkoituksena on toimia välikappaleena vastaanottimen ja lähettimen välillä, jotta ne voivat molemmat kytkeytyä antenniin ja toimia samanaikaisesti. Lähetetty signaali kulkee ilman halki kohteeseen, joka heijastaa sen takaisin kohti tutkaa. Tutkan lähettämät signaalit osuvat myös muihin ympäristön esineisiin ja ei-toivottuihin kohteisiin, joista myös tapahtuu signaalin heijastumista takaisin. Näistä ei-toivotuista signaaleista käytetään englanninkielistä termiä *clutter*. Antenni ottaa vastaan nämä kaikki heijastuneet signaalit ja siirtää ne vastaanottimeen, joka koostuu useasta osasta. Vastaanottimen osien tarkoituksena on vahvistaa signaalia, muuntaa RF-signaali välitaajuudeksi (*eng. intermediate frequency, IF*) ja viedä signaali ADC-muuntimen läpi. Tunnistin (*eng. detector*) on se osa, joka poistaa moduloidusta signaalista kantajaosan. Lopuksi signaali annetaan lajittelua ja analysointia varten signaalin prosessorille (*eng. signal processor*).



Kuva 3.1: Tutkan pääkomponentit signaalin lähetykseen ja vastaanottamiseen kirjaa [39, s.5] mukailten.

Useimmiten tutkan signaalin prosessointi perustuu sen hyödyntämiseen, että kohteiden, kohinan ja ympäristöstä heijastuneiden signaalien aiheuttamissa Doppler taajuuksissa on eroa. Jos tutka ja kohde eivät ole toisiinsa nähden levossa, lähetetty taajuus f ja vastaanotettu taajuus f_r eroavat toisistaan Doppler-ilmiön takia. Tätä lähetetyn ja vastaanotetun taajuuden eroa kutsutaan Doppler-taajuudeksi tai Doppler-siirtymäksi (*Doppler frequency, Doppler shift*), joka voidaan määrittellä seuraavasti

$$f_d = \frac{2v}{c}f = \frac{2v}{\lambda} \quad , \quad (3.4)$$

missä λ on lähetetty aallonpituus ja v :n positiiviset arvot vastaavat lähestyviä kohteita. Doppler-siirtymää mitataan suorittamalla spektrianalyysi vastaanotetulle signaalille jokaisella etäisyyden lisäysvälillä. Spektrianalyysi suoritetaan yleensä lähettämällä useiden pulssien sarja ja suorittamalla *K- pisteen diskreetti Fourier-muunnos* (*eng. discrete Fourier transform, DFT*) palanneille signaaleille jokaiselta etäisyyden väliltä. Useimmiten käytetään DFT:stä muotoa *nopea Fourier-muunnos* (*eng. fast Fourier transform, FFT*). Doppler-vaihe on tärkeä ominaisuus mitata, sillä sen tiedon avulla voidaan vähentää kohinan vaikutusta, havaita useampi kohde samalta etäisyydeltä ja lisäksi havaita ja luokitella liikkuvia kohteita tai kohteita, joilla on liikkuvia osia. Tällä tavalla Doppler-vaihetta voidaan käyttää lajittelemaan kohteista palaavia signaaleja ympäristön heijastamista signaaleista käyttämällä suodatinta signaalin prosessorissa. Tämä on olennainen osa liikkuvan kohteen havaitsemista (*eng. moving target indication, MTI*).

Tutkan tuottama pulssijono voidaan kuvailla neljän eri aikaulottuvuuden mittarilla: radiotaajuuden aallon jakso (*eng. RF wave period*), pulssin leveys, pulssien toistoväli (*eng. pulse repetition interval, PRI*) ja yhtenäinen prosessointiväli (*eng. coherent processing interval, CPI*). Vastaavasti pulssijono voidaan kuvailla taajuuden mittareilla: tutkan radiotaajuus, pulssin taajuusalue, pulssien toistotaajuus (*eng. pulse repetition frequency, PRF*) ja spektriviivan leveys (*eng. spectral line bandwidth*). Fouriermuunnoksen ydin on siinä, että aika ja taajuus mitoilla on käänteinen yhteys toisiinsa: mitä pidempi ja levenneempi funktio on ajan suhteen, sitä lyhyempi tai kompaktimpi on vastaava taajuusspektri, ja toisinpäin.

Tutkan tavoitteena on mitata vastaanotetun signaalin amplitudi A , aikaviive ja vaihe-ero. Tutkan ilmaisim (*eng. detector*) saa signaalin sinimuotoisena pulssina ja ulostulona se tuottaa jännitettä (*eng. voltage*), jota kutsutaan kantataajuuden tai tutkan videon signaaliksi (*eng. baseband or radar video signal*). Useimmat nykyajan tut-

kista prosessoivat vastaanotetun signaalin vektoriksi, jolla on amplitudi (A) ja signaalin vaihe (θ). Tyypillisesti tutkan ilmaisimpiiri (*eng. detector circuit*) on synkroninen ilmaisin, joka muodostaa vektorista samanvaiheisen (*eng. in-phase, I*) komponentin ja 90 asteen (*eng. quadrature, Q*) vaihekomponentin. I ja Q komponentit voidaan määrittellä seuraavasti:

$$I = A \cos\left(\frac{4\pi R}{\lambda}\right) \quad (3.5)$$

$$Q = A \sin\left(\frac{4\pi R}{\lambda}\right) \quad (3.6)$$

missä R on etäisyys kohteeseen, λ on aallonpituus ja A on signaalin amplitudi. Käytännössä tämä toteutetaan siten, että vastaanotettu signaali jaetaan kahteen kanavaan joko koherentissa tai I/Q ilmaisimessa. Ylempää kanavaa kutsutaan I-kanavaksi tai vaiheessa olevaksi kanavaksi. Alempi kanava on Q-kanava ja on ylempään kanavaan verrattuna 90 asteen vaihe-erossa. I- ja Q-kanavien ulostulona muodostuu signaali y , joka on kompleksiluku ja muodostuu I:stä ja Q:sta seuraavasti:

$$\begin{aligned} y[0] &= y_I[0] + j \cdot y_Q[0] \\ &= A'(\cos \theta' + j \sin \theta') \\ &= A' \exp(j\theta') \end{aligned} \quad (3.7)$$

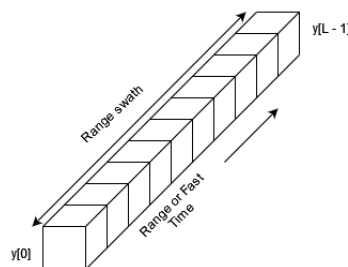
missä A' on amplitudi ja θ' signaalin vaihe. Kun tätä prosessia jatketaan useampaan näytteeseen peräkkäin, saadaan kompleksilukuna diskreetti aikasignaali, jota kutsutaan myös analyttiseksi signaaliksi (*eng. analytical signal*):

$$y[m] = y_I[m] + j \cdot y_Q[m] \quad (3.8)$$

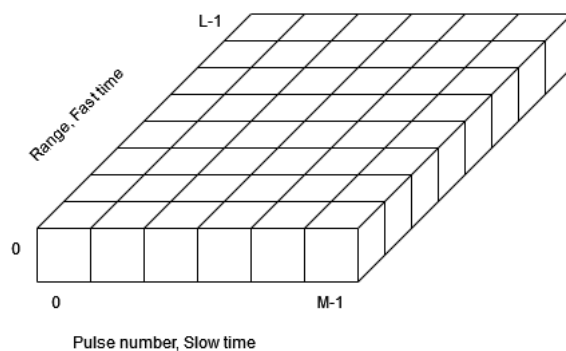
Edellä kuvatulla tavalla tutkan ilmaisin mittaa vastaanotetusta signaalista amplitudin ja vaiheen. Aikaviive t_0 voidaan arvioida signaalin lähettämisen jälkeen poimimalla tietyn väliajoin vastaanottimen saamia signaaleja toistuvasti ja havaitsemalla aika, jolloin palannut signaali vastaanotettiin. Näitä aikanäytteitä (*eng. time samples*) otetaan esimerkiksi pulssin pituuden, τ , välein. Näin ollen vastaanotettu ulostulo näytteistetään joka τ sekunnin jälkeen tietyn aloitusajan, t_1 ja lopetusajan t_2 välillä. Tällöin, koska vastaanotettu signaali on τ sekunnin pituinen, heijastunut signaali vastaa yhtä tiettyä aikanäytettä t' . Näin voidaan päätellä aikaviiveen kohteeseen olevan t' sekuntia ja siten etäisyys kohteeseen on $R' = ct'/2$ metriä, missä c on valonnopeus.

Jokainen aikanäyte näytteistetään muodostaen, y , joka koostuu I ja Q osista. Näytteistyksellä (*eng. sampling*) tarkoitetaan yksittäisen muuttujan diskretisointia (*eng. discretization*) eli jatkuvan äärettömän muuttujan muuttamista äärelliseksi, diskreeteiksi osiksi. Tämä tehdään useimmiten ajan suhteen. Jos diskretisointi tapahtuu amplitudin suhteen, puhutaan kvantisoinnista (*eng. quantization*). Tyypillisesti nämä sarjana tulevat näytteet tallennetaan tietokoneen muistiin yksiulotteisena vektorina. Tätä havainnollistetaan kuvassa 3.2. Tätä näytteiden sarjaa kutsutaan nimellä *range bin* eli etäisyyskori ja väliä ensimmäisen ja viimeisen näytteen välillä kutsutaan nimellä *range swath* eli etäisyysalue ja se ulottuu etäisyydeltä $R_1 = ct_1/2$ etäisyyteen $R_2 = ct_2/2$ asti.

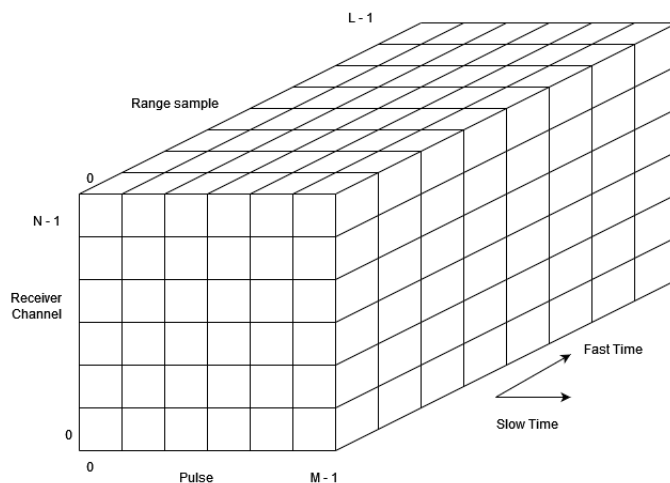
Tutka lähettää M määrän pulsseja ja jokaista lähetettyä pulssia vastaa *range bin* kuten kuvassa 3.2. Nämä laitetaan muistiin kaksiulotteiseen matriisiin kuten kuvassa 3.3. Jokaista lähetettyä signaalia kohden palaa takaisin sarja heijastuneita signaaleita peräkkäisiltä etäisyyskorilta. Voidaankin sanoa, että nämä etäisyysnäytteet (*eng. range sample*) kerätään nopeudella, joka vastaa vähintään signaalin kaistanleveyttä. Tätä ulottuvuutta kutsutaan termillä *fast time*. Kuvassa 3.3 on havainnollistettu tilannetta, jossa M on pulssien lukumäärä ja L on etäisyysnäytteiden määrä. Ensimmäisen ja viimeisen etäisyysnäytteen ajat määritellään alkuetäisyyden ja loppuetäisyyden sekä valonnopeuden avulla laskien $t = 2R/c$. Jokaiselle pulssille tehdään sama alku- ja etäisyysajan määrittely, jolloin muodostuu kaksiulotteinen etäisyyspulssi (*eng. range-pulse number*) matriisi kerätystä datasta. Pulssien lukumäärä akselia kutsutaan termillä *slow time*, koska siinä suunnassa tutkan pulssin toistoväli on suurempi kuin *fast time* suunnassa. Koska tutkalla on useita vastaanottavia kanavia, voidaan vastaanotetuista signaaleista muodostaa päällekkäisiä edellä kuvattuja 2-ulotteisia matriiseja, jolloin saadaan niin kutsuttu tutkakuutio (*eng. radar cube*). Tätä havainnollistetaan kuvassa 3.4, jossa N on vastaanottavien kanavien määrä.



Kuva 3.2: Tutkan etäisyyskori (*eng. range bin*) [39, s.327] mukailten.

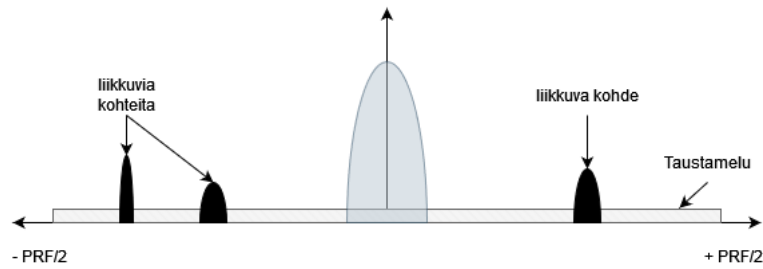


Kuva 3.3: Fast time - slow time matriisi pulssitutkasta [39, s.539] mukaillen.

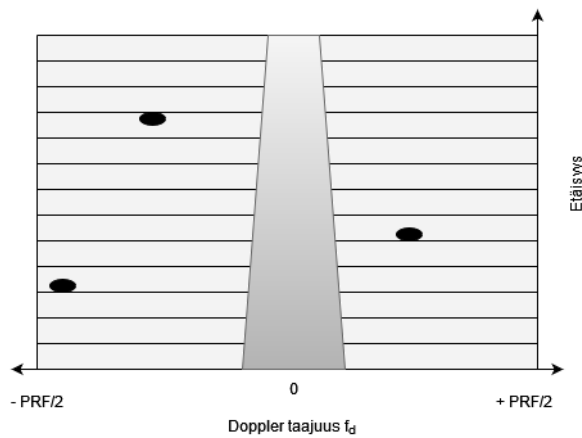


Kuva 3.4: Tutkan datakuutio [39, s.540] mukaillen.

Tutkan vastaanotin voi havaita signaaleja, joita tulee itse kohteesta (paikallaan pysyvä tai liikkuva), vastaanottimen kohinaa, ilmakehän kohinaa, ympäristöstä heijastuvaa kohinaa tai esimerkiksi häiriötä matkapuhelimista, jotka toimivat samalla taajuudella. Etäisyyskorin (*range bin*) Doppler-spektri on diskreetti-aika Fouriermuunnos (*eng. discrete-time Fourier transform, DTFT*) hitaasta ajasta (*slow time*) kyseisellä etäisyyskorilla. Jokaisella signaalilla, jonka tutka havaitsee, on erilainen Doppler-spektri. Tällöin kokonaisuudessaan tutkan havainnoima Doppler-spektri on kaikkien sen havainnoimien signaalien superpositio.

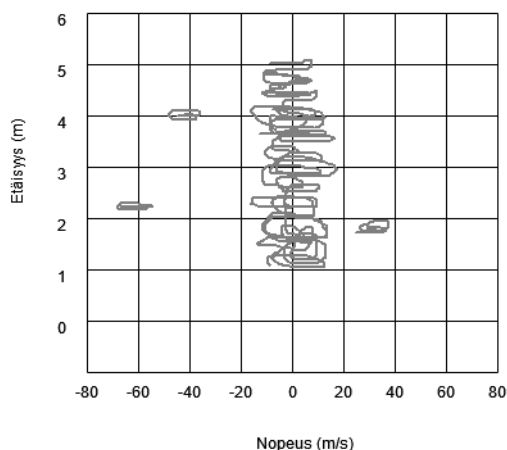


Kuva 3.5: Teoreettinen tilanne yhden etäisyyskorin Doppler-spektristä, paikallaan olevan tutkan havainnoimana Richards et al.[39, s.331] mukaan.



Kuva 3.6: Teoreettinen tilanne etäisyys-Doppler kuvaajassa, paikallaan olevan tutkan havainnoimana Richards et al.[39, s.332] mukaillen.

Kuvassa 3.5 havainnollistetaan tilannetta, jossa on yhden tutkan etäisyyskorin Doppler-spektri väliltä $-PRF/2$ ja $+PRF/2$ eli kahden pulssin toistotaajuuden väli. Keskellä kuvaajaa on harmaa pylväs, jossa Doppler-taajuus saa arvon 0. Tämä kuvaa kaikkea paikallaan olevaa ympäristöstä tuleva heijastumaa, kuten puista ja rakennuksista takaisin heijastuvat signaalit. Harmaa, matalampi energinen alue taustalla kuvastaa vastaanottimen valkoista kohinaa (*white noise*). Kolme mustaa pientä pylvästä ovat liikkuvia kohteita, kaksi negatiivisella Doppler-taajuudella tarkoittaen sitä, että ne liikkuvat pois päin tutkasta ja yksi positiivisella Doppler-taajuudella, joka kulkee kohti tutkaa. Useimmiten etäisyyskoreja on useita ja kohteita voi sijaita siten useamman korin alueella. Myös ympäristöstä heijastuneet signaalit vaihtelevat etäisyyden mukaan. Taustamelua esiintyy kaikilla etäisyyksillä. Kuvassa 3.6 on esitetty tilanne, jossa kohteet, ympäristön signaalit ja taustamelu ovat etäisyys-Doppler kuvaajassa. Taustakohina on tasaista koko alueella, kolme kohdetta ovat nyt samoil-



Kuva 3.7: Etäisyys-nopeus spektri paikallaan olevan tutkan havainnoimana Richards et al.[39, s.333] mukaillen.

la Doppler-taajuuksilla, mutta eri etäisyyskoreissa verrattuna kuvaan 3.5. Keskellä oleva harmaa pylväs kuvastaa edelleen ympäristön paikallaan pysyvistä esineistä heijastuneita signaaleja ja värin haalistuminen taas sitä, että heijastuminen vähenee etäisyyden lisääntyessä. Kun halutaan tilanteesta esitys etäisyys-nopeus kuvaajana, voidaan se esittää kuvan 3.7 tavoin. Kuvassa näkyy edelleen keskellä oleva paikallaan olevien kohteiden aiheuttama signaalien voimakkuus ja kolme pienempää, liikkuvaa kohdetta on nähtävillä omilla etäisyyksillään. Taustakohinaa ei enää ole esityksessä mukana.

Doppler-prosessointi (*eng. Doppler processing*) tarkoittaa sitä suodatusta tai spektrin analysointia *slow time* signaalille, joka saadaan tietyltä etäisyyskorilta (*range bin*) tietyn ajanjakson yli vastaten useampaa pulssia. Doppler-prosessointi suoritetaan jokaiselle tutkittavalla etäisyyskorille erikseen. Se voidaan jakaa kahteen luokkaan: liikkuvan kohteen indikaatio (MTI) ja pulssi-Doppler prosessointi. MTI-prosessoinnilla tarkoitetaan tässä *slow time* -signaalin prosessointia kokonaisuudessaan aikaulottuvuudessa ja useimmiten se tehdään käyttäen yhtä ylipäästösuodatinta (*eng. hig-pass filter*). Pulssi-Doppler -prosessoinnilla puolestaan tarkoitetaan sitä prosessia, jossa signaalia prosessoidaan taajuusulottuvuudessa, johon useimmiten käytetään *nopeaa Fourier-muunnosta*, FFT. MTI-prosessointi tuottaa rajallisen määrän tietoa, mutta vaatii vähemmän laskentaa verrattuna FFT:hen. FFT puolestaan tarjoaa enemmän tietoa ja paremman signaali-häiriösuhteen (*eng. signal-to-interference rattiio, SIR*).

MTI-prosessointi käyttää lineaarista suodatinta *slow time* -informaatioon, jotta saadaan vähennettyä ympäristöstä heijastuvien signaalien määrää. Tällä tavalla saa-

daan muokattu *slow time* -signaali, joka sisältää osia kohinasta, mutta myös kohteista. Tämä signaali kulkee detektorin läpi ja tietyn amplitudin ylittyessä, kohde havaitaan. MTI-prosessointi tarjoaa vain liikkuvan kohteen tiedon, eikä esimerkiksi sitä, mihin suuntaan kohde liikkuu tai millä nopeudella. Doppler-prosessointi sen sijaan tekee spektrianalyysin *slow time* -datalle jokaisella etäisyyskorilla. Näin saadaan etäisyys-Doppler-matriisi, joissa ulottuvuudet ovat *fast time* ja Doppler-taajuus. FFT-prosessoinnilla jaetaan Doppler-ulottuvuus Doppler-koreihin (*eng. Doppler bin*). Etäisyys-Doppler datassa liikkuvasta kohteesta tuleva energia erotetaan ympäristön muusta signaalista ja se kilpailee vain taustakohinan kanssa kohteen Doppler-korissa. Lisäksi sen perusteella, missä Doppler-korissa kohde sijaitsee, voidaan päätellä sen Doppler-siirtymä ja siten radiaalinen nopeus. Spektrianalyysi tehdään suorittamalla Fourier-muunnos jokaiselle *slow time* -riville käyttäen FFT algoritmia. Pulssi-Doppler-prosessoinnin etuna onkin sen tarjoama tieto kohteen nopeudesta sekä mahdollisuus havaita useampia kohteita.

4 Syväoppiminen

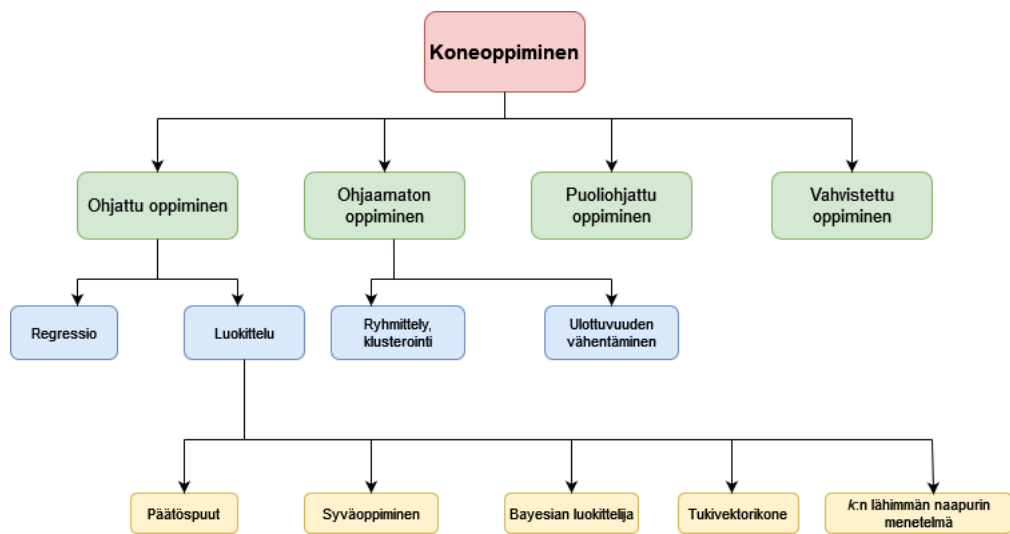
Tämä luku tekee katsauksen syväoppimiseen. Aliluku 4.1 kuvailee lyhyesti syväoppimisen perusteita ja jaottelua. Seuraava aliluku 4.2 ja sen aliluvut keskittyvät yhden syväoppimismenetelmän, nimittäin konvoluutioneuroverkkojen, kuvailuun. Viimeinen aliluku 4.3 kokoaa yhteen syväoppimisen suorituskyvyn arvioinnissa tarvittavia työkaluja ja metriikkoja.

4.1 Yleistä syväoppimisesta

Koneoppiminen (*eng. machine learning*, ML) tarkoittaa prosessia, jossa kone oppii saadusta aineistosta ilman, että sitä on erikseen ohjelmoitu [27]. Se on tekoälyn (*eng. Artificial Intelligence*, AI) alamuoto, joka mallintaa systeemin ominaisuuksia [9]. Kone voidaan opettaa tunnistamaan tai luokittelemaan aineistoa syöttämälle sille raakadataa ja sitä voidaan hyödyntää monenlaisissa sovelluksissa, kuten kohteiden tunnistamisessa kuvista, puheen kääntämisessä tekstiksi tai olennaisten hakutulosten valitsemisessa [28]. Syväoppiminen (*eng. deep learning*) on yksi koneoppimisen muoto, jota tapahtuu usealla eri tasolla ja josta voidaan todeta muodostuvan tietynlainen verkosto, niin kutsuttu neuroverkko [28].

Kuvassa 4.1 nähdään koneoppimisen luokittelu Kumar et al. mukaan [27]. Koneoppiminen voidaan jakaa ohjattuun oppimiseen, ohjaamattomaan oppimiseen, puoliohjattuun oppimiseen sekä vahvistettuun oppimiseen. Nämä luokat jakaantuvat edelleen alaluokkiin. Ohjattu oppiminen tarkoittaa sitä, että kone saa joukon syötteitä ja ulostuloja tunnisteineen ja harjoittellessaan se oppii syötteen ja ulostulon yhteyden ja riippuvuuden. Ohjattu oppiminen jakaantuu edelleen regressioon eli jatkuvaan luokitteluun sekä erillisiin ryhmiin luokitteluun. Luokittelu voidaan edelleen jakaa loogiseen luokitteluun, josta päätöspuut on esimerkki, perseptronipohjaiseen luokitteluun (syväoppiminen), tilastolliseen luokitteluun (Bayesian ja tukivektorikone) sekä instanssipohjaiseen (k :n lähimmän naapurin menetelmä).

Edelleen Kumar [27] jatkaa koneoppimisesta seuraavasti. Ohjaamaton oppiminen viittaa sellaiseen oppimiseen, missä syötteeseen ei ole annettuna tunnistettua ulostuloa, vaan malli yrittää löytää yhteydet datasta. Tarkoituksena on luokitella sa-

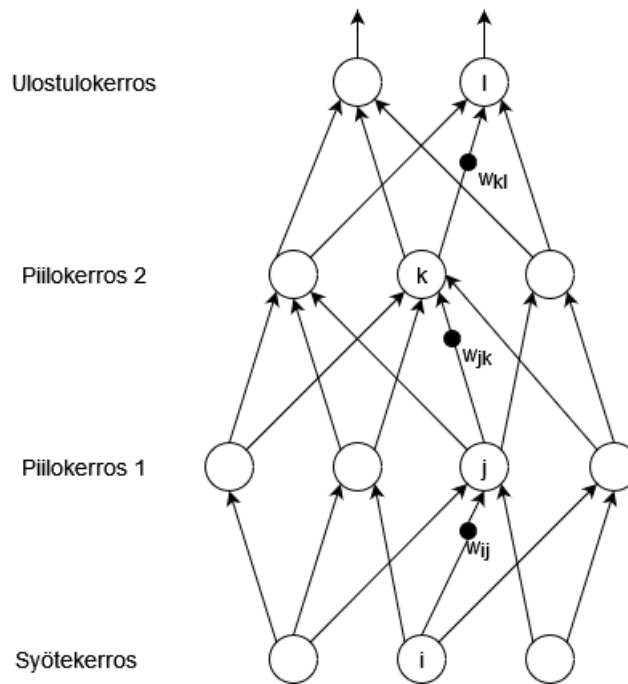


Kuva 4.1: Koneoppimisen luokittelu Kumar et al. mukaan [27, s.3], muokattu.

mankaltaisten mallien joukko ryhmiin, vähentämällä joukon ulottuvuuksia tai huomaamalla poikkeavuuksia joukoista. PuoliOhjattu oppiminen puolestaan toimii sekä tunnistetun datajoukon että merkitsemättömän datajoukon parissa. Tarkoituksena on ennustaa merkitsemättömän datan tunnistetut tai tulevien testidatojen tunnistetut ennustaminen. Viimeinen kategoria on vahvistettu oppiminen, jolla tarkoitetaan sellaista koneoppimista, jossa kone oppii ympäristöstä saamansa palautteen perusteella.

Syväoppiminen toimii koneoppimisalgoritmeja hyväksikäyttäen ja keinotekoiseseen hermoverkkojärjestelmään (*eng. Artificial Neural Network, ANN*) pohjautuen [9]. Verkko ottaa oppimisalgoritmeja ja kasvattaa koko ajan datan määrää prosessissa, jolloin prosessin tehokkuus kasvaa. Oppimisprosessia sanotaan syväksi, koska neuroverkon tasojen määrä kasvaa ajan myötä. Syväoppiminen jakautuu kahteen vaiheeseen, nimittäin harjoitusvaiheeseen ja päättelyvaiheeseen. Harjoitusvaiheessa suuret datamäärät merkitään ja määritetään. Päättelyvaiheessa tehdään johtopäätökset ja merkitään uutta, tutkimatonta dataa hyödyntäen aikaisempaa osaamista. Syväoppimisen avulla voidaan saavuttaa ymmärrys monimutkaisista havainnoista suurella tarkkuudella. Oppiminen systeemissä voi tapahtua joko ohjatusti tai ohjaamattomasti.

Kuvassa 4.2 on esitettynä neuroverkon yleinen rakenne. Artikkelin [28] mukaan neuroverkko muodostuu sisääntulo- ja ulostulokerroksista, joiden välissä on yhdestä useampaan piilotettuja kerroksia. Jokaisessa kerroksessa lasketaan syötteenä saattujen arvojen painotettu summa. Tähän yleensä lisätään neuronin vakiotermin.



Kuva 4.2: Monikerroksinen neuroverkko LeCun et al. [28, s.437] mukailleen.

Neuroverkoille tyypillistä on käyttää niin sanottuja aktivaatiofunktioita. Dong et. al [12] kuvailevat aktivaatiofunktioiden toimintaa seuraavasti. Nämä funktiot generoivat nonlinearisia yhteyksiä sisääntulon ja ulostulon välille. Saatu epäjatkuvuus sekä verkon neuronit ja useat kerrokset matkivat ihmisaivojen rakennetta, mistä nimi neuroverkko tulee. Aktivaatiofunktioita on erilaisia, joista muutamia käytetyimpiä ovat Sigmoid-funktio, Hyperbolinen tangetti ja ReLu (*eng. Rectified Linear Unit*). Aktivaatiofunktion tehtävänä on muuntaa data paremmin luokiteltavampaan muotoon. Esimerkiksi Sigmoid-funktion tehtävänä on ottaa mikä tahansa syöte ja antaa ulostulona arvo 0 ja 1 välillä, jolloin tätä voidaan käyttää luokitteluongelman ratkaisuun.

Artikkelin [1] mukaisesti voidaan tiivistää seuraavaa syväoppimisen aineistosta. On erityisen tärkeää, että tekoälyä hyödyntävällä sovelluksella on käytössään laadukas ja tarpeeksi laaja datasetti eli aineisto, jolla harjoittelu toteutetaan. Aineiston on myös tärkeää mahdollisimman hyvin vastata reaalimaailman tilanteita sekä sisältää erilaisia tapauksia. Riippuen sovelluksesta ja kuinka paljon se tekee itsenäisesti päätöksiä, kuten automaattinen auton ohjaus, tarvitaan myös määrällisesti hyvin paljon esimerkkejä, jotta voidaan taata sovelluksen turvallinen toiminta. Usein myös data tulee ennen opetusprosessia merkitä (*eng. label*) hyvin tarkasti.

4.2 Konvoluutioneuroverkko

Konvoluutioneuroverkkoa (*eng. convolutional neural network, CNN*) käytetään tyypillisesti kuvamuotoisen datan käsittelyyn ja tunnistukseen tai esimerkiksi käsin kirjoitetun tekstin tunnistukseen [9]. Konvoluutioverkon sanotaan olevan eteenpäin syöttävä (*eng. feedforward*), mikä tarkoittaa sitä, että verkossa neuronit kartoittavat kiinteän kokoisen syötteen, kuten kuvan, kiinteän kokoiseksi ulostuloksi, kuten ennustukseksi kuvan luokasta [28]. Verkon arkkitehtuuria voidaan verrata näköaistiin, sillä konvoluutioverkon ytimet (*eng. kernels*) vastaavat erilaisia näköaistin reseptoreita, jotka reagoivat eri ominaisuuksiin, ja aktivointifunktiot kuvaavat aistin toimintaa, jossa vain tietyn kynnyksen ylittävät hermosähkösignaalit aiheuttavat signaalin etenemisen seuraavalle neuronille [32]. Konvoluutioneuroverkolle on tyypillistä myös se, että jokainen neuroni yhdistyy vain osaan edellisen kerroksen neuroneista verrattuna tavanomaiseen neuroniverkkoon, jossa yhteydet ovat kaikkiin neuroneihin [12].

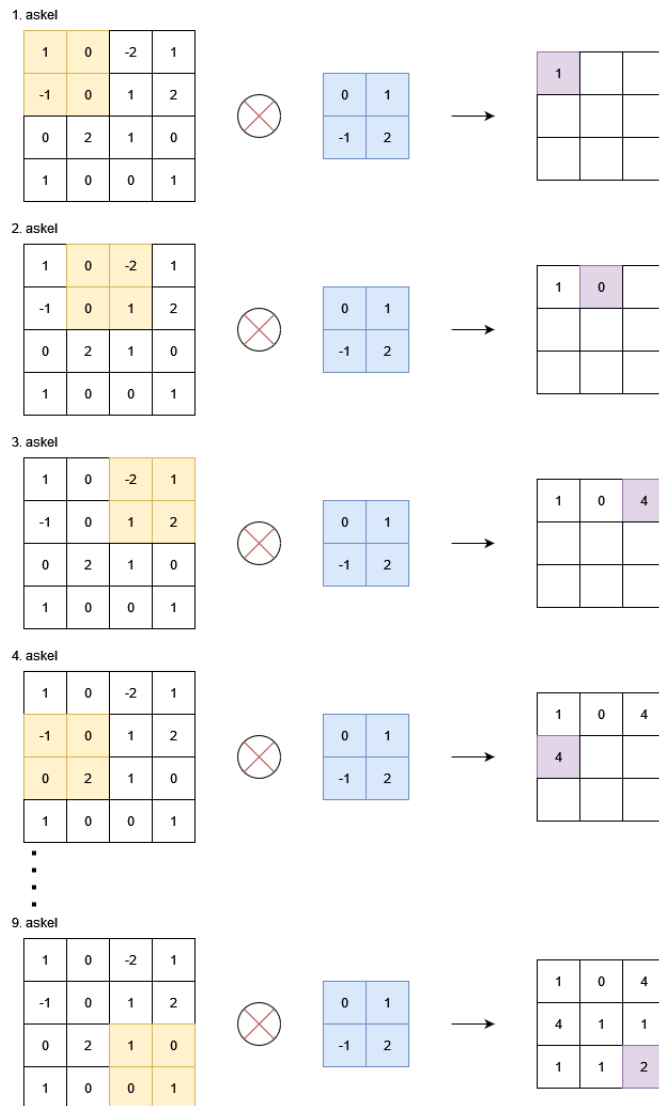
4.2.1 Konvoluutioverkon rakenne

Konvoluutioneuroverkossa on useita kerroksia ja jokaisella tasolla on oma tehtävänsä, siten että alimmat kerrokset tunnistavat muodon reunat ja seuraavat tarkempia ominaisuuksia reunojen sisältä [9]. Tyypillisesti rakenne on päällekkäisiä kerroksia, joissa vaihtelevat konvoluutiokerrokset (*eng. convolutional layers*), alinäytteistyskerrokset (*eng. pooling*) ja lopuksi tulevat yhdistävät kerrokset (*eng. fully connected layers*) [58].

0	1
-1	2

Kuva 4.3: 2x2 kerneli Ghosh et al. [17, s.6] mukaillen.

Konvoluutiokerrokset muodostavat verkossa sen osan, joka tunnistaa piirteet ja se sisältää joukon konvolutionaalisia kerneleitä (*eng. kernels*) eli filttäreitä [17]. Kernelit ovat erillisten arvojen tai numeroiden muodostama $n \times n$ -kokoinen matriisi, kuten kuvassa 4.3 on esitetty. Jokainen numero matriisissa edustaa painokerrointa (*eng. weight*) kernelissä. Verkon opetuksen alussa painokertoimet ovat alustettu satunnaisesti, mutta opetuksen edetessä arvoja päivitetään ja kerneli on oppinut merkityksellisiä piirteitä aineistosta.

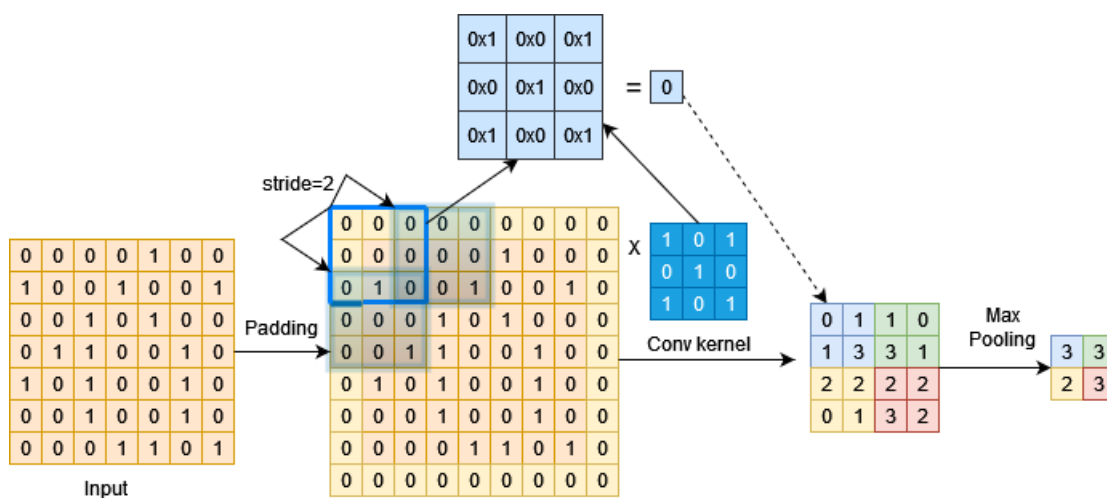


Kuva 4.4: Konvoluutio-operaation askeleet Ghosh et al. [17, s.7] mukaillen.

Konvoluutioneuroverkko saa syötteenään monikanavaisen kuvan, kuten kolmekanavaisen RGB-kuvan tai yksikanavaisen harmaaskaalatun kuvan ja suorittaa konvoluutio-operaatioita [17]. Kuvassa 4.4 on esitettyä konvoluutio-operaation toimintaa neljän ensimmäisen vaiheen sekä viimeiseen askeleen osalta. Kuvassa on 4x4-kokoinen kuva sekä 2x2-kokoinen kerneli. Konvoluutio-operaatiossa kerneli liikkuu kuvan päällä vaakasuorassa ja pystysuorassa. Ensimmäisessä vaiheessa kerneli on kuvan ylimmässä vasemmassa nurkassa ja tapahtuu pistetulo kuvan 2x2-kokoisen osa-alueen ja kernelin välillä. Tällöin matriisien vastaavissa kohdissa olevat arvot kerrotaan keskenään ja kaikki saadut tulot lasketaan yhteen, jolloin saadaan yksi lukuarvo ulostulona tulevaan piirrekarttaan (*eng. feature map*). Toisessa

askeleessa kerneli liikkuu yhden askeleen verran oikealle kuvan päällä ja tekee saman laskutoimituksen kuten edellä. Näin jatketaan koko kuvan läpi, jolloin tässä tapauksessa on tuloksena 3x3-kokoinen piirrekartta kuvasta, joka on nähtävissä kuvan 4.4 viimeisen vaiheen ulostulona. Tässä esimerkissä askelkoko (*eng. stride*) on asetettu 1, mutta se voi olla muutakin. Jos sitä kasvatetaan, saadaan kooltaan pienempi piirrekartta.

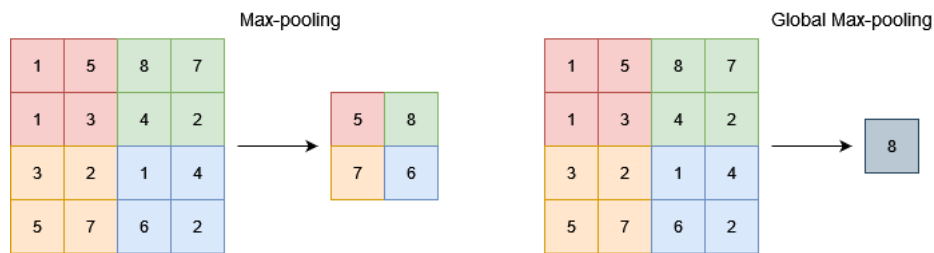
Kuvassa 4.5 on esitetty toinen konvoluutio-operaatio. Kuvassa verkko saa syöteenä 7x7 kokoisen kuvan, joka koostuu 0:sta ja 1:stä. Kun kernel asetetaan tietyn kokoiseksi, reunalta voi hävitä tietoa ja tällöin käytetäänkin täyttämistä (*eng. padding*) kuvan reunoille [32]. Kuvassa 4.5 kerneli on 3x3 kokoinen sinisellä merkitty matriisi, joka kulkee kuvamatriisin läpi ottaen kerrallaan yhden 3x3-kokoisen osamatriisin kuvasta ja kertoo siinä olevat luvut ja summaa ne yhteen. Tästä yhden 3x3 matriisin käsittelystä saadaan yksi luku, joka tulee seuraavan kerroksen 4x4-kokoisen matriisin ensimmäiseen paikkaan. Kuvassa askelkoko (*eng. stride*) on 2, joten kerneli aloittaa seuraavan osajoukon käsittelyn 2 solun päästä edellisen alusta. Tällä tavalla kerneli käy koko 9x9-kokoisen kuvamatriisin läpi, josta tulee yhteensä 16 lukua muodostaen 4x4-kokoisen matriisin.



Kuva 4.5: Konvoluutioverkon toiminta Li et al. [32, s.2] mukailten.

Toisena osana neuroverkkoa on alinäytteistys (*eng. sub-sampling, down-sampling*), joissa tapahtuu piirrekartan ulottuvuuksien vähentämistä, mikä vähentää opittavien parametrien määrää [58]. Alinäytteistys tapahtuu poolauksen (*eng. pooling*) kautta, joka tarkoittaa osaotantaa, jolla vähennetään signaalin kokoa, siten että saavutetaan parempi kestävyys kohinaa vastaan [35]. Se antaa ulostuloarvon kaikille ar-

voille, jotka on kernelissä tutkittu yhdellä kertaa läpi ja se voi perustua esimerkiksi maksimi- tai keskiarvoon [12]. Kuvassa 4.5 4x4 matriisista valitaan maksimiarvo jokaisesta 2x2 osamatriisista, jolloin lopputuloksena on 2x2-kokoinen matriisi sisältäen suurimmat arvot. Alinäytteistyskerroksen lisäksi voidaan käyttää myös yleistä eli globaalia alinäytteistystä joko keskiarvoon tai maksimiarvoon perustuen. Globaali maksimialinäytteistyskerros (*eng. global max-pooling*) antaa koko matriisin suurimman arvon ulostulona [25]. Kuvassa 4.6 on esitettyä tavallisen ja globaalin maksimialinäytteistyskerroksen ero. Ensimmäisessä otetaan kustakin osa-alueesta suurin luku, kun taas globaalissa otetaan koko matriisin suurin luku.



Kuva 4.6: Maksimialinäytteistyskerroksen ja globaalin maksimialinäytteistyskerroksen ero [25, s.10] mukaan.

Lineaarisen operaation, kuten edellä kuvatun konvoluution, ulostulo ohjataan tämän jälkeen epälineaarisen aktivaatiofunktion läpi [58]. Käytetyin aktivointifunktio on ReLU (*eng. rectified linear unit*), joka voidaan määritellä seuraavasti

$$f(x) = \max(0, x), \quad (4.1)$$

missä funktio f saa arvon 0 kun $x \leq 0$ ja arvon x kun $x > 0$ [28]. Vaikka ReLU-funktiolla on epäjatkuvuuskohta 0:ssa, sen on todettu olevan tehokkaampi ja nopeampi kuin tavalliset tanh- ja sigmoid-aktivointifunktiot [18].

Toinen käytetty aktivointifunktio on ELU (*Exponential Linear Unit*), joka eroaa ReLU:sta siinä, että sillä on myös negatiivinen osa, mikä nopeuttaa laskentaa varsinkin jos syöte on negatiivista [18]. ELU-aktivointifunktio voidaan määritellä seuraavasti

$$a_{i,j,k} = \max(z_{i,j,k}, 0) + \min(\lambda(e^{z_{i,j,k}} - 1), 0), \quad (4.2)$$

missä λ on ennalta määritelty parametri, jolla kontrolloidaan negatiivista syötettä [18].

Viimeisen konvoluutio- tai alinäytteistyskerroksen jälkeen tulee tyypillisesti flatten-kerros eli tasoituskerros ja sen jälkeen tulee useimmiten ainakin yksi yhdistävä kerros.

ros eli dense-kerros [58]. Flatten-kerros muuntaa piirrekartan yksiulotteiseksi vektoriksi. Dense-kerroksella puolestaan jokainen syöte yhdistetään ulostuloon opittavan painokertoimen avulla. Viimeisessä yhdistävässä kerroksessa on yhtä monta ulostuloa kuin on luokkia aineistossa. Jokaisen yhdistävän kerroksen jälkeen tulee epälineaarinen funktio, kuten ReLU.

Viimeisen yhdistävän kerroksen aktivaatiofunktio poikkeaa aiempien kerrosten funktioista, koska se tulee valita sen mukaan, millaisesta luokitteluongelmasta on kyse [58]. Sigmoid-aktivaatiofunktio on tyypillinen aktivointifunktio binääriluokittelussa, jossa on kaksi luokkaa, sillä se ottaa minkä tahansa syötteen ja antaa arvon 0 ja 1 välillä [12]. Moniluokitteluongelmassa puolestaan käytetään useimmiten Softmax-funktiota, sillä se normalisoi viimeisen kerroksen ulostuloarvot kohde-luokkien todennäköisyyksiksi [58].

4.2.2 Konvoluutioneuroverkon opettaminen ja optimointi

Konvoluutioneuroverkon opettamisella tarkoitetaan sitä prosessia, jossa etsitään konvoluutiokerrosten kernelit ja yhdistävien kerrosten painokertoimet, joilla saavutetaan ulostuloarvot, jotka poikkeavat mahdollisimman vähän annetuista, todellisista totuuksista [58]. Opetus tapahtuu kahdessa vaiheessa. Ensimmäinen vaihe on eteenpäin tapahtuvaa oppimista (*eng. forward training phase*), joka tarkoittaa sitä prosessia, joka tapahtuu kun verkkoon annetusta syötteestä lasketaan ulostuloarvo [12]. Toinen vaihe on takaisin syöttävää (*eng. backward propagation phase*) eli siinä lasketaan ideaalisen eli totuudenmukaisen datan informaation ja toisaalta verkon antaman ulostulodatan informaation välistä eroavaisuutta [12]. Tämän jälkeen opittavia parametreja, kuten kerneleitä ja painokertoimia, säädetään vastavirta-algoritmin (*eng. back propagation algorithm*) mukaan [58].

Konvoluutioverkon viimeisellä, luokittelevalla kerroksella tappiofunktio (*eng. loss function*) laskee ennustusvirheen (*eng. prediction error*), mikä tarkoittaa sitä, kuinka hyvin ennustus on osunut oikein verrattuna todelliseen arvoon ja tätä virhetä käytetään edellä kuvatussa takaisin syöttävässä mallin säätelyssä [17]. Käytetty tappiofunktio riippuu luokitteluongelman tyypistä ja esimerkiksi moniluokitteluongelmissa käytetään useimmiten cross-entropy -funktiota [58]. Cross-entropy -funktio laskee todennäköisyysjakauman ja todellisen jakauman välisen eron ja saakottaa sitä enemmän mitä suurempi ero on [32].

Tappiofunktioiden lisäksi konvoluutioneuroverkoissa käytetään usein optimoijaa (*eng. optimizer*). Optimoijaa tarvitaan erityisesti laskennallisesti haastavassa ope-

tusvaiheessa, jotta saadaan optimaaliset parametrit laskettua kohtuullisessa ajassa [32]. Usein käytetty optimointialgoritmi verkon opetuksessa on gradienttilasku (*eng. gradient descent*), joka iteratiivisesti päivittää opittavia parametreja minimoidakseen tappiota [58]. Gradientti-pohjainen oppimisalgoritmi voi olla tyypiltään erä (*eng. batch gradient descent*), osajoukko (*eng. mini batch gradient descent*) tai stokastinen (*eng. stochastic gradient descent*) menetelmä [17]. Toinen käytetty optimointialgoritmi on Adam (*Adaptive Moment Estimation*), joka laskee mukautuvan oppimisnopeuden jokaiselle verkon parametrille ja on muistiltaan tehokkaampi kuin monet muut algoritmit ja tarvitsee siten vähemmän laskentatehoa [17].

Hyperparametri	Esimerkkiarvo	Selitys
Oppimisnopeus	0.001	Kuinka usein painokertoimia päivitetään
Epochs	150	Koko opetusdatan läpikäynti kierroksina
Aktivaatiofunktio	ReLU	Muuntaa datan paremmin luokiteltavampaan muotoon
Mini-batch	12	Osajoukkokokoo, jonka jälkeen tehdään parametrien päivitys
Alustusmenetelmä	glorot_uniform	Kerroksen painokertoimien alustusmenetelmä
Optimoija	Adam	Apufunktio, jota tarvitaan optimaalisten parametrien laskemiseksi kohtuullisessa ajassa
Filtterit	[6, 12, 24, 24]	Filttereiden määrä konvoluutiokerroksilla
Testikoko	0.15	Missä suhteessa opetus- ja validointidata jaetaan

Taulukko 4.1: Esimerkkejä useimmin käytetyistä hyperparametreista ja niiden selitykset lyhyesti.

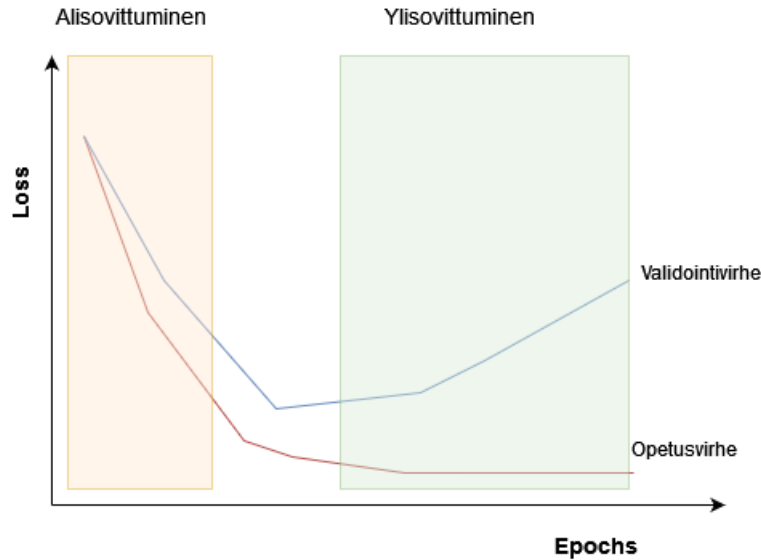
Aktivaatiofunktion, optimoijan ja tappiofunktion lisäksi CNN-verkossa on monia muitakin hyperparametreja. Hyperparametreiksi sanotaan sellaisia muuttujia, jotka määritellään ennen kuin varsinainen verkon opetus aloitetaan eivätkä ne kernelien tavoin ole opeteltavissa [58]. Taulukossa 4.1 on käytyä läpi joitain yleisiä

hyperparametreja. Hyperparametreista tärkeimpiä on oppimisnopeus (*eng. learning rate*), joka kuvastaa sitä, kuinka usein verkon painokertoimia päivitetään [32]. Liian pieni oppimisnopeus tarvitsee enemmän opetuskiertoja (*eng. epocheja*), jotta se pääsisi optimaaliseen ratkaisuun ja liian suuri oppimisnopeus taas voi viedä ratkaisun täysin harhaan optimaalisesta [17]. Termillä *epoch* tarkoitetaan puolestaan opetusvaiheen kierrosten lukumäärä, jolloin koko opetusaineisto viedään verkon läpi [32]. *Epochien* lukumäärää voidaan pitää sopivana, kun opetusdatan ja validointidatan välinen ero tarkkuudessa (*eng. accuracy*) on pieni [32].

Painokertoimien alustus (*eng. initialization*) on myös yksi tekijä, jota voidaan vaihdella ja joka siten voi olla hyperparametrina. Painokertoimia ei kannata alustaa nolllaksi, koska silloin menetetään takaisinoppimalla saatu tieto, esimerkiksi vastavirta-algoritmeista [17]. Sen sijaan usein tehdäänkin niin, että painokertoimia ei alusteta samalla tavalla vaan ne satunnaistetaan käyttämällä jotain painokertoimien alustusmekanismia [17]. Lisäksi puhutaan usein *batch* koosta tai *mini-batch*, jolla tarkoitetaan sitä, että opetusdata jaetaan useisiin pienempiin osiin (eli *mini-batch*) ja jokaisen *mini-batchin* jälkeen päivitetään mallin parametreja [17].

Syväoppimisessa puhutaan usein ylisovittumisesta (*eng. overfitting*). Tällä tarkoitetaan tilannetta, jossa neuronien määrää lisätään ja ongelma tulee monimutkaisemmaksi ja tällöin voi käydä niin, että malli sovittuu liian hyvin harjoitusdataan [42]. Tästä on seurauksena se, että mallin tunnistuskyky ennennäkemätöntä dataa kohtaan heikkenee eli se ei yleisty tarpeeksi hyvin [58]. Kuvassa 4.7 voidaan nähdä, kuinka ylisovittuminen alkaa siinä kohdalla, kun mallin opetusvirhe ja validointivirhe erkanevat toisistaan ja validointivirhe alkaa kasvaa. Alisovittuminen (*eng. underfitting*) on puolestaan sitä, että mallin tunnistuskyky on liian heikko [58]. Paras tapa ehkäistä ylisovittumista on lisätä opetusdatan määrää, sillä malli, joka on opetettu suuremmalla aineistolla, usein myös yleistyy paremmin [58]. Ylisovittumista estämään voidaan käyttää myös niin kutsuttua *dropout*-kerrosta, jossa osa neurooneista suljetaan pois satunnaisesti ja se määritellään tietyllä ennalta määritellyllä todennäköisyydellä [12]. Muita käytettyjä metodeja ylisovittumisen ehkäisemiseksi on esimerkiksi datan augmentointi eli datan lisääminen tai laajentaminen keinotekoisesti sellaisissa tilanteissa, joissa ei ole tarpeeksi aineistoa saatavana [17]. Datat augmentointia voi olla esimerkiksi kuvadatan satunnainen kääntäminen, kiertäminen, rajaaminen ja pyyhkiminen, jolloin saadaan toisistaan poikkeavia havaintoja aineistoon [58]. Yksi vaihtoehto ylisovittumisen estämiseen on käyttää osajoukon normalisointia (*eng. batch normalization*), mikä tarkoittaa sitä, että jokaisen kerrok-

sen ulostulo normalisoidaan vähentämällä keskiarvo ja jakamalla keskihajonnalla [17].



Kuva 4.7: Ylisovittumisen ja alisovittumisen suhde validointi- ja opetusvirheeseen artikkelin [58, s.620] mukaan, muokattu.

4.3 Syväoppimismallin suorituskyvyn analysoinnista

Useimmiten syväoppimista käyttävissä sovelluksissa halutaan saada ratkaisu luokitteluongelmaan. Luokitteluongelmat voidaan jakaa ryhmiin riippuen siitä, kuinka monta luokkaa on kyseessä. Binäärinen (*eng. binary*) luokitteluongelma on kyseessä, kun annettu syöte on laitettava yhteen luokkaan ja valittavana on kaksi eri luokkaa, jotka eivät ole päällekkäisiä [47]. Moniluokittelu (*eng. multi-class*) ongelmassa puolestaan luokkia on enemmän kuin kaksi, mutta luokat ovat erillisiä ja syöte voi kuulua vain yhteen luokkaan [47].

Sekaannusmatriisi (*eng. confusion matrix*) on taulukko, jossa voidaan kuvata luokittelun mahdolliset tulokset. Taulukossa 4.2 on esitettyä yleinen esitys binääri-luokittelun sekaannusmatriisista. Artikkelin [47] mukaan sekaannusmatriisin tulosvaihtoehdot voidaan jakaa neljään luokkaan. Oikea positiivinen (*eng. true positive, tp*) tarkoittaa sitä, että syöte on luokiteltu positiiviseksi, kun se on ollut positiivinen. Oikea negatiivinen (*eng. true negative, tn*) puolestaan tarkoittaa sitä, että syöte on luokiteltu oikein negatiiviseksi, kun se on ollut negatiivinen. Väärä positiivinen (*eng. false positive, fp*) tarkoittaa sitä, että syöte on luokiteltu positiiviseksi, vaikka

Luokka	Luokiteltu positiiviseksi	Luokiteltu negatiiviseksi
positiivinen	oikea positiivinen (<i>eng. true positive, tp</i>)	väärä negatiivinen (<i>eng. false negative, fn</i>)
negatiivinen	väärä positiivinen (<i>eng. false positive, fp</i>)	oikea negatiivinen (<i>eng. true negative, tn</i>)

Taulukko 4.2: Sekaannusmatriisi binääriluokittelulle Sokolova et al. [47, s.3] mukailen.

se todellisuudessa oli negatiivinen. Väärä negatiivinen (*eng. false negative, fn*) taas merkitsee sitä, että syöte on luokiteltu negatiiviseksi vaikka se todellisuudessa oli positiivinen.

Binääriluokittelun sekä moniluokittelun suorituskyvyn mittareina käytetään useita eri parametreja. Tarkkuus eli virheettömyys (*eng. accuracy*) kuvastaa todennäköisyytenä sitä, kuinka hyvä malli on ennustamaan syötteen luokan oikein [46]. Se voidaan määrittää seuraavasti

$$accuracy = \frac{tp + tn}{tp + fp + fn + tn} \quad , \quad (4.3)$$

missä tp on oikeat positiiviset, tn oikeat negatiiviset, fp väärät positiiviset ja fn väärät negatiiviset [46].

Sensitiivisyys (*eng. sensitivity*), josta käytetään myös nimeä *recall* tarkoittaa todennäköisyyttä, jolla malli tunnistaa positiiviset syötteet oikein [47]. Spesifisyys (*eng. specificity*) puolestaan tarkoittaa sitä, kuinka hyvin negatiiviset syötteet tunnistetaan [47]. Nämä voidaan määrittää seuraavasti

$$sensitivity = \frac{tp}{tp + fn} \quad , \quad (4.4)$$

$$specificity = \frac{tn}{fp + tn} \quad , \quad (4.5)$$

missä tp on oikeat positiiviset, tn oikeat negatiiviset, fp väärät positiiviset ja fn väärät negatiiviset [46].

Toinen tarkkuuteen viittaava määre on *precision*, joka voidaan myös suomentaa osumatarkkuudeksi. Osumatarkkuudella kuvataan algoritmin ennustusvoimaa eli sitä, kuinka hyvin se ennustaa joko negatiivisen tai positiivisen arvon [46]. Erona virheettömyyteen (*eng. accuracy*) on se, että *accuracy* arvioi kokonaisuudessaan algoritmin ennustusvoimaa [46]. *Precision* voidaan määrittää seuraavasti

$$precision = \frac{tp}{tp + fp} \quad (4.6)$$

missä tp on oikeat positiiviset ja fp väärät positiiviset [46].

Viimeisenä mainittakoon $F1$ -score, joka on tarkkuuden (precision) ja sensitiivisyyden (recall) harmoninen keskiarvo ja se voidaan määritellä seuraavasti

$$F1 = \frac{2tp}{2tp + fp + fn} \quad (4.7)$$

missä tp on oikeat positiiviset, fp väärät positiiviset ja fn väärät negatiiviset [34].

$F1$ -mittaa käytetään tyypillisesti binääriluokittelun yhteydessä, kun toinen luokista on harvinaisempi. Tätä käytetään esimerkiksi syövän tunnistuksessa, koska sekä väärin positiivisten että väärin negatiivisten määrä on kriittistä eli niitä ei saisi tulla [34]. Yhteenveto suorituskvyn mittareista on taulukossa 4.3.

Mittari	Englanninkielinen nimi	Selite
Virheettömyys	Accuracy	Todennäköisyys, jolla malli ennustaa syötteen luokan oikein
Sensitiivisyys	Recall	Todennäköisyys, jolla malli tunnistaa positiiviset syötteen oikein
Spesifisyys	Specificity	Todennäköisyys, jolla malli tunnistaa negatiiviset syötteen oikein
Osumatarkkuus	Precision	Ennustusvoima, kuinka hyvin ennustaa joko negatiivisen tai positiivisen arvon
F1-mitta	F1-score	Osumatarkkuuden ja sensitiivisyyden harmoninen keskiarvo

Taulukko 4.3: Yhteenveto suorituskvyn mittareista

5 Liikkeiden tunnistus tutkan ja syväoppimisen avulla

Tutkalla vaikuttaisi olevan hyviä ominaisuuksia ajatellen liikkeiden tunnistusta. Li et al. [31] tiivistävätkin artikkelissaan tutkan edut neljään kohtaan. Ensimmäkin tutka on toimintavarma erilaisissa valaistus- ja sääolosuhteissa. Toiseksi se ei loukkaa ihmisen yksityisyydensuojaa, sillä sen sijaan, että se kuvaisi suoraan kohdetta, sen toiminta perustuu takaisin tulevien signaalien tuomaan tietoon kohteen nopeudesta ja etäisyydestä. Kolmanneksi tutka pystyy erottamaan jopa seinien läpi kohteita, jolloin sen käyttömahdollisuudet laajenevat. Neljänneksi se ei vaadi minkäänlaisten sensorien pukemista päälle, jolloin se on myös käyttäjäystävällisempi. Näistä syistä tutkapohjaisten ratkaisujen etsiminen liikkeiden tunnistukseen onkin ollut vilkasta viime aikoina.

Syväoppiminen on tuonut myös uusia mahdollisuuksia tehdä luokitteluja ja liikkeiden tunnistuksia tutkan tuottamasta datasta. Li et al. [31] kertovat artikkelissaan aktiivisuuden tunnistukseen käytettävän tyypillisesti seuraavia malleja: konvoluutioneuroverkkoa (CNN), takaisinkytkettyä neuroverkkoa (*eng. recurrent neural network*, RNN), autoenkoodereita (*eng. auto-encoder*, AE) tai erilaisia hybridiratkaisuja. Taulukossa 5.1 on koottuna lyhyesti näiden mallien ominaisuuksia. Konvoluutioneuroverkkoja käsitellään tarkemmin luvussa 4.2. Lyhyesti kerrattuna konvoluutioneuroverkko on monikerroksinen syväoppimismalli, joka voi oppia erilaisia ominaisuuksia syötetystä, usein kuvamuotoisesta, datasta ja jossa aloituskerrokset oppivat ja erottelevat ylemmän tason piirteitä pienemmällä abstraktiolla ja syvemmat kerrokset oppivat ja erottelevat taas matalamman tason piirteitä korkeammalla abstraktiolla [17].

Autoenkooderi on ohjaamattoman oppimisen neuroverkko, joka koostuu tyypillisesti kolmesta kerroksesta: sisääntulo kerroksesta, piilotetusta kerroksesta sekä dekodauskerroksesta [9]. Verkon ideana on ensin asettaa tavoitetulokset yhtä suureksi syötekerroksen kanssa, jonka jälkeen verkko yrittää rekonstruoida syötteen ulostuloksi ymmärtämällä ja muodostamalla arvion siitä funktiosta, jolla ulostulo saadaan. Piilotettu kerros pakotetaan oppimaan paras esitys sisääntulosta rekonstruktion avulla. Autoenkoodereista on olemassa erilaisia variaatioita, kuten harva (*eng.*

sparse autoencoder) ja kontraktoiva (*eng. contractive autoencoder*) autoenkooderi.

Takaisinkytketty neuroverkko tarkoittaa sellaista neuroverkkoa, jossa on yhteys edellisiin kerroksiin, jolla pystytään antamaan palautetta edellisille kerroksille [9]. Yksi esimerkki takaisinkytketystä neuroverkosta on LSTM (*eng. long short-term memory*), joka koostuu muistilohkoista, joiden sisältöä ja sitä, mitä niihin tallennetaan, luetaan ja poistetaan, hallinnoivat portit [42].

Malli	Ominaisuuksia
CNN	erinomainen piirteiden erottelija, taajuus- ja nopeustietojen saaminen eroteltuihin piirteisiin
RNN	ajallisen suhteen tutkiminen datasta, erilaisia variaatioita kuten LSTM
Autoenkooderi	eteenpäin syöttävä neuroverkko, joka oppii ohjaamattomalla tavalla
Hybridi	yhdistelemällä joidenkin syväoppimismallien parhaimpia ominaisuuksia

Taulukko 5.1: Liikkeiden tunnistukseen käytettyjä syväoppimismalleja ja niiden ominaisuuksia Li et al. [31] mukaan, muokattu.

Li et al. [31] esittävät katsauksessaan seuraavan kappaleen mukaisen jaottelun syväoppimisen käytöstä tutkan tuottamaan dataan. Tutkasta saatava data sisältää tietoa ajasta, etäisyydestä ja Doppler-taajuudesta, joten syväoppimismallin tulisi sopia käsittelemään juuri näitä ominaisuuksia. Tutkan palauttamaa signaalia voidaan tutkia ulottuvuuksien mukaan kolmella eri tavalla, nimittäin 3-, 2- ja 1-ulotteisina syötteinä. Tutkan signaaleja voidaan muuntaa 3-ulotteiseksi aika-etäisyys-Doppler kuutioiksi, jossa Doppler-vaikutuksen avulla voidaan tunnistaa kohteen radiaalinen nopeus. Tämä saatu 3D videonkaltainen näkymä kuvastaa kohteen ajallista aktiivisuutta. Toiseksi tutkan signaaleja voidaan esittää myös 2-ulotteisena aika-Doppler, aika-etäisyys tai etäisyys-Doppler -karttoina. Kolmantena voidaan tutkan data esittää yksiulotteisena esityksenä, jossa aika-etäisyys-Doppler -data projektoidaan etäisyys profiiliksi, jolloin saadaan korkea resoluutio etäisyysprofiili (*eng. high resolution range profile, HRP*).

5.1 Syväoppimismallit tutkan 3D-muotoisen datan kanssa

Artikkelissa [31] kuvataan 3D-muotoista tutkan dataa seuraavasti. Etuna kolmiulotteisessa datassa on se, että siihen saadaan mukaan sekä tilallinen että ajallinen näkökulma. Verrattuna 1D- ja 2D-signaaleihin aika-etäisyys-Doppler signaalit sisältävät lähes kaiken tiedon, mitä tutka voi saada liikkeestä. Useimmiten käytetty syväoppimismalli 3D-muotoiseen signaalin kanssa on konvoluutioneuroverkot.

Vuonna 2022 ilmestyneessä artikkelissa Li et al. [30] tutkivat yhteensä 11 eri liikkeen (makaaminen, makuuasennosta nouseminen, kyykky, seisominen, ylös nouseminen, istuminen, kävely, juokseminen, nosto ja kumarrus) tunnistamista aineistosta, johon oli kerätty 10 eri henkilön liikkeitä 77 GHz:n millimetriaaltotutkalla havaittuna. Datan augmentoinnilla eli muuntamisella ja lisäämisellä saatiin datasetin kokoa kasvatettua 7-kertaiseksi ja samalla lisättyä systeemin toimintavarmuutta. Yhteensä 1600 kaatumistapahtumaa ja 12000 muuta liikettä kerättiin tutkimuksen aikana. Koska tutkimuksessa haluttiin ottaa mukaan aika, etäisyys ja Doppler, konvoluutioneuroverkoksi muodostettiin 3-ulotteinen konvoluutioneuroverkko, jossa rakenteen muodostivat syötekerroksen jälkeen kolme konvoluutiokerrosta ja niiden jälkeen tulevat yhdistävät kerrokset. Datan augmentaatiolla saatiin tarkkuudeksi 98.8% ja ilman augmentaatiotakin päästiin 97.0% tarkkuuteen..

Ihmisen aktiivisuutta tutkivat Hernangomez et al. [20], jotka keräsivät yhteensä 1188 liikenäytettä 11 koehenkilön suorittamana. Tutkittavat liikkeet olivat istuminen, kävely, juoksu, nyrkkeily paikallaan ja kävellen eteenpäin, vilkutus, makuulle meneminen ja istumaan nouseminen makuulta. Lisäksi aineistoon otettiin ylimääräinen luokka, joka oli ikkunaverhojen liikehdintä. Tutkan tuottamalle raakadatalle tehtiin ensin normalisointia. Tutkimuksessa testattiin kolmea erilaista konvoluutioneuroverkko mallia. Ensimmäinen malli koostui viidestä 2-ulotteisesta konvoluutiokerroksesta ja kahdesta yhdistävästä kerroksesta, jolle syötettiin Dopplerspektrogrammi ilman etäisyystietoa. Toisena mallina toimi 3-ulotteisia konvoluutiokerroksia käyttävä neuroverkko, jolle syötteeseen otettiin edellisen lisäksi myös etäisyystieto. Viimeisenä mallina oli kahden ensimmäisen yhdistelmä, jossa malli haarautui kahteen osaan ja yhdistettiin lopussa yhdeksi ulostuloksi. Tutkijat totesivat 3-ulotteista etäisyys-Doppler-karttaa käyttävän mallin olevan paras ennustuskyvyltään ollen kuitenkin 4 kertaa suurempi kuin 2-ulotteinen malli.

Tutkimuksessaan Chen et al.[6] puolestaan painottivat 3D-konvoluutioneuroverkon tärkeää roolia siinä, että tutkan saama tieto sekä ajasta että paikasta on tärkeää aktiivisuuden tunnistuksessa. He tutkivat yhteensä 6 eri liikkeen (kaatuminen, hyppi-

minen, hölkkä, kyykistys, askellus, kävely) tunnistusta 5.8 GHz:n tutkalla havaittuna. Keskimäärin tarkkuudeksi saatiin 90%:n tarkkuus, mutta tutkijoiden lisäämällä ajallisella painokertoimella saatiin tarkkuutta nostettua jopa 96%:iin.

5.2 Syväoppimismallit tutkan 2D-muotoisen datan kanssa

Li et al. [31] jatkavat katsaustaan kaksiulotteisen tutkan signaalin kuvailulla seuraavasti. Vaikka 3D-signaalit tutkasta sisältävät paljon tietoa, niiden käyttö syväoppimisen keinoin vaatii paljon prosessointia ja laskentatehoa. Niinpä usein käytetäänkin 2D-signaalia, joista muotoina voi olla aika-Doppler -kartta, aika-etäisyys -kartta ja etäisyys-Doppler -kartta. Yleensä näitä kaksiulotteisia signaaleja käsitellään kuvina, joten näissäkin tapauksissa konvoluutioneuroverkot ovat myös käytetyin syväoppimismenetelmä.

Käytetyin vaihtoehto kaksiulotteisessa datassa on aika-Doppler -kartta, jota usein kutsutaan myös nimellä mikro-Doppler -piirre (*eng. micro-Doppler signature*). Aika-Doppler -kartan etuna voidaan pitää sitä, että se saadaan helposti raakadatasta käyttämällä esimerkiksi STFT-tekniikkaa (*eng. short time frequency transform*) ja kuvat ovat usein myös intuitiivisia ja ymmärrettäviä, koska erilaiset liikkeet aiheuttavat erilaisen muutoksen kartassa [31]. Esimerkiksi Trommel et al. [54] käyttivät mikro-Doppler -spektrogrammia tutkiessaan useamman henkilön kävelyä. Tutkijat käyttivät eri taajuuksia tutkimuksessaan ja huomasivat systeemin pystyvän tunnistamaan kävelyn myös matalammilla taajuuksilla. Myös Kim et al. [26] tekivät tutkimuksen, jossa käytettiin aika-Doppler spektrogrammia tunnistamaan ihminen muista kohteista (auto, koira ja hevonen) ja siinä saavutettiinkin noin 97%:n tarkkuus. Lisäksi 7 eri liikkeen (juoksu, kävely, kävely kepin kanssa, ryömiminen, nyrkkeily ja kävely eteenpäin, nyrkkeily paikallaan ja istuminen liikkumatta) tunnistamista konvoluutioverkolla tutkittiin saaden siinäkin noin 90%:n tarkkuuden.

Toisena vaihtoehtona on käyttää kaksiulotteisen tutkan signaalin tutkimisessa aika-etäisyys -karttaa. Tämä sisältää aikariippuvan esityksen kohteen ja tutkan välisestä etäisyydestä ja ihmisen liikkeessa eri ruumiinosilla on erilaiset suhteelliset etäisyydet, jolloin voidaan tehdä päätelmiä ihmisen liikkeistä [31]. Esimerkiksi tutkimuksessa [40] päädyttiin käyttämään aika-etäisyys -tietoa Doppler-tiedon sijaan, jotta systeemistä saataisiin toimintavarma myös sellaisissa tilanteissa, joissa radiaalinen nopeus oli pieni. Yhteensä seitsemää eri liikettä käytettiin tutkimuksessa ja lopulliseksi tarkkuudeksi liikkeen tunnistuksessa saatiin noin 95%:n tarkkuus.

Kolmantena vaihtoehtona kaksiulotteisen datan tutkimisessa tulevat etäisyys-Doppler -esitykset, jotka kuvaavat etäisyyden ja Doppler-tiedon muuttumista ajan suhteen, jolloin eri ruumiinosien liikkuminen ja kohteen paikantaminen on mahdollista, samoin kuin useamman kohteen seuraaminen [31]. Tutkimuksessa [23] käytettiin etäisyys-Doppler -spektrogrammeja ihmisen kaatumisen tunnistamiseen. Syväoppimismalliksi päädyttiin käyttämään kahta pinottua autoenkooderia ja luokitteluun logistista regressiota. Kaatumistapahtuma erotettiin istuutumisesta, kävelystä ja kumartamisesta yli 90%:n todennäköisyydellä.

5.3 Yksiulotteisen datan käyttö syväoppimisen keinoin

Kuten edellä todettiin, projektoimalla aika-etäisyys-Doppler -kuutio etäisyydeksi saadaan HRRP eli korkea resoluutio etäisyysprofiili (HRRP) [31]. Tutkimuksessa [29] toteutettiin liikkeiden tunnistussysteemi, *BiGRU*, jossa syöteenä annettiin tutkan HRRP-signaali. Syväoppimismallina käytettiin LSTM-pohjaista rakennetta. Sovelluksen tarkkuudeksi saatiin 97.5%. Tutkan yksiulotteista dataa voidaankin pitää myös eräänlaisena aikasarjana, joka vastaa esimerkiksi kiihtyvyyssanturista saatavaa tietoa, joten myös esimerkiksi RNN ja LSTM kävisivät hyvin myös tällaisen datan analysointiin [31].

5.4 Hybridiratkaisuja liikkeen tunnistukseen

Eri syväoppimismalleilla on omat hyvät ja huonot puolensa. Siksi voi olla hyödyllistä yhdistää useamman mallin parhaimpia puolia. Hybridiratkaisuissa useimmiten on yhdistettynä konvoluutioneuroverkko (CNN) ja takaisinkytketty neuroverkko (RNN), koska CNN tarjoaa malliin tilallisen ulottuvuuden (*eng. spatial relationship*) ja RNN tuo taas ajallisen ulottuvuuden (*eng. temporal relationship*) [31].

Tutkimuksessaan Zhang et al. [59] toteuttivat käden liikkeitä tunnistavan systeemin, jossa syväoppimismallina oli yhdistelmä CNN- ja RNN-verkkoja. Syöte vietiin ensin konvoluutioneuroverkolle, jossa oli kolme 3-ulotteista konvoluutiokerrosta. Tämän jälkeen konvoluutioneuroverkon ulostulo vietiin takaisinkytkettyyn verkkoon, joka käytti LSTM (*eng. long short-term memory*) rakennetta. Lopuksi varsinainen luokittelu tapahtui käyttäen CTC-algoritmia (*connectionist temporal classification*), joka on tutkijoiden kehittämä dynaamiseen liikkeen luokittelun algoritmi eikä tarvitse syöteen segmentointia. Systemillä saavutettiin jopa 96%:n tarkkuus

tunnistuksessa.

Toisessa tutkimuksessa Ding et al. [11] toteuttivat konvoluutioneuroverkkoja ja LSTM-mallia käyttävän hybridiratkaisun ihmisen liikkeiden tunnistukseen. Tutkijat hyödynsivät monipuolisesti tutkan tarjoamaa tietoa, sillä aika-etäisyys -kartta ja aika-Doppler -kartta syötettiin kukin omalle 1D-konvoluutiokerroksista muodostuvalle verkolle sekä niiden jälkeen tuleville LSTM-kerroksille ja etäisyys-Doppler -signaali vietiin 2D-konvoluutiokerroksista muodostuvalle verkolle. Yhdessä nämä kaksi 1D-konvoluutioneuroverkon ja LSTM-verkon muodostamaa verkkoa ja yksi 2D-konvoluutioneuroverkko muodostivat systeemin piirteiden erottelijan. Näiden jälkeen verkkojen tiedot yhdistettiin ja lopulta luokittelu tapahtui yhdistävällä kerroksella ja Softmax-funktion avulla. Tutkimuksessa saavutettiin lupaavia tuloksia liikkeiden tunnistukseen.

5.5 Yhteenveto liikkeiden tunnistuksesta tutkan ja syväoppimisen avulla

Edellä kuvattiin ja luokiteltiin liikkeiden tunnistusta tutkan tuottamasta datasta sen perusteella, käyttääkö se 1-, 2- vai 3-ulotteista dataa syötteenään ja toisaalta, mitä syväoppimismallia se käyttää. Lisäksi mainittiin muutamia ratkaisuja, joissa on käytetty useampia eri syväoppimismalleja. Taulukossa 5.2 on koottuna artikkelien keskeisimmät tiedot. Konvoluutioneuroverkot vaikuttavat olevan eniten käytetty malli liikkeen tunnistuksessa, mutta jokaisella mallilla on omat hyvät ja huonot puolensa. Artikkelissa [56] pohditaan syväoppimismallin valintaa seuraavasti. RNN ja LSTM sopivat tilanteisiin, joissa on lyhyitä aktiviteetteja, joilla on luonnollinen järjestys, kun taas CNN on parhaimmillaan tilanteissa, joissa tarvitaan pidempiaikaisten toistuvien liikkeiden tunnistusta. Tämä johtuu RNN-mallin kyvystä tulkita ajallista yhteyttä tutkan datasta ja CNN-malli taas oppii syviä yhteyksiä ja piirteitä rekursiivisella mallilla.

Li et. al [31] pohtivat liikkeiden tunnistuksen suuntaa tulevaisuudessa seuraavan kappaleen mukaisesti. Nykyisten tutkimusten pääpaino on ollut tunnistaa yksinkertaisia liikkeitä, kuten kävely tai istuutuminen, mutta monimutkaisempien liikkeiden, kuten ruoanlaitto, joka sisältää useita samanaikaisia liikkeitä, tunnistukseen tulisi myös käyttää resursseja. Hybridimallit, jotka yhdistävät useampia syväoppimisen menetelmiä, voisivat olla ratkaisu monimutkaisempien liikkeiden tunnistukseen. Samoin tutkan 2D-muotoisen datan tutkiminen on ollut suosittua, mutta myös

3D- ja 1D-data voisivat tuoda uutta näkökulmaa erityisesti monimutkaisten liikkeiden tunnistuksessa. Lisäksi tulisi ottaa huomioon, millaisissa olosuhteissa ja paikoissa liikkeiden mittausta todellisuudessa tehdään. Useimmiten tutkan data tulee jatkuvana syötteenä ja sen käsittelyyn tarvitaan toimintavarmaa ja verkossa olevaa käsittelyä ja luokittelua. Niinpä erityisesti ns. kevyet syväoppimismallit ovat tulevaisuudessa erityisen mielenkiinnon kohteena. Toisaalta ihmisen liikkeet todellisuudessa tapahtuvat peräkkäisinä ja vapaasti, ei niinkään tutkimustilanteessa annettujen käskyjen mukaisesti. Niinpä tutkimukset siitä, miten liikkeitä erotellaan eli segmentoidaan ja tunnistetaan samanaikaisesti datavirrasta, ovat tärkeitä, jotta tunnistussovellukset voisivat toimia myös tosielämässä. Ongelma on myös se, että usein syväoppimismallit tarvitsevat suuren määrän luokiteltua dataa, jotta malli saadaan opetettua ja yleistymään ennennäkemätöntä dataa vastaan. Tämä voi joissain tilanteissa olla iso haaste, niinpä ohjaamattoman oppimisen, joka ei tarvitse suurta opetusdataa, käyttäminen tutkimuksissa voisi olla aiheellista.

Artikkeli	Käytetty malli	Tutka	Taajuus	Syöte
[30]	CNN	FMCW	77 GHz	3D
[20]	CNN	FMCW	60 GHz	3D
[6]	CNN	FMCW	5.8 GHz	3D
[54]	CNN	CW	8 GHz	2D
[26]	CNN	Doppler	2.4 GHz	2D
[40]	CNN	UWB	3.9 GHz	2D
[23]	AE	FMCW	25 GHz	2D
[29]	LSTM	CW	4 GHz	1D
[59]	CNN ja LSTM	FMCW	24 GHz	3D
[11]	CNN ja LSTM	FMCW	5.8 GHz	2D

Taulukko 5.2: Yhteenveto katsauksen artikkeleista.

6 Tutkimus

Tässä luvussa perehdytään työn empiiriseen osaan. Ensimmäinen aliluku 6.1 käsittelee työssä käytettyä PARrad-aineistoa [5]. Toisessa aliluvussa 6.2 käydään läpi tutkimuksen tavoitteet. Lopuksi kolmannessa aliluvussa 6.3 perehdytään tutkimuksen toteutukseen ja syväoppimismallin kehitykseen.

6.1 Tutkimuksen taustaa

PARrad (Patient Activity Recognition with Radar sensors) ¹ on tutkimusryhmän Bhavanasi et al. [5] keräämä aineisto potilaiden liikkeiden seurannasta. Tutkimus toteutettiin kahdessa eri ympäristössä, joista toinen oli oikeassa sairaalahuoneessa ja toinen sairaalahuoneeksi tehdyssä huoneessa. Kumpaankin tilaan oli sijoitettuna kaksi FMCW-tutkaa sekä yksi web-kamera. Kameraa käytettiin vain tutkimustilanteiden validointiin ja huomioiden tekemiseen. Tutkat käyttivät taajuuksia 77 GHz ja 60 GHz ja ne sijoitettiin eri asentoihin ja korkeudelle, jotta saatiin systeemistä luotettavampi ja enemmän dataa kerättyä. Tutka, joka käytti taajuutta 77 GHz, sijoitettiin seinälle katonrajaan ja 60 GHz:n tutka puolestaan sijoitettiin silmien korkeudelle samaan kohtaan, missä kamera sijaitti. Tutkimuksessa haluttiin selvittää, mikä on tutkimuksessa kehitellyn konvoluutioneuroverkkomallin tarkkuus sekä toisaalta, millainen sen toimintavarmuus on riippuen ympäristöstä, henkilön iästä, käytetyistä apuvälineistä sekä tutkan sijainnista.

Tutkimukseen osallistui yhteensä 29 henkilöä iältään 24 - 90 vuotta. Henkilöitä pyydettiin tekemään satunnaisesti 14 eri liikettä 10 minuuttia kestävässä session aikana. Liikkeet olivat seuraavat

- Huoneessa tapahtuvat liikkeet
 - kävely huoneeseen (*walk to room*)
 - kaatuminen lattialle (*fall on the floor of room*)
 - nousu ylös lattialta (*stand up from the floor*)

¹<https://www.imec-int.com/en/PARrad>

- kävely tuolille (*walk to chair*)
- istuutuminen tuolille (*sit down on chair*)
- nousu ylös tuolilta (*stand up from chair*)
- kävely sängylle (*walk to bed*)
- istuutuminen sängylle (*sit down on bed*)
- nousu ylös sängyltä (*stand up from bed*)
- Sängyssä tapahtuvat liikkeet
 - makuulle meneminen (*get in bed*)
 - sängyssä makaaminen (*lie in bed*)
 - kääntyminen sängyllä (*roll in bed*)
 - istuminen sängyllä (*sit in bed*)
 - sängystä nouseminen (*get out bed*)

Yllä olevista 14 liikkeistä yhdisteltiin samankaltaiset toiminnot kuten kävely huoneeseen, kävely sängylle ja kävely tuolille luokiteltiin kaikki kävelyksi. Lisäksi istuutuminen sängylle ja istuutuminen tuolille luokiteltiin istuutumiseksi sekä nouseminen ylös sängyltä ja nouseminen ylös tuolilta luokiteltiin ylös nousemiseksi. Näin saatiin 10 eri luokkaa liikkeille

- kävely
- kaatuminen lattialle
- nousu ylös lattialta
- nousu ylös
- istuutuminen
- makuulle meneminen
- sängyssä makaaminen
- kääntyminen sängyllä
- istuminen sängyllä

- sängystä nouseminen

Osa tutkittavista henkilöistä sairaalaympäristössä käyttivät liikkumisen apuna joko kävelykeppiä tai kävelytukea. Niinpä tästä ympäristöstä kerätyssä datassa on liikkeitä tehty joko ilman välineitä, niiden kanssa tai niiden yhdistelmällä. Kotiympäristössä tehdyissä mittauksissa ei ollut käytössä apuvälineitä. Lisäksi joissain tapauksissa kaatuminen lattialta ja sieltä ylös nouseminen ei ollut mahdollista huomioiden potilaan ikä ja terveys, joten tällaiset tapaukset jätettiin pois aineistosta. Myös sellaiset tapaukset, joissa liike kesti alle 2 sekuntia jätettiin pois lopullisesta aineistosta.

	Tutka 1	Tutka 2
Tyyppi	TI-xWR14xx	TI-xWR68xx
Taajuus	77 GHz	60 GHz
Maksimi kaistanleveys	4 GHz	4 GHz
Vastaanottavia antennoja	4	3
Lähettäviä antennoja	3	3
Frame rate	11.1 fps	11.1 fps
Range bins	93	93
Doppler bins	128	128
Maksiminopeus	4.01 m/s	5.11 m/s
Nopeuden resoluutio	0.06 m/s	0.08 m/s
Maksimietäisyys	22.50 m	22.50 m
Etäisyyden resoluutio	9.77 cm	9.77 cm

Taulukko 6.1: PARrad aineiston keruussa käytetyt tutkat ja niiden tärkeimpiä ominaisuuksia artikkelin [5] mukaan.

Tutkimuksessa saatu raakadata tutkasta käsiteltiin ensin kaksiuotteisella Fourier-muunnoksella, jolloin saatiin RD-map eli aika-etäisyys-Doppler -kartta ja tämän jälkeen RD-signaali muunnettiin desibeleiksi (dB). RD-map on kolmiulotteinen ja koostuu etäisyydestä, Doppler-taajuudesta sekä ajasta. MD eli mikro-Doppler -signaali saadaan puolestaan, kun summataan RD-signaali etäisyyden yli. MD-signaali on siis kaksiuotteinen ja koostuu nopeudesta ja Doppler-taajuudesta. Datasta saatava näyte on sarja peräkkäisiä kehyksiä (*eng. frames*), jotka yhdessä muodostavat liikkeen. Artikkelin tutkimuksessa käytettiin tutkassa nopeutena 11.1 fps eli 11.1 kehystä sekuntia kohden. Lisäksi tutkijat huomasivat, että 40 kehystä riittää optima-

liseen näytteeseen. Mikäli liikkeen kesto oli alle 40 kehystä, mikä tarkoittaa noin 3.7 sekuntia, näytteen loppuun lisättiin viimeistä kehystä niin monta, jotta 40 saatiin täyteen. MD ja RD tietojen lisäksi aineisto sisältää aikaleimalla varustetun csv-tiedoston, jossa on riveittäin tieto ajanhetkestä sekä mikä liike on kyseessä. Tällä tavalla koko 10 minuutin jakso voidaan paloitella aikaleimojen perusteella näytteisiin ja luokitella ne. Artikkelin mukaan MD- ja RD-karttaa muokattiin sen verran, että kuvien dimensioiksi saatiin 63x98. Tutkimuksessa käytetyt sensorit ovat Texas Instrumentin MIMO (Multiple Input Multiple Output) käyttäviä tutkia [21]. Niiden tärkeimpiä ominaisuuksia on koottuna taulukkoon 6.1.

6.2 Tutkimuksen tavoitteet

Tämän työn tarkoituksena on tehdä syväoppimismalli, joka tunnistaa tutkasensorin havainnoimasta datasta ihmisen liikkeitä. Edellä kuvattiin työn pohjana oleva artikkeli [5]. On kuitenkin otettava huomioon, että tässä työssä käytetty PARrad aineisto on vain osa artikkelissa mainitusta aineistosta eli alkuperäisessä tutkimuksessa käytössä oli enemmän aineistoa, jolla malli opetettiin. Lisäksi tiedossa ei ole tutkimushenkilöistä tarkentavia tietoja kuten ikää tai sukupuolta eikä myöskään tutkimusympäristöä. Tarkoituksena ei ole siis tehdä samanlaista liikkeen tunnistusta ja mallia kuten alkuperäisessä työssä tehtiin. Sen sijaan tämän työn empiirinen osuus keskittyy luomaan mallin, joka olisi parametreiltaan optimaalinen tunnistamaan eri liikkeitä, mutta samalla myös kooltaan pieni, jotta sen voisi sijoittaa langattomaan sensoriverkkoon. Lisäksi tehdään vertailua kahden eri tutkan, joilla on eri taajuudet ja eri sijainti huoneessa, suorituskykyä eri testiaineistoilla.

6.3 Tutkimuksen toteutus

PARrad-aineisto on vapaasti käytettävissä ja ladattavissa osoitteessa PARrad². Tämä aineisto on vain osa siitä yli 21 000 näytteen aineistosta, joka alkuperäisessä tutkimuksessa kerättiin. PARrad-datasetti sisältää yhteensä 24 eri henkilön liikkeitä kerättyä dataa. Aineisto on jaoteltu henkilöiden mukaan kansioihin ja jokaisesta henkilöstä on noin kolme eri kansiota sisältäen yhden noin 10 minuutin ajanjakson datasetin. Yksi datasetti, joka on kooltaan noin 10 Gb, koostuu aina yhdestä h5-tiedostosta, joka sisältää kummastakin tutkasta noin 6684 kehyksen pituisen mikro-

²<https://www.imec-int.com/en/79GHz-140GHz-radar-systems/parrad>

Doppler ja etäisyys-Doppler -kartan. Lisäksi mukana on aikaleimalla varustettu csv-tiedosto suoritetuista liikkeistä.

	set000	set001	set002	set003
subject_00	O	O	O	-
subject_01	O	O	O	-
subject_02	O	O	O	E
subject_03	T	T	-	T
subject_04	O	O	E	O
subject_05	O	O	O	O
subject_06	O	E	O	-
subject_07	O	O	O	-
subject_08	O	O	O	O
subject_09	-	O	O	-
subject_10	O	O	E	-
subject_11	O	O	-	-
subject_12	T	T	-	-
subject_13	O	O	O	O
subject_14	O	E	O	O
subject_15	O	O	O	O
subject_16	O	O	-	-
subject_17	O	O	-	-
subject_18	O	O	-	-
subject_19	T	-	-	T
subject_20	O	O	O	O
subject_21	E	O	O	O
subject_22	O	O	O	-
subject_23	E	O	O	-

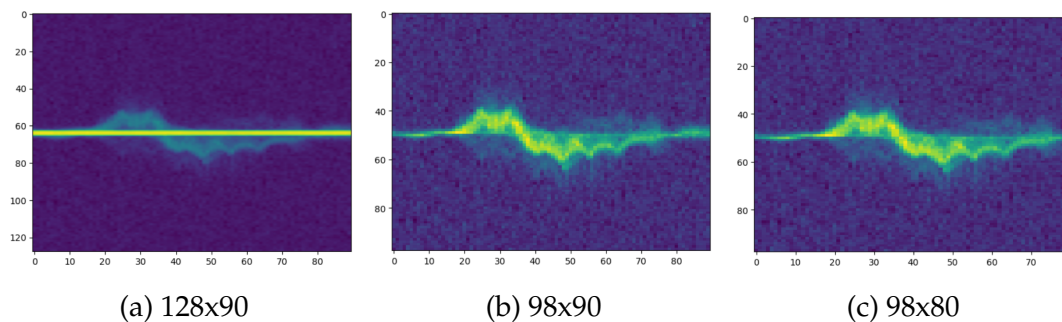
Taulukko 6.2: PARrad aineistosta valitut henkilöt ja aineisto. O = opetus, E = evaluointi ja T = testaus. Lisäksi - tarkoittaa, että datasettiä ei ollut, se oli puutteellinen tai että kyseistä datasettiä ei valittu.

Taulukossa 6.2 on esitettyinä valitut henkilöt ja aineistot tutkimusta varten. Koska artikkelissa tai aineiston kotisivuilla ei ollut tarkempia tietoja siitä, mikä on henkilön ikä, kummassa ympäristössä mittaus tehtiin tai oliko käytössä apuvälineitä,

henkilöiden ja datasettien valintaan ei ole ollut mitään erityisiä kriteereitä, vaan aineistoa pyrittiin saamaan tasaisesti kaikilta kohteilta, kuitenkin siten, että varsinaiseen testiaineistoon valittuja henkilöitä ei käytetty opetus- tai evaluointivaiheessa. Koottu aineisto jaettiin kolmeen ryhmään: opetusdata, evaluointidata ja testausdata. Opetusaineisto oli suurin ollen noin 80% koko ladatusta aineistosta. Validointiin eli opetuksen arviointiin käytetty data puolestaan otettiin opetusdatan sisältä siten, että 85% jäi varsinaiseen mallin opetukseen ja 15% opetuksen validointiin.

6.3.1 Datan esikäsittely

Jokainen ladattu datasetti käytiin läpi ja sieltä valittiin työssä käytettäväksi mikro-Doppler -signaali, josta saadaan syötteenä neuroverkolle kaksiulotteinen kuva. Tiedostoon oli valmiiksi tehtynä tutkan raakadatalle FFT eli fast Fourier-muunnos sekä etäisyys-Doppler (RD-map) kartan summaaminen, jotta saatiin mikro-Doppler -piirros (*micro-Doppler signature, MD-map*) eli MD-kartta. Alkuperäisessä datassa MD-kartan Doppler-ulottuvuuden pituus oli 128 koria (*eng. bins*). Dataa esikäsiteltiin samoin kuin alkuperäisessä artikkelissa oli tehty. Ensinnäkin MD-kartan keskellä pystysuunnassa olevat Doppler-korit (*eng. Doppler bins*) poistettiin, koska nämä kuvaavat paikallaan olevia kohteita, joilla ei ole nopeutta ja joilla ei siten ole merkitystä tunnistuksen kannalta. Uloimpien Doppler-korien suhteen artikkelissa oli empiirisesti todettu, että alun 13 ja lopun 14 koria voitiin myös poistaa ilman merkittävää tiedon menetystä. Kuvassa 6.1 on esitettynä vaiheittain kuvan esikäsittely.



Kuva 6.1: Esimerkkikuva datan esikäsittelystä: a-kuvassa lähtötilanne, kuvan koko (128x90), b-kuvassa on poistettu keskimmäiset kolme vaakariviä sekä poistettu 13 ja 14 vaakariviä sivuilta. Kuvat on käännetty transpoosilla 90 astetta, jotta ne soveltuvat neuroverkon syötteeksi paremmin.

Tutkan data pilkottiin aktiviteettien mukaan osiin käyttäen apuna csv-tiedostoa,

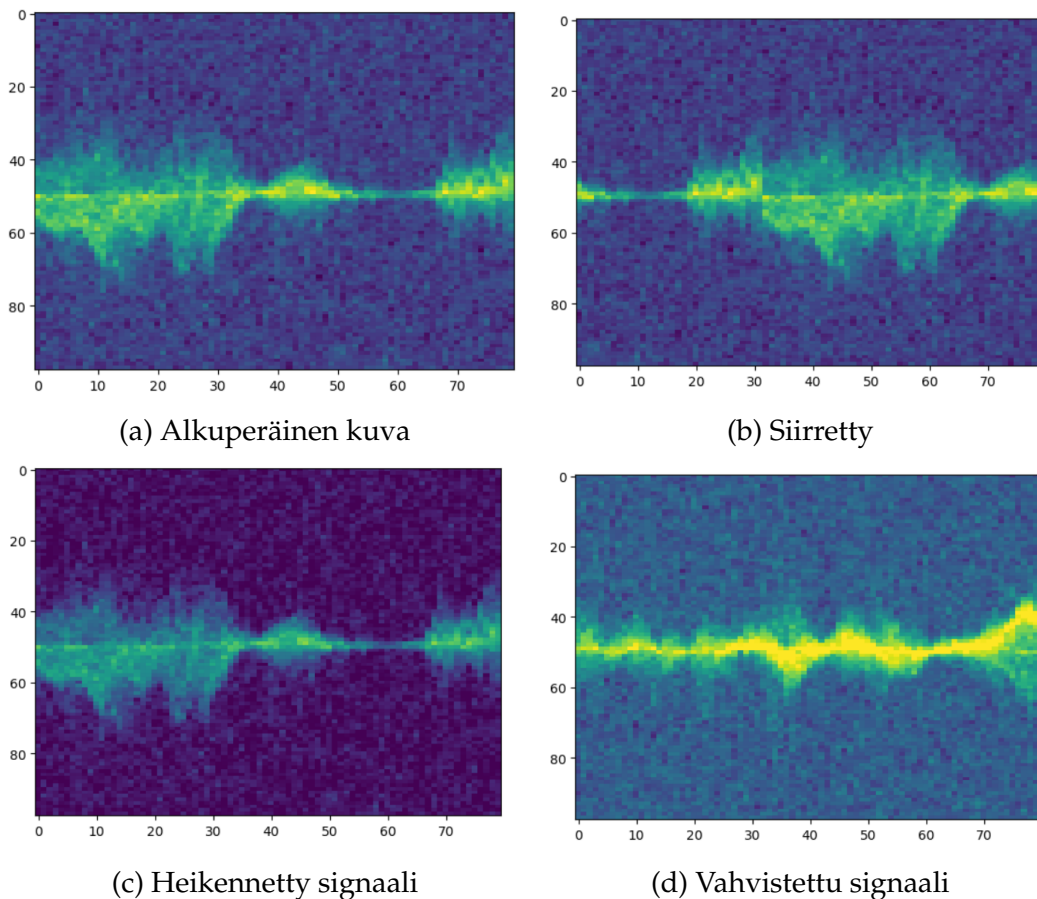
johon oli merkittynä aikaleima, jolloin toiminnon suoritus aloitettiin sekä mikä toiminto oli kyseessä. Näiden perusteella data pilkottiin ja näytteet laitettiin aktiviteettien mukaan omiin kansioihinsa. Dataa käsiteltiin vielä niin, että sellaiset näytteet, joiden pituus oli alle 2 sekuntia eli noin 23 kehystä, jätettiin pois. Alussa toimittiin niiden periaatteiden mukaan, jotka artikkelissa oli tehty eli jokaisesta liikkeestä otettiin ensimmäiset 40 kehystä. Kuitenkin eri malleja testattaessa huomattiin ongelmana olevan se, että 40 kehystä ei riittänyt kaikissa tapauksissa koko liikkeen saamiseen näytteisiin, sillä joidenkin henkilöiden liikkeet alkoivat vasta näytteen loppupuolella. Siksi päädyttiin ottamaan isompia näytteitä kuin alkuperäisessä artikkelissa, jotta saatiin edustavampia otoksia datajoukosta. Näytteen kokoa kasvatettiin 80 kehyksen pituiseksi, jolloin lopullisten kuvien kooksi saatiin esikäsittelyiden jälkeen (80,98). Lisäksi, mikäli näytteen pituus oli 23 ja 80 välillä, tehtiin kuten artikkelissa, eli otettiin näytteen viimeinen kehys ja monistettiin tätä, kunnes saatiin 80 kooksi.

Sen jälkeen kun aineisto oli saatu pilkottua edellä olevan prosessin mukaisesti, vuorossa oli testauksia erilaisilla malleilla ja hyperparametreilla. Testien aikana huomattiin kuitenkin, että oli vaikea saada mallia yleistymään, kun mallia testattiin validointidatalla. Opetusaineistosta ei tällöin saatu tarpeeksi kattavaa tietoa, jotta malli kykenisi tunnistamaan sekä validointi- että evaluointidatan. Chollet [8] kuvaa kirjassaan yleistämistä opetetun mallin kyvyksi tunnistaa dataa, jota se ei ole aiemmin nähnyt ja useimmiten yleistämiskyvyn puute voi johtua liian vähäisestä datamäärästä opetusvaiheessa, mikä johtaa ylisovittumiseen. Koska tässä vaiheessa oli otettu jo käyttöön kaikki saatu aineisto, oli vaihtoehtona käyttää data-augmentaatiota.

Data-augmentaatiolla tarkoitetaan aineiston lisäämistä keinotekoisesti olemassa olevasta datasta satunnaisten muutosten, kuten kuvien kiertämisen, tarkkuuden muuntamisen, kääntämisen jne. avulla, jolloin saadaan alkuperäisen kuvan kaltaisia mutta silti erilaisia variaatioita [8]. Augmentaatiovariaatioita on olemassa erilaisia, mutta tämän työn osalta päädyttiin käyttämään tutkan kuviin sopivia metodeja eli ns. shiftausta, mikä tarkoittaa kuvaajan alkamisajankohdan siirtämistä eteenpäin ja loppuosan siirtämistä alkuun, sekä signaalin heikentämistä ja vahvistamista. Aineistoa tutkimalla on helppo havaita, että kävelynäytteitä on selvästi enemmän kuin muita luokkia (6.4). Niinpä päädyttiin lisäämään neljän pienimmän luokan, eli makuulle meneminen ja nouseminen sekä kaatuminen ja siitä ylösnouseminen, määrien lisäämistä data-augmentaation keinoin. Lisäksi aineiston opetuksessa käytettiin luokkien painokertoimia (*eng. class weights*), joilla pienempiä luokkia pystyttiin

painottamaan enemmän verrattuna isompiin luokkiin.

Luokat makuulle meneminen ja makuulta nouseminen olivat pienimmät luokat ja niistä oli alle 300 näytettä kummastakin opetusaineistossa. Näiden määriä haluttiin kaksinkertaistaa, joten kummastakin kategoriasta valittiin satunnaisesti 60 luokkaa edustavaa näytettä, joille tehtiin viisi eri variaatiota: siirto, signaalin heikennys, signaalin vahvistus, siirto ja heikennys sekä siirto ja vahvistus. Näin 60:llä kuvalla saatiin luotua 300 augmentoitua kuvaa molempiin luokkiin. Kuvassa 6.2 on esitettyinä esimerkkikuvat alkuperäisestä signaalista ja siitä tehdyt versiot (siirretty, heikennetty ja vahvistettu). Taulukkoon 6.3 on koottuna luokittain, millainen vaikutus data-augmentoinnilla on näytteiden määriin.



Kuva 6.2: Esimerkkikuva data-augmentaatiosta. Kuva a edustaa alkuperäistä kuvaa. Kuvassa b on siirretty 32 askelta eteenpäin kuvaajaa. Kuvassa c on heikennetty signaalia alkuperäisessä kuvassa ja kuvassa d on vahvistettu signaalia.

Luokan nimi	Tunnus	Opetusaineisto ilman data-augmentaatiota	Opetusaineisto data-augmentaatiolla
walk	0	1614	1614
sit down	1	704	704
stand up	2	701	701
fall	3	457	687
stand up from floor	4	457	687
get in bed	5	284	584
get out bed	6	276	576
yhteensä		4493	5533

Taulukko 6.3: Opetusaineiston kasvattaminen data-augmentaation keinoin.

6.3.2 Datan luokittelu ja jakautuminen

PARrad aineistossa liikkeitä on yhteensä 14, kuten edellisissä luvuissa on kuvattu. Liikkeet yhdistettiin kävelyn, istuutumisen ja ylösnousemisen osalta, kuten artikkelissa tehtiin, jolloin saatiin lopulliseen aineistoon 10 eri liikettä. Tällä luokkamäärällä aloitettiin syväoppimismallin kehittäminen, mutta koska oli vaikeuksia saada mallia sovitettua aineistoon, päädyttiin aineiston luokittelua pienentämään ja jätettiin pois kolme sängyllä tapahtuvaa liikettä (lie in bed, roll in bed ja sit in bed). Aineisto jaettiin opetus-, validointi-, evaluointi- ja testausosaan. Liikkeet luokiteltiin numeroin 0-6. Kaiken kaikkiaan työssä käytetyssä aineistossa oli yli 6000 näytettä, joista luokkien karsimisen jälkeen jäi jäljelle noin 5600 näytettä, jotka jakaantuvat eri luokkiin ja aineistoihin taulukon 6.4 mukaisesti.

Opetusaineistosta osa otetaan validointiaineistoksi, koska sillä lasketaan tappioarvo (*eng. loss value*) ja metriikoita opetuksen aikana ja se on tärkeää pitää erillään opetusdatasta, sillä validoinnilla seurataan sitä, onko se, mitä malli oppii, hyödyllistä [8]. Evaluointidataa puolestaan käytetään siinä vaiheessa, kun malli on muodostettu ja sen toimintaa testataan ennen varsinaista testausvaihetta.

Luokan nimi	Tunnus	Opetus	Evaluointi	Testaus	yhteensä
walk	0	1614	201	196	2011
sit down	1	704	88	85	877
stand up	2	701	87	84	872
fall	3	457	60	55	572
stand up from floor	4	457	58	54	569
get in bed	5	284	40	36	360
get out bed	6	276	39	33	348
yhteensä		4493	573	543	5609

Taulukko 6.4: Aineiston jakautuminen opetus-, evaluointi- ja testausdataan sekä luokkien mukaisesti yhden tutkan datasta.

6.3.3 Syväoppimismallin kehittäminen

Syväoppimismallin kehittäminen aloitettiin ottamalla pohjaksi artikkelissa [5] tehty 2-ulotteinen konvoluutioneuroverkko. Tuota mallia lähdettiin muokkaamaan siten, että opeteltavien parametrien määrää saadaan pienemmäksi, koska tarkoituksena oli luoda pienikokoinen malli, joka sopisi langattoman sensoriverkkoon. Näin päätettiin testaamaan malleja, joissa pohjana on 1-ulotteinen konvoluutioverkko.

Taulukossa 6.5 on esitettyä ensimmäisen iteraation hyperparametreja, joita käytettiin etsittäessä optimaalisia parametreja eli kun tehtiin ns. *gridsearch*. Hyperparametreista on kerrottu tarkemmin luvussa 4.2.2 ja taulukossa 4.1. Eri hyperparametrien variaatioita muodostui yhteensä 288 tässä ensimmäisessä vaiheessa. Seuraavalle kierrokselle valittiin parhaimpien mallien mukaan osa hyperparametreista. Tätä jatkettiin kunnes mallien virheettömyys (*eng. accuracy*) ja oppimiskäyrät eivät enää parantuneet ja jäljelle jäi paras malli. Koska tutkia oli kaksi ja siten aineistojakin kaksi, tehtiin kummallakin tutkan datalla omat mallit. Jokaisen kehitettävän mallin kohdalla aloitettiin aina ensimmäisen vaiheen 288 konfiguraation ajamisella, jonka jälkeen hyperparametrien määrää vähennettiin saatujen tulosten perusteella.

Datan skaalaukseen otettiin mukaan testeihin MinMax Scaler ja Standard Scaler. Aineiston skaalauksella tarkoitetaan sitä, että aineiston arvoja saatetaan lähemmäksi toisiaan. Varsinkin jos arvot poikkeavat hyvin paljon toisistaan, malleja on vaikea saada sovittumaan dataan ilman skaalausta. Kaksi tyypillistä keinoa on käyttää minimi-maksimi skaalausta (*eng. min max scaling*) tai standardointia (*eng. standar-*

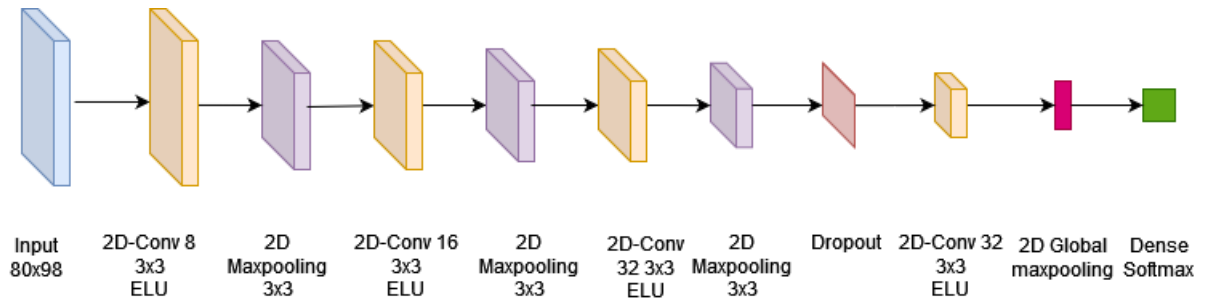
Parametri	Arvo
Filtterit	[4, 8, 12, 24], [6, 12, 24, 24], [8, 16, 32, 32]
Aktivaatiofunktio	ELU, ReLU
Alustusmekanismi	glorot_uniform he_uniform
Dropout	0.1, 0.2, 0.3, 0.4
Testikoko	0.15
Oppimisnopeus	0.001, 0.0005, 0.0001
Epochs	100
mini-batch koko	12
Skaalajat	minmax, std
Luokkien lukumäärä	7

Taulukko 6.5: Ensimmäisen vaiheen hyperparametrivaihtoehdot, joilla syväoppimismallia lähdettiin etsimään.

dization). Minimi-maksimi skaalauksessa aineiston arvot asetetaan tyypillisesti välille 0 ja 1 ja standardisaatiossa taas standardisoitujen arvojen keskiarvoksi saadaan aina 0 ja keskihajonnaksi 1 [15]. Jokaisen mallin muodostuksen yhteydessä muodostettiin myös mallin skaalaja, joka tallennettiin .pkl-muotoiseen tiedostoon, jotta sitä pystyttiin hyödyntämään mallin arvioinnissa ja testauksessa myöhemmin. Varsinainen malli tallennettiin .hdf5-muotoon jatkossa tapahtuvia testauksia varten.

Ensimmäisenä lähdettiin kehittämään konvoluutioneuroverkkoa, joka käyttää 1-ulotteisia konvoluutiokerroksia eli conv1D-kerroksia sekä yksiulotteisia maksimialinäytteistyskerroksia eli maxpooling1D-kerroksia. Tällä conv1D-mallilla ei kuitenkaan saavutettu haluttua lopputulosta, koska sen mallinuskkyky ei ollut riittävä. Siitä saatuja oppeja lähdettiin työstämään 2-ulotteista konvoluutioneuroverkkoa varten. Tämä 2-ulotteinen neuroverkko muodostui kolmesta konvoluutiokerroksen ja maksimialinäytteistyskerroksen muodostamasta blokista, jonka jälkeen tuli dropout-kerros ja sen jälkeen vielä yksi konvoluutiokerros ja globaali maksimiali-

näytteistyskerros. Lopullinen luokittelu tapahtui Dense-kerroksen Softmax-funktiota käyttäen. Kuvassa 6.3 on esitettyä kaavakuva 2D-konvoluutioneuroverkosta, jossa on käytetty 3x3 suodinkokoa ja filttereinä kerroksilla 8, 16, 32 ja 32 sekä alinäytteistystä 3x3 koolla.

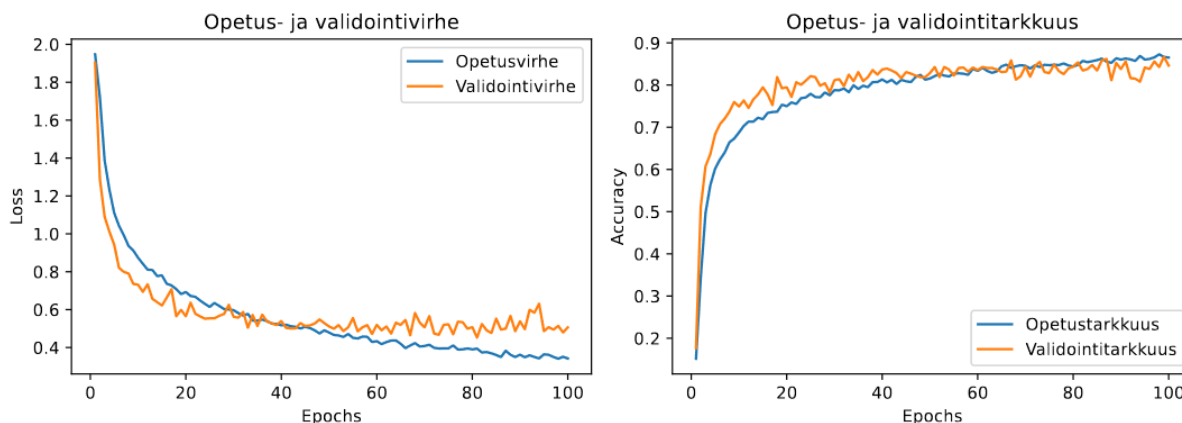


Kuva 6.3: Kaavakuva tehdystä 2D-konvoluutioneuroverkosta.

Jokaisen iteraatiokerran jälkeen arvioitiin tuloksia tarkastelemalla mallien oppimiskäyriä sekä virheettömyys (accuracy) arvoja ja valittiin uudet parametrit seuraavalle kierrokselle parhaimpien mallien perusteella. Viimeisessä vaiheessa parasta mallia testattiin vielä ajaa 5 kertaa 150 epochin verran, koska joka kerta, kun mallia ajetaan, opetus- ja validointidata jakautuvat hieman eri tavalla ja tällä tavoin selvitettiin, saadaanko vielä parempi malli aikaiseksi.

Kuvassa 6.4 voidaan nähdä yhden opetetun mallin oppimiskäyrät opetus- ja validointiaineistoilla. Mallin opetus tehtiin 100 epochin ajan. Opetus- ja validointivirheikäyrät laskevat jyrkästi aluksi, kunnes noin 50 epochin kohdalla saavutetaan piste, jossa käyrät erkanevat toisistaan. Tässä pisteessä alkaa ylisovittuminen, joka ei kuitenkaan tällä mallilla ole kovinkaan reilua. Lisäksi toisesta kuvaajasta voidaan nähdä, että opetus- ja validointitarkkuus (accuracy) seuraavat toisiaan ja tarkkuus paranee vielä epochien määrän kasvaessa 100:aan. Nämä oppimiskäyrät toimivat eri mallien arvioimisessa ja seuraavien iteraatioiden suunnittelussa, koska hyperparametrien arvoja vaihdellessa, voidaan seurata oppimiskäyrien käyttäytymistä. Jos esimerkiksi oppimisnopeutta hidastettaisiin ja opetus- ja validointivirheet alkaisivat erota toisistaan suuresti eli tapahtuisi reilusti ylisovittumista, silloin voitaisiin vetää se johtopäätös, että oppimisnopeutta ei tule hidastaa, ja se sulki tietyt hyperparametriarvot pois.

cfg nro. 34: >>> 83.24608



Kuva 6.4: Yhden työssä opetetun mallin oppimiskäyrät.

6.3.4 Työssä käytetyt ohjelmistot ja laitteisto

Työn ohjelmakoodi tehtiin käyttäen Python-ohjelmointikieltä, josta oli käytössä versio 3.9.12. Pythonin ohjelmakirjastoista oli käytössä esimerkiksi taulukoiden muodostamiseen ja laskentaan NumPy (versio 1.23.5), kuvaajien piirtämiseen Matplotlib (versio 3.6.2) ja datan käsittelyyn ja laskentaan pandas (versio 1.5.2). API-rajapintana käytettiin Kerasia ³, joka toimii tekoäly alustana toimivan TensorFlow'n päällä. Lisäksi data-analyysia varten käytössä oli scikit-learn ohjelmakirjastoja esimerkiksi aineiston jakamiseen sekä oppimismetriikoiden ja sekaannusmatriisien tulostukseen. Ohjelmakoodit mallien testauksessa ja optimaalisen mallin löytämisessä suoritettiin käyttäen Applen MacBook Pro kannettavaa tietokonetta, jossa on M1 Pro -prosessori ja 16-ytiminen Applen oma näytönohjain.

³<https://keras.io/>

7 Tutkimuksen tulokset ja analyysi

Edellä olevassa luvussa kuvattiin syväoppimismallien implementointi ja optimointi. Taulukossa 7.1 on koottuna valittujen konvoluutioneuroverkkojen arkkitehtuurit. Malli A on opetettu ja hyperparametrit optimoitu käyttäen tutkan TI-xWR14xx tuottamaa dataa. Tämä tutka oli alkuperäisen tutkimuksen mukaan kiinnitetty seinään, katonrajaan ja se käytti taajuutta 77 GHz. Tätä tutkaa kutsutaan jatkossa nimellä tutka 1. Malli B on puolestaan opetettu ja optimoitu käyttäen dataa tutkasta TI-xWR68xx, joka käytti taajuutta 60 GHz ja oli sijoitettuna silmien korkeudelle. Tämä tutka on jatkossa nimellä tutka 2. Yhteenvedona käytettävistä tutkien nimistä:

- Tutka 1: TI-xWR14xx, taajuus 77 GHz
- Tutka 2: TI-xWR68xx, taajuus 60 GHz

Tarkemmin tietoja tutkien ominaisuuksista on kuvattu taulukossa 6.1

	Malli A	Malli B	Malli C
opetusaineisto	Tutka 1	Tutka 2	Tutka 1 augmentoitu
Filtterit	[8, 16, 32, 32]	[8, 16, 32, 32]	[8, 16, 32, 32]
Kierrokset (<i>eng. epochs</i>)	100	100	100
Osajoukko (<i>eng. mini-batch</i>)	12	12	12
Aktivaatiofunktio	ReLU	ELU	ELU
Alustusfunktio	he_uniform	glorot_uniform	glorot_uniform
Dropout	0.3	0.3	0.5
Skaalaaja (<i>eng. scaler</i>)	MinMax	StandardScaler	MinMax
Oppimisnopeus (<i>learning rate</i>)	0.0005	0.0005	0.001
Testikoko	0.15	0.15	0.15
Parametreja	15 367	15 367	15 367

Taulukko 7.1: Valitut arkkitehtuurit konvoluutioneuroverkoille

Lopulliset testaukset toteutettiin kummallekin mallille käyttäen erillistä testausaineistoa, jossa oli ennalta näkemättömiä näytteitä henkilöiltä, joiden dataa ei oltu

käytetty opetukseen tai validointiin. Henkilöt 3, 12 ja 19 oli säästetty kokonaan testausta varten. Mallin A testaus tehtiin sekä tutkan 1 että tutkan 2 testiaineistolla. Vastaavasti tehtiin mallin B kohdalla. Eli tarkoituksena oli tehdä havaintoja myös siitä, vaikuttaako tuloksiin se, että malli opetetaan ensin yhden tutkan datalla ja sitten lopulliset testaukset tehdään myös toisen tutkan datalla. Lisäksi toteutettiin malli C, jonka opetukseen oli käytetty augmentoitua dataa TI-xWR14xx-tutkan datasta ja tälle tehtiin testit molempien tutkien testidatoilla.

7.1 Mallin A testitulokset

Malli A on opetettu käyttäen tutkan 1 aineistoa, ja sen evaluointi ja ensimmäinen testaus tehtiin tutkan 1 aineistoilla, jonka jälkeen tehtiin testi vielä tutkan 2 testiaineistolla. Mallin A tulokset sekaannusmatriisin muodossa evaluointidataa vastaan ovat nähtävissä taulukossa 7.2.

		Predicted label						
		walk	sit down	stand up	fall	stand up floor	get in bed	get out bed
True label	walk	167	2	17	1	3	3	8
	sit down	3	78	3	0	0	3	1
	stand up	7	6	73	0	0	0	1
	fall	6	11	0	42	1	0	0
	stand up floor	0	2	7	0	49	0	0
	get in bed	2	0	1	0	0	37	0
	get out bed	2	0	2	0	2	3	30

Taulukko 7.2: Mallin A sekaannusmatriisi tutkan 1 evaluointidatalla testattuna.

Taulukossa sarakkeiden nimet ovat ennustettuja luokkia ja rivien nimet oikeita luokkia. Keskelle taulukkoa muodostuu isompien lukujen muodostama "lävistäjä", jolla ovat oikein menneiden ennustusten lukumäärät eli oikeat positiiviset. Muut vaakariveillä olevat arvot ovat väärin menneiden ennustusten lukumääriä eli vää-

riä negatiivisia. Esimerkiksi walk eli kävelyluokan rivillä ensimmäisessä sarakkeessa on luku 167, mikä tarkoittaa oikein menneiden ennustusten lukumäärää ja loput luvut (2, 17, 1, 3, 3, 8) samalla rivillä ovat väärin menneiden ennustusten lukumääriä. Evaluointidata, kuten edellä olevassa luvussa kuvattiin, on erillinen aineisto, eikä siinä ole samoja näytteitä kuin opetusaineistossa. Evaluointia käytetään siinä vaiheessa, kun malli tai malleja muodostetaan ja tutkitaan sen suorituskkyä ennen varsinaista testausta ennen näkemättömällä aineistolla. Mallin A kohdalla on huomattavissa taulukosta 7.5, että evaluointidataa vastaan tehty testi tuottaa 83%:n virheettömyyden ja 84%:n osumatarkkuuden. Tarkemmin sekaannusmatriisista 7.2 on nähtävissä, että kävelyluokasta 17 näytettä on luokiteltu virheellisesti ylös nousemiseksi ja 8 näytettä sängystä nousemiseksi. Samoin 11 kaatumisnäytettä on ennustettu virheellisesti istuutumiseksi ja ylös nousemista on ennustettu kävelyksi 7 tapauksessa.

		Predicted label						
		walk	sit down	stand up	fall	stand up floor	get in bed	get out bed
True label	walk	166	3	20	1	1	1	4
	sit down	4	71	0	8	0	2	0
	stand up	0	6	71	2	3	0	2
	fall	5	3	0	43	1	3	0
	stand up floor	2	0	1	2	49	0	0
	get in bed	5	3	0	0	0	25	3
	get out bed	0	0	0	0	1	0	32

Taulukko 7.3: Mallin A sekaannusmatriisi tutkan tutkan 1 testidatalla testattuna.

Ennen näkemättömällä aineistolla tehty testaus toteutettiin ensin käyttäen tutkan 1 testiaineistoa, mistä tulokset on nähtävissä sekaannusmatriisissa 7.3. Eniten väärinmenneitä ennustuksia on edelleen kävelyluokassa. Toisaalta tämä on ymmärrettävää, sillä tämä luokka oli isoin, jolloin myös virheitäkin sattuu eniten. Toisaalta yksi merkittävä seikka on se, että kävely havainto sekoitetaan jopa 20 näytteen osal-

ta ylös nousemiseen. Myös ylösnousua lattialta ja sängyltä nousemista on ehdotettu kävelynäytteen luokaksi joissakin tapauksissa. Nämä liikkeet ovat siis sekoittuneet toisiinsa, eikä malli ole kaikissa tapauksissa oppinut tunnistamaan oikein piirteitä. Toinen sekaantumista aiheuttanut luokka oli istuutuminen, joka 8 tapauksessa luokiteltiin kaatumiseksi. Kolmantena ylös nouseminen tulkittiin joissain tapauksissa istuutumiseksi. Muissa luokissa virheitä tuli vain muutamia.

		Predicted label						
		walk	sit down	stand up	fall	stand up floor	get in bed	get out bed
True label	walk	138	17	13	1	9	1	17
	sit down	6	72	5	0	0	2	0
	stand up	5	30	41	1	1	0	6
	fall	5	42	0	3	0	5	0
	stand up floor	9	29	11	2	2	0	1
	get in bed	11	20	0	0	1	4	0
	get out bed	11	5	3	1	5	0	38

Taulukko 7.4: Mallin A sekaannusmatriisi tutkan 2 testidatalla suoritettuna.

Mallin A sekaannusmatriisi, kun testaus tehtiin tutkan 2 testausaineistoa vastaan, on nähtävillä taulukossa 7.4. Tässä on nähtävissä selvä ero aiempiin testeihin. Kaksi ensimmäistä luokkaa (kävely ja istuutuminen) ennustetaan oikein vielä kohdalaisen hyvin, mutta loput luokat taas eivät. Sekaannusmatriisissa ei ole enää nähtävissä selkeää "lävistäjää", jolle oikein menneet ennustukset asettuisivat. Kaatuminen, lattialta ylös nouseminen ja makuulle meneminen on tulkittu pääosin väärin, sillä kaikissa näissä on ennustettu luokaksi eniten istuutumista. Myös ylös nouseminen on ennustettu lähes yhtä usein istuutumiseksi kuin ylös nousemiseksi. Vaikuttaisi siis siltä, että tutkan 1 datalla opetettu malli ei yleisty kovin hyvin tutkan 2 aineistoon.

Mittari	Evaluointidata	Testidata tutka 1	Testidata tutka 2
Virheettömyys (<i>eng. accuracy</i>)	0.83072	0.84162	0.49355
Tarkkuus (<i>eng. precision</i>)	0.84153	0.84512	0.49489
Sensitiivisyys (<i>eng. recall</i>)	0.83072	0.84162	0.49355
F1-mitta (<i>eng. F1-score</i>)	0.83170	0.84176	0.45860

Taulukko 7.5: Mallin A suorituskyky mittareita eri testausaineistolla tehtynä.

Taulukosta 7.5 nähdään yhteenvedona eri testien tulokset mallilla A. Evaluointidatalla tehdyt testit antoivat kaikilla mittareilla yli 80%:n todennäköisyyksiä. Lisäksi ennen näkemättömällä testausaineistolla tutkasta 1 tehty testi oli tuloksiltaan jopa hiukan parempi, mitä evaluointi antoi odottaa. Sen sijaan tutkalla 2 tehty testi ei tuottanut toivottua tulosta siitä, että malli osaisi tunnistaa myös toisen tutkan dataa yhtä tarkasti.

7.2 Mallin B tulokset

Malli B opetettiin käyttäen tutkan 2 opetusaineistoa ja testattiin ensin käyttäen tutkan 2 evaluointiaineistoa ja testausaineistoa, jonka jälkeen viimeisenä testattiin vielä tutkan 1 testiaineistolla sen toimivuutta. Tulokset evaluointidatalla tehdystä testistä ovat nähtävissä taulukossa 7.7. Tämä testi antaa samansuuntaisia tuloksia kuin mallin A testit evaluointi- ja testiaineistolla. Kävelyhavainnot sekoittuvat edelleen ylös nousemiseen 16 tapauksessa, mikä ei ole yhtä paljon kuin mallilla A. Sen sijaan kävely ennustetaan myös istuutumisiksi 14 tapauksessa ja muutamissa tapauksissa myös muihin luokkiin. Ylös nouseminen on ennustettu 7 tapauksessa olevan kävelyä ja 6 tapauksessa istuutumista. Muissa luokissa väärin menneitä ennustuksia tulee joitain, mutta isompia virheitä ei tule esiin.

		Predicted label						
		walk	sit down	stand up	fall	stand up floor	get in bed	get out bed
True label	walk	159	14	16	2	4	4	2
	sit down	4	73	4	3	0	4	0
	stand up	7	6	71	0	2	0	1
	fall	5	3	0	50	2	0	0
	stand up floor	3	0	6	2	45	0	2
	get in bed	2	0	1	1	1	34	1
	get out bed	4	1	3	0	0	0	31

Taulukko 7.6: Mallin B sekaannusmatriisi tutkan 2 evaluointidatalla testattuna.

		Predicted label						
		walk	sit down	stand up	fall	stand up floor	get in bed	get out bed
True label	walk	148	5	29	3	2	5	4
	sit down	9	72	0	3	0	1	0
	stand up	5	10	61	2	3	2	1
	fall	6	3	0	44	0	1	1
	stand up floor	0	2	4	3	38	0	7
	get in bed	8	1	1	0	0	25	1
	get out bed	3	1	0	0	0	1	28

Taulukko 7.7: Mallin B sekaannusmatriisi tutkan 2 testidatalla testattuna.

Taulukossa 7.7 on esitettyä tulokset tutkan 2 testiaineistolla. Tätä aineistoa malli

ei ole siis aiemmin nähnyt. Verrattuna evaluointidatalla tehtyyn testiin luokkien ennustuksissa on jokseenkin enemmän hajontaa. Kävely sekoittuu nyt enemmän ylös nousemiseen, jopa 29 tapauksessa. Lisäksi istuutuminen sekoitetaan 9 tapauksessa kävelyyn, ylös nouseminen 10 tapauksessa istuutumiseen, kaatuminen 6 tapauksessa kävelyyn ja makuulle meneminen 8 tapauksessa kävelyyn. Linja on kuitenkin pääosin samansuuntainen kuin evaluointiaineistolla tehty testi.

Mallin B testaus tutkan 1 testausaineistolla antoi hyvin vaihtelevia tuloksia, joten sen sekaannusmatriisi jätetään tässä näyttämättä. Taulukosta 7.8 voidaan kuitenkin nähdä vaatimattomat tulokset suorituskykymittareissa. Kaiken kaikkiaan mallin B ennustuskyky ei parantunut siirryttäessä evaluointidatasta ennen näkemättömään testiaineistoon eikä se vakuuttanut suorituskykymittareilla. Lisäksi ennustuskyky toisen tutkan (tutka 1) testiaineistoa vastaan oli olematon.

Mittari	Evaluointidata	Testidata tutka 2	Testidata tutka 1
Virheettömyys (<i>eng. accuracy</i>)	0.80803	0.76611	0.11786
Tarkkuus (<i>eng. precision</i>)	0.81359	0.77446	0.48172
Sensitiivisyys (<i>eng. recall</i>)	0.80803	0.76611	0.11786
F1-mitta (<i>eng. F1-score</i>)	0.80912	0.76730	0.06991

Taulukko 7.8: Mallin B suorituskykymittareita eri testausaineistolla tehtynä.

7.3 Mallin C tulokset

Malli C opetettiin tutkan 1 augmentoidulla datalla. Tämä tarkoitti sitä aineistoa, jonka määrää oli lisätty keinotekoisesti pienempien luokkien osalta. Testit suoritettiin ensin tutkan 1 evaluointi- ja testausaineistolla ja sitten tutkan 2 testiaineistolla. Sekaannusmatriisi, jossa on kuvattuna evaluointidataa vastaan tehdyn testin tulokset, on nähtävissä taulukossa 7.9. Tulokset ovat samansuuntaisia kuin malleilla A ja B. Kävelyluokka aiheuttaa edelleen eniten vääriä tuloksia, joista eniten virheellisesti tulkitaan kävely ylösnousemiseksi 16 tapauksessa. Myös istuutumista ja kaatumista on ehdotettu kävelyn luokaksi joissain tapauksissa. Muiden luokkien ennustuksissa

ei nouse esiin yhtä suuria eroja.

		Predicted label						
		walk	sit down	stand up	fall	stand up floor	get in bed	get out bed
True label	walk	164	6	16	6	2	4	3
	sit down	4	75	3	1	0	4	1
	stand up	4	5	70	2	1	1	4
	fall	4	4	0	51	1	0	0
	stand up floor	0	0	6	0	46	1	5
	get in bed	0	2	0	0	0	38	0
	get out bed	4	1	0	0	0	1	33

Taulukko 7.9: Mallin C sekaannusmatriisi tutkan 1 evaluointidatalla testattuna.

		Predicted label						
		walk	sit down	stand up	fall	stand up floor	get in bed	get out bed
True label	walk	168	2	19	0	2	2	3
	sit down	4	76	0	4	0	1	0
	stand up	3	7	71	0	2	0	1
	fall	2	1	0	52	0	0	0
	stand up floor	1	1	8	1	43	0	0
	get in bed	2	2	0	0	0	31	0
	get out bed	0	0	1	0	0	0	32

Taulukko 7.10: Mallin C sekaannusmatriisi tutkan 1 testidatalla testattuna.

Malli C testattiin myös tutkan 1 testiaineistolla ja sen tulokset ovat nähtävissä sekaannusmatriisissa 7.10. Tässäkin on nähtävissä samantapaisia tuloksia kuin edellä. Kävelyä luultiin edelleen ylösnousemiseksi, nyt 19 tapauksessa, mutta verrattuna evaluointidataan, istuutumiseen ja kaatumiseen se ei enää merkittävästi sekoittunut. Muissa luokissa tulokset osuvat oikeaan muutamia kertoja lukuun ottamatta.

		Predicted label						
		walk	sit down	stand up	fall	stand up floor	get in bed	get out bed
True label	walk	158	12	8	4	9	2	3
	sit down	6	73	3	1	2	0	0
	stand up	12	16	50	2	4	0	0
	fall	1	15	0	39	0	0	0
	stand up floor	3	0	6	3	41	0	1
	get in bed	6	12	1	4	5	8	0
	get out bed	4	0	5	1	9	0	14

Taulukko 7.11: Mallin C sekaannusmatriisi tutkan 2 testidatalla testattuna.

Malli C testattiin lopuksi vielä tutkan 2 testiaineistolla. Tämä testi antoi lupaavampia tuloksia kuin mallien A ja B testit toisen tutkan aineistolla. Sekaannusmatriisista 7.11 nähdään, että luokkien ennustukset pysyvät paremmin kasassa ja vain sänkyyn menemistä on ennustettu useammin väärin kuin oikein. Hajontaa löytyy kyllä luokkien sisällä, mutta suunta on parempi kuin edellisten mallien vastaavilla ennustuksilla. Lisäksi kävelyä ei ennusteta enää yhtä usein ylösnousemiseksi, vaan hiukan enemmän istuutumiseksi.

Yhteenvetona taulukkoon 7.12 on koottuna mallin C suorituskyvyn mittareita eri testiaineistoilla. Mallin C kyky yleistyä ennalta näkemättömään testidataan tutkasta 1 on jopa parempi kuin evaluointidatalla tehty testaus antaisi odottaa. Lisäksi tutkan 2 testiaineistolla tehty testi antoi lupaavia tuloksia siitä, että malli pystyy tunnistamaan oikein myös toisen tutkan dataa.

Mittari	Evaluointidata	Testidata tutka 1	Testidata tutka 2
Virheettömyys (<i>eng. accuracy</i>)	0.83246	0.87109	0.70534
Tarkkuus (<i>eng. precision</i>)	0.84043	0.87791	0.72710
Sensitiivisyys (<i>eng. recall</i>)	0.83246	0.87109	0.70534
F1-mitta (<i>eng. F1-score</i>)	0.83375	0.87236	0.69600

Taulukko 7.12: Mallin C suorituskyky mittareita eri testausaineistolla tehtynä.

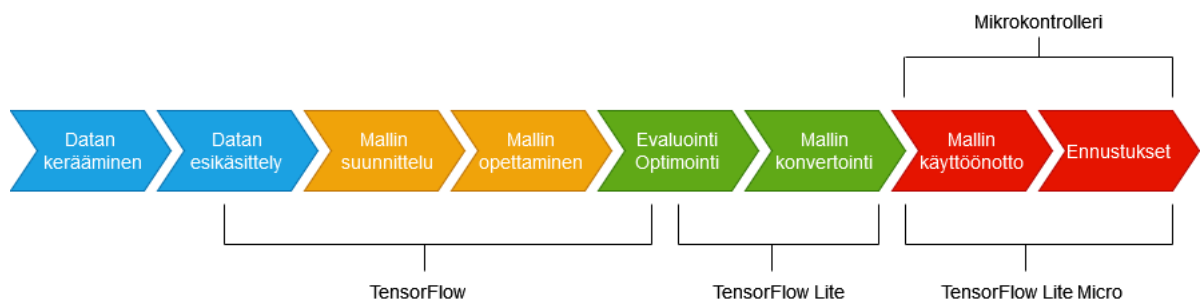
7.4 Mallin siirtäminen langattomaan sensoriverkkoon

Luvussa 2.1 käsiteltiin lyhyesti langattomia sensoriverkkoja ja niiden rajoituksia. Langattomat sensoriverkot asettavat omat haasteensa verkon toiminnalle, jotka tulisi ottaa huomioon verkkoa suunniteltaessa [10]. Useimmiten suurin haaste on se, että sensorin tulisi pärjätä rajoitetulla energiankäytöllä, sillä sensorit toimivat usein pattereilla tai ladattavilla akuilla. Yleensä sensorit toimivat siten, että ne keräävät tietoa ympäristöstä ja kerätty data siirretään muualle analysointia varten. Sensorien keräämät datamäärät saattavat kuitenkin olla hyvin suuria ja sen lähettäminen verkon yli muualle käsiteltäväksi vaatisi suuren kaistanleveyden ja siten aiheuttaisi verkon kuormittumista [13]. Lähettäminen ei ole siis kannattava ratkaisu tilanteissa, joissa tarvitaan isojen datamäärien siirtoa. Lisäksi usein sensoreilla kerätään arkaluontaista tietoa, ja sen lähettäminen altistaa myös ongelmatilanteille [10]. Siksi viime aikoina on siirretty huomiota datan analysoinnin ja laskennan siirtämiseksi suoraan sensorin mikrokontrollerissa. Mikrokontrolleri (*eng. microcontroller unit, MCU*) vastaa sensorin datan keräämisestä, lähetyksestä ja vastaanottamisesta [13]. TinyML on uusi konsepti, jolla saadaan optimoituja koneoppimisen malleja toimimaan mikrokontrollereilla, jolloin voidaan toteuttaa reaaliaikaista analysointia datalle suoraan tietoa keräävässä sensorissa ja samalla säästää kustannuksissa ja välttää arkaluontoisen tiedon lähettämistä [4].

Toisaalta on kuitenkin muistettava se, että mikrokontrollerin muisti on pieni, ja siihen tulisikin mahtua syväoppimismallin kaikki painokertoimet ja muut opitut tiedot [4]. Tämä tarkoittaa sitä, että TinyML-malli on optimoitava käyttämään ra-

joitettua muistikapasiteettia. Lisäksi mallin opetus vaatii paljon tehoa, energiaa ja muistia, jolloin sitä ei voida toteuttaa sensorissa, vaan mallin opetus ja optimointi on tehtävä muualla ennen kuin malli tuodaan sensorin käytettäväksi [13].

TinyML:n eduiksi artikkeli [13] mainitsee seuraavat asiat. Ensinnäkin se mahdollistaa laskennan viemisen suoraan sensoriin ja siten suurten datamäärien hyödyntämisen laitteessa sen sijaan, että kaikki lähetettäisiin muualle käsittelyyn. Toiseksi se parantaa tietoturvallisuutta ja yksityisyydensuojaa, sillä suuria määriä arkaluontoista dataa ei tarvitse lähettää laitteista verkon yli pilveen. Lisäksi voidaan huomata latenssin vähenemistä, koska dataa ei tarvitse lähettää ja prosessoida muualla, pois-tuu monta vaihetta prosessiketjussa eikä siten tule myös viivettä. Toisaalta TinyML tuo myös energiatehokkuuteen parannusta, koska jos dataa ei tarvitse lähettää suuria määriä, se kuluttaa myös vähemmän energiaa. Lisäksi, koska dataa kuljetetaan vähemmän, myös kaistanleveyden käyttö vähenee ja siten käyttö on edullisempaa.



Kuva 7.1: Syväoppimismallin kehittäminen valmiiksi malliksi mikrokontrolleriin [38, s.5] mukailten.

Tässä työssä tavoitteena oli tehdä syväoppimismalli, joka saataisiin sellaiseen kokoon, että se mahtuisi mikrokontrolleriin ja siten se olisi käytettävissä suoraan langattomaan sensoriverkkoon liitetystä tutkasensorissa. Kuvassa 7.1 on esitettyä artikkelin [38] mukainen työnkulku syväoppimismallin saattamiseksi mikrokontrollerin käyttöön, johon on lisätty tässä työssä käytyt tekoälyalustan TensorFlow:n muodot. Kun tarvittava aineisto mallia varten on kerätty, se esikäsitellään tarvittaessa, jonka jälkeen alkaa mallin suunnittelu ja opettaminen. Nämä kaikki tehdään TensorFlow-alustalla, samoin kuin osa suorituskyvyn optimoinnista. Tämän jälkeen lisää optimointia sekä mallin konvertointi tapahtuvat käyttäen TensorFlow Litea. TensorFlow Lite tarjoaa mahdollisuuden ajaa TensorFlowlla tehdyt mallit matkapuhelimissa, sulautetuissa systeemeissä ja reunalaskentaa hyödyntävissä laitteissa [52]. Konvertoinnilla (*eng. conversion*) tarkoitetaan mallin muuntamista sellaiseen

muotoon, jossa se mahtuu mikrokontrolleriin. Esimerkiksi TensorFlow tapauksessa otetaan TensorFlow-malli ja muunnetaan se TensorFlow Lite -malliksi (.tflite tiedostoksi) [50].

Yksi tapa toteuttaa konversio on tehdä mallille opetuksen jälkeen kvantisointia (*eng. post-training quantization*) [51]. Tällä tavalla mallin kokoa voidaan pienentää ja myös parantaa laitteiston latenssia vain vähäisellä mallin tarkkuuden heikkeneemisellä. Kvantisointia voidaan tehdä dynaamisesti, jolloin mallin painokertoimet muunnetaan liukuluvuista kokonaisluvuiksi. Mikäli on saatavilla edustava testiaineisto, voidaan kvantisointia jatkaa edelleen, sillä tällöin voidaan tehdä kokonaisuudessaan muunnos kokonaislukuun (*eng. full integer quantization*). Testiaineistoa tarvitaan mallin sisään- ja ulostulon sekä aktivaatioiden kalibroimiseen. Edelleen, mikäli mallia halutaan käyttää 8-bittisillä mikrokontrollereilla, jotka käyttävät vain kokonaislukuja, on sekin mahdollista toteuttaa pakottamalla koko malli käyttämään kokonaislukuja.

Kun alkuperäinen TensorFlow malli on saatu TensorFlow Lite -muotoon (.tflite-tiedostoksi), se voidaan edelleen muuntaa C-kielen taulukoksi [49]. Tämä tehdään sen takia, koska useat mikrokontrollerit, joissa mallia halutaan käyttää, eivät sisällä käyttöjärjestelmää tai tiedostojärjestelmää, mutta taulukkomuotoista dataa ja alemman tason kieltä ne pystyvät käyttämään.

Edellä kuvattiin kolme konvoluutioneuroverkkomallia, joista parhaimmaksi osoitettiin malli C. Tämä malli opetettiin käyttäen aineistoa, jossa aineiston määrää oli keinotekoisesti kasvatettu data-augmentaation keinoin. Malli C valittiin tähän viimeiseen vaiheeseen, jossa mallista muodostettiin .tflite-muotoinen malli mikrokontrolleria varten. Tarkoituksena oli tehdä täysin kvantisoitu malli, jossa on siis vain kokonaislukuja käytössä (*eng. integer only*). Ensimmäisessä vaiheessa aineistosta valittiin 339 näytteen muodostama edustava datasetti (*eng. representative dataset*), jota käytettiin mallin kvantisoinnin kalibroinnissa. Tämän jälkeen suoritettiin varsinaisen kvantisointi, jonka jälkeen .hdf5-muotoinen alkuperäinen TensorFlow malli oli muutettu .tflite-muotoon. Alkuperäinen malli .hdf5-tiedostona on 235 kilotavua kooltaan ja kvantisoinnin jälkeen sen koko saatiin 24 kilotavuun eli kymmenesosaan alkuperäisestä. Tämä malli muunnettiin vielä C-kieliseksi taulukoksi ja sen koko oli samaa luokkaa kuin kvantisoidun mallin. Mallit, niiden koot ja tarkkuudet on tiivistettynä taulukossa 7.13. Lopuksi .tflite-muotoiselle mallille tehtiin vielä testaus tutkan 1 testiaineistolla. Tästä tuloksena saatiin vastaava ja jopa hieman parempi tulos kuin alkuperäisellä mallilla, eli 87.477% virheettömyys (*eng. accuracy*) kun alkupe-

räisellä mallilla se oli 87.109%. C-kielistä taulukkoa ei testattu käytännössä, sillä tähän olisi tarvittu varsinainen sensori ja mikrokontrolleri, jolla mallia olisi testattu oikeissa mittaustilanteissa. Tämä ei kuulunut enää tämän työn laajuuteen, mutta se on toteutettavissa jatkossa, mikäli mallin kehitystä ja käyttöönottoa halutaan jatkaa.

Malli	Koko	Virheettömyys (<i>eng. accuracy</i>)
malli C (.hdf5-tiedosto)	235 kt	87.109%
tflite_quant_model.tflite	24 kt	87.477%
C-kielinen taulukko	24.4 kt	-

Taulukko 7.13: Mallin C muunnokset ja tarkkuudet.

7.5 Tulosten vertailua alkuperäiseen tutkimukseen

Tässä työssä käytetty aineisto oli osa PARrad-aineistoa ja Bhavanasi et al. [5] tutkimuksen aineistoa. Heidän tekemänsä tutkimus sisälsi enemmän aineistoa ja malli tehtiin tunnistamaan 10 eri liikettä. Tässä esitelty työ jouduttiin kuitenkin aineiston vähäisyyden takia rajaamaan niin, että luokkia otettiin vain 7. Näin ollen näiden kahden työn tulokset eivät ole täysin vertailtavissa. Kuitenkin joitain kohtia voidaan tässä vielä nostaa esiin.

Ensinnäkin töiden lähtökohdat olivat erilaiset. Bhavanasi et al. [5] toteuttivat kaksi konvoluutioneuroverkkomallia, joista toinen oli 2D-konvoluutiokerroksia ja toinen 3D-konvoluutiokerroksia käyttävä malli, joten mallien koot olivat isommat. Tässä pro gradu työssä toteutetuissa malleissa oli käytössä esimerkiksi voimakkaampi poolaus eli alinäytteistys sekä globaali maksimialinäyteistys, jota ei ollut ollenkaan alkuperäistutkimuksen malleissa. Näillä toimenpiteillä saatiin vähennettyä parametrien määrää ja siten mallin kokoa. Tämän työn tavoite oli tehdäkin parametreiltaan mahdollisimman pieni malli, joka olisi mahdollisuus saada langattoman sensoriverkon mikrokontrolleriin. Tätä näkökohtaa Bhavanasi et al. [5] eivät tuoneet esiin.

Toisaalta suoritusmittareilla tarkasteltuna ei päästy ihan samoihin lukemiin alkuperäisen tutkimuksen kanssa. Siinä parhaimmillaan etäisyys-Doppler syötettä käyttävä 3D-konvoluutioneuroverkko sai jopa 95%:n virheettömyyden [5]. Tähän tietysti vaikutti se, että aineistoa oli enemmän käytössä ja luokkiakin oli eri määrä. Mutta tässä pro gradu työssä oli lupaavaa se, että malli C onnistui kohtuullisesti ennus-

tamaan myös toisen tutkan dataa, jolla sitä ei oltu opetettu. Myös Bhavanasi et al. [5] tutkivat tätä ja he saivat tulokseksi MD-karttaa syötteenä käyttäen 71%:n virheettömyyden, kun malli oli opetettu 77 GHz:n tutkan datalla ja testattiin 60 GHz:n tutkan datalla. Tässä työssä käytettiin myös MD-karttaa ja mallin C virheettömyys vastaavilla tutkilla opetettuna ja testattuna saatiin likimain sama eli 70.534%.

7.6 Pohdintaa saaduista tuloksista ja jatkokehitysideoita

Edellä kuvattiin kolmen eri mallin tuloksia eri testiaineistoilla testattuna. Malli A ja C näyttävät tulosten valossa jopa parantavan ennustuskykyään kun siirrytään evaluointidatasta saman tutkan testidatalla tehtäviin testeihin. Lisäksi malli C antaa hyviä tuloksia myös, kun sitä testataan toisen tutkan testiaineistolla. Tämä antaa viitetä siitä, että mallilla C on kykyä yleistyä myös ennennäkemättömään dataan. Tällä on merkitystä esimerkiksi tilanteessa, jos tutkaa tai sen sijaintia muutetaan, jolloin se vaikuttaa myös mallin saamaan syötteeseen. Malli B sen sijaan ei vakuuttanut suorituskyvyllään. Vain evaluointidatalla tehty testi antoi yli 80%:n virheettömyyden. Ennennäkemättömän datan kanssa se suoriutui huonommin ja toisen tutkan dataa vastaan erittäin heikosti. Taulukkoon 7.14 on koottuna yhteenvetona kaikkien kolmen mallin tiedot sekä tulos testistä, joka tehtiin saman tutkan ennen näkemättömällä testiaineistolla. Malli C antaa selvästi parhaimman tuloksen yltäessään yli 87%:n virheettömyyteen. Lisäksi samasta mallista tehty optimoitu TinyML-malli mikrokontrollerille antaa myös lupaavia tuloksia mallin mahdollisuuksista toimia myös langattomassa sensoriverkossa.

Malli	Opetusdata tutkasta	Virheettömyys (<i>eng. accuracy</i>)
Malli A	TI-xWR14xx	0.84162
Malli B	TI-xWR68xx	0.76611
Malli C	TI-xWR14xx augmentoitu	0.87109

Taulukko 7.14: Yhteenveto tehtyjen mallien tiedoista ja tuloksista testistä, joka tehtiin saman tutkan datalla.

Empiirisen tutkimuksen haasteeksi osoittautui aineiston vaihtelevuus sekä luokkien epätasaisuus. Aineisto ja sen pilkkominen näytteiksi osoittautui hankalaksi, sillä alkuperäisen tutkimuksen mukaan tehty näytekoko toimi huonosti eikä siihen kokoon mahtunut riittävästi tietoa liikkeestä mukaan. Toisaalta eri liikkeet vaikutti-

vat toisinaan olevan hyvinkin samankaltaisia kuvissa, mikä herätti epäilyksen, onko aineiston kokoamisessa tai luokittelussa tapahtunut virhettä. Joka tapauksessa näihinkin ongelmiin olisi saattanut löytyä ratkaisu, jos aineisto olisi ollut suurempi ja sitä kautta mallillakin olisi ollut enemmän dataa, mistä oppia. Erityisesti kävelyliikkeen sekoittuminen ylösnousemiseen oli selvää, eikä se tuntunut läpi iteraatioiden ja erilaisten mallien kokeilujen kautta muuttuvan mihinkään. Tässä auttaisi varmaan myös aineiston parempi tuntemus sekä itse mittausympäristön, laitteiden ja olosuhteiden tuntemus. Tuntemus siitä, millaiselta esimerkiksi kävelyn tulisi näyttää tutkan havainnoimana, auttaisi myös tekemän johtopäätöksiä mallin toimivuudesta ja siitä, olisiko sen pitänyt tunnistaa jokin liike oikein. Aineistoa tutkittiin osittain tarkemmin sekoittuvien luokkien kävely ja ylös nousu osalta kuva kerrallaan sekä väärin menneissä tapauksissa että myös oikein menneissä tapauksissa. Vaikuttaa siltä, että kävely tuottaa jatkuvaa aaltoilevaa liikettä kuvaajaan, kun taas ylösnouseminen tuottaa äkillisen muutoksen signaaliin ja sitten taas toisaalta vaimeamman muutoksen liikkeen loputtua. Joissakin tapauksissa kävelynäytteet olivat hie-man samankaltaisia, katkonaisia signaaleja, jolloin virheellinen tulkintakin saattoi tapahtua. Koska aineisto ei ole tämän työn tekijän keräämää, on vaikea sanoa, onko tällaiset tapaukset olleet poikkeavia tai virheellisesti luokiteltuja alun perin. Mutta selvää näyttäisi olevan se, että kävelyhavainnot ovat olleet joiltain osin hyvin vaihtelevan näköisiä ja siksi aiheuttaneet mallissa virheellisiä tulkintoja luokasta.

Tässä tutkimuksessa tehtyjen mallien ja niiden suorituskykytestien perusteella voidaan sanoa, että datan augmentaatiolla eli keinotekoisella lisäämisellä saatiin parhaimmat tulokset sekä evaluointidatalla että ennennäkemättömällä datalla testattuna. Seuraavassa vaiheessa voisi kokeilla entisestään lisätä datan määrää, myös niissä luokissa, joissa sitä ei vielä tehty, lukuun ottamatta suurinta luokkaa eli kävelyä. Data-augmentaatio vaikuttaa siis olevan tehokas keino parantaa mallin toimivuutta ja ennustuskykyä. Esimerkiksi tutkimuksessa vuodelta 2022 Li et al. [30] käyttivät data-augmentaatiota aineiston lisäämiseksi, jolla he saivat aineiston 7-kertaiseksi. Tällä tavalla malli saatiin toimivammaksi ja sen suorituskyky parani verrattuna malliin, jossa ei oltu käytetty data-augmentaatiota.

Jatkossa vastaavissa liikkeentunnistussovelluksissa tulisi kiinnittää huomiota siihen, että aineistoa on tarpeeksi saatavilla mallin opetusta varten. Myös se, että data on arvioitu ja todettu olevan asianmukaista, auttaa varmasti mallin kehittämisessä ja toimivuudessa. Lisäksi usein tällainen liikkeentunnistus tapahtuu langattomissa sensoriverkoissa, mikä haastaa resursseja, niin energian kuin käytettävissä olevan

muistin ja laskentatehon kannalta, jolloin olisi tärkeää myös etsiä sellaisia syväoppimismalleja, jotka saataisiin mahtumaan pieneen tilaan eivätkä vaatisi esimerkiksi paljon muistia ja laskentatehoa.

Aineiston riittävän määrän lisäksi seuraavissa vaiheissa olisi hyvä miettiä myös tilanteita ja paikkoja, missä liikkeen tunnistusta tarvitaan. Esimerkiksi yksin asuvan henkilön kohdalla tilanne on yksinkertaisempi, mutta miten tunnistaminen toimii, kun tarkkailtavassa tilassa on useita henkilöitä ja miten liikkeet voidaan päätellä kuuluvan tietylle ihmiselle. Lisäksi virheellisiä tulkintoja saattavat aiheuttaa kotieläimet tai huonekalujen sijoittelu. Kaiken kaikkiaan liikkeen tunnistuksessa tutkan ja syväoppimisen keinoin on vielä paljon tutkittavaa mutta samalla myös paljon potentiaalia hyödyntää moniin eri tarkoituksiin.

8 Yhteenveto

Tässä työssä tutustuttiin aluksi langattomaan sensoriverkkoon ja erilaisiin sensoreihin liikkeen tunnistuksessa. Tämän jälkeen käsiteltiin tutkaa, sen ominaisuuksia sekä sen tuottamaa informaatiota. Tutkasta siirryttiin syväoppimiseen käsitteenä ja tutustuttiin tarkemmin konvoluutioneuroverkon toimintaan. Lisäksi käytiin läpi yleisiä periaatteita ja suuntaviivoja syväoppimismallin optimoinnista ja erilaisista metriikoista, joilla arvioidaan sen ennustuskykyä. Tämän jälkeen tehtiin katsausta siihen, miten tutkan ja syväoppimisen avulla on tehty tutkimusta liikkeiden tunnistamiseksi.

Työn empiirisessä osuudessa toteutettiin liikkeen tunnistusta varten kolme erilaista konvoluutioneuroverkkomallia, joiden opetukseen käytettiin PARrad-aineistoa [5]. Tavoitteena oli toteuttaa malli, joka kykenisi tunnistamaan 7 eri liikettä tutkan datasta ja joka olisi kooltaan sopiva käytettäväksi langattomassa sensoriverkossa. Parhaiten suoriutui malli, jonka aineistoa oli lisätty keinotekoisesti. Jatkossa aineistoa tulisi olla enemmän ja sen laatu tulisi varmistaa, jotta mallista voitaisiin saada entistä parempi ja tunnistuskykyisempi. Tämän lisäksi kuvattiin mallin muuntaminen pienemmäksi TinyML-malliksi ja C-kieliseksi taulukoksi, joita voidaan jatkossa käyttää suoraan tutkasensorin mikrokontrollerissa. Testeissä saatiin lupaavia tuloksia niin tavallisen mallin kuin pienemmän TinyML-mallin osalta, joissa tarkkuudeksi saatiin parhaimmillaan yli 87%:n virheettömyys.

Liikkeiden tunnistukselle on olemassa useita sovelluskohteita, kuten erilaiset valvontaan liittyvät, terveydentilan tarkkailuun ja kaatumisen tunnistukseen liittyvät sovellukset. Tulevaisuudessa tarvitaan lisää tutkimusta siitä, kuinka tunnistetaan ja erotellaan usean kohteen liikkeet toisistaan. Myös laadukkaan ja kattavan aineiston saaminen on olennaista toimivan mallin luomiseksi. Lisäksi syväoppimismallien kokoon sekä siihen, että ne on mahdollista saada mikrokontrollerien käyttöön tulisi kiinnittää jatkossa huomiota, jotta ne sopisivat todellisiin käyttötilanteisiin esimerkiksi langattomissa sensoriverkoissa.

Lähteet

- [1] ABDU, F. J., ZHANG, Y., FU, M., LI, Y., JA DENG, Z. Application of deep learning on millimeter-wave radar signals: A review. *Sensors* 21, 6 (2021), 1951.
- [2] ALIZADEH, M., SHAKER, G., DE ALMEIDA, J. C. M., MORITA, P. P., JA SAFAVI-NAEINI, S. Remote monitoring of human vital signs using mm-wave fmcw radar. *IEEE Access* 7 (2019), 54958–54968.
- [3] ALWAN, M., RAJENDRAN, P. J., KELL, S., MACK, D., DALAL, S., WOLFE, M., JA FELDER, R. A smart and passive floor-vibration based fall detector for elderly. *Julkaisusarjassa 2006 2nd International Conference on Information & Communication Technologies* (2006), vol. 1, IEEE, 1003–1007.
- [4] BANBURY, C., ZHOU, C., FEDOROV, I., MATAS, R., THAKKER, U., GOPE, D., JANAPA REDDI, V., MATTINA, M., JA WHATMOUGH, P. Micronets: Neural network architectures for deploying tinyml applications on commodity microcontrollers. *Proceedings of machine learning and systems* 3 (2021), 517–532.
- [5] BHAVANASI, G., WERTHEN-BRABANTS, L., DHAENE, T., JA COUCKUYT, I. Patient activity recognition using radar sensors and machine learning. *Neural Computing Applications* 34, 18 (09 2022), 16033–16048.
- [6] CHEN, H., DING, C., ZHANG, L., HONG, H., JA ZHU, X. Human activity recognition using temporal 3dcnn based on fmcw radar. *Julkaisusarjassa 2022 IEEE MTT-S International Microwave Biomedical Conference (IMBioC)* (2022), IEEE, 245–247.
- [7] CHEN, V., LI, F., HO, S.-S., JA WECHSLER, H. Micro-doppler effect in radar: phenomenon, model, and simulation study. *IEEE Transactions on Aerospace and Electronic Systems* 42, 1 (2006), 2–21.
- [8] CHOLLET, F. *Deep learning with Python*. Simon and Schuster, 2021.
- [9] DARGAN, S., KUMAR, M., AYYAGARI, M. R., JA KUMAR, G. A survey of deep learning and its applications: a new paradigm to machine learning. *Archives of Computational Methods in Engineering* 27, 4 (2020), 1071–1092.

- [10] DARGIE, W., JA POELLABAUER, C. *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons, 2010.
- [11] DING, W., GUO, X., JA WANG, G. Radar-based human activity recognition using hybrid neural network model with multidomain fusion. *IEEE Transactions on Aerospace and Electronic Systems* 57, 5 (2021), 2889–2898.
- [12] DONG, S., WANG, P., JA ABBAS, K. A survey on deep learning and its applications. *Computer Science Review* 40 (2021), 100379.
- [13] DUTTA, L., JA BHARALI, S. Tinyml meets iot: A comprehensive survey. *Internet of Things* 16 (2021), 100461.
- [14] FU, B., DAMER, N., KIRCHBUCHNER, F., JA KUIJPER, A. Sensing technology for human activity recognition: A comprehensive survey. *IEEE Access* 8 (2020), 83791–83820.
- [15] GÉRON, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. "O'Reilly Media, Inc.", 2022.
- [16] GHOSH, A., SANYAL, A., CHAKRABORTY, A., SHARMA, P. K., SAHA, M., NANDI, S., JA SAHA, S. On automatizing recognition of multiple human activities using ultrasonic sensor grid. *Julkaisusarjassa 2017 9th international conference on communication systems and networks (COMSNETS)* (2017), IEEE, 488–491.
- [17] GHOSH, A., SUFIAN, A., SULTANA, F., CHAKRABARTI, A., JA DE, D. Fundamental concepts of convolutional neural network. *Recent trends and advances in artificial intelligence and Internet of Things* (2020), 519–567.
- [18] GU, J., WANG, Z., KUEN, J., MA, L., SHAHROUDY, A., SHUAI, B., LIU, T., WANG, X., WANG, G., CAI, J., ET AL. Recent advances in convolutional neural networks. *Pattern recognition* 77 (2018), 354–377.
- [19] HAN, T., KANG, W., JA CHOI, G. Ir-uwB sensor based fall detection method using CNN algorithm. *Sensors* 20 (10 2020), 5948.
- [20] HERNANGÓMEZ, R., SANTRA, A., JA STAŃCZAK, S. Human activity classification with frequency modulated continuous wave radar using deep convolutional neural networks. *Julkaisusarjassa 2019 International Radar Conference (RADAR)* (2019), IEEE, 1–6.

- [21] IOVESCU, C., JA RAO, S. The fundamentals of millimeter wave sensors. *Texas Instruments* (2017), 1–8.
- [22] JACOBI, R., JA AGINSKIY, A. Choosing 60-ghz mmwave sensors over 24-ghz to enable smarter industrial applications. *Nov 2018, Texas Instruments*, <http://www.ti.com/lit/wp/spry328/spry328.pdf> (2018).
- [23] JOKANOVIC, B., JA AMIN, M. Fall detection using deep learning in range-doppler radars. *IEEE Transactions on Aerospace and Electronic Systems* 54, 1 (2018), 180–189.
- [24] KHURANA, R., AHUJA, K., YU, Z., MANKOFF, J., HARRISON, C., JA GOEL, M. Gymcam: Detecting, recognizing and tracking simultaneous exercises in unconstrained scenes. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (2018), 1–17.
- [25] KIM, H., JA JEONG, Y.-S. Sentiment classification using convolutional neural networks. *Applied Sciences* 9, 11 (2019), 2347.
- [26] KIM, Y., JA MOON, T. Human detection and activity classification based on micro-doppler signatures using deep convolutional neural networks. *IEEE geoscience and remote sensing letters* 13, 1 (2015), 8–12.
- [27] KUMAR, D. P., AMGOTH, T., JA ANNAVARAPU, C. S. R. Machine learning algorithms for wireless sensor networks: A survey. *Information Fusion* 49 (2019), 1–25.
- [28] LECUN, Y., BENGIO, Y., JA HINTON, G. Deep learning. *Nature* 521 (05 2015), 436–44.
- [29] LI, C., HE, Y., LI, X., JA JING, X. Bigru network for human activity recognition in high resolution range profile. *Julkaisusarjassa 2019 International Radar Conference (RADAR)* (2019), IEEE, 1–5.
- [30] LI, W., ZHANG, D., LI, Y., WU, Z., CHEN, J., ZHANG, D., HU, Y., SUN, Q., JA CHEN, Y. Real-time fall detection using mmwave radar. *Julkaisusarjassa ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2022), 16–20.
- [31] LI, X., HE, Y., JA JING, X. A survey of deep learning-based human activity recognition in radar. *Remote Sensing* 11, 9 (2019), 1068.

- [32] LI, Z., LIU, F., YANG, W., PENG, S., JA ZHOU, J. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems* (2021).
- [33] LIM, J.-H., JANG, H., JANG, J., JA PARK, S.-J. Daily activity recognition system for the elderly using pressure sensors. *Julkaisusarjassa 2008 30th annual international conference of the ieee engineering in medicine and biology society* (2008), IEEE, 5188–5191.
- [34] LIPTON, Z. C., ELKAN, C., JA NARAYANASWAMY, B. Thresholding classifiers to maximize f1 score. *arXiv preprint arXiv:1402.1892* (2014).
- [35] LIU, W., WANG, Z., LIU, X., ZENG, N., LIU, Y., JA ALSAADI, F. E. A survey of deep neural network architectures and their applications. *Neurocomputing* 234 (2017), 11–26.
- [36] MENZEL, W. Millimeter-wave radar for civil applications. *Julkaisusarjassa The 7th European Radar Conference* (2010), IEEE, 89–92.
- [37] RAMASUBRAMANIAN, K., JA RAMAIAH, K. Moving from legacy 24 ghz to state-of-the-art 77-ghz radar. *ATZelegtronik worldwide* 13, 3 (2018), 46–49.
- [38] REDDI, V. J., PLANCHER, B., KENNEDY, S., MORONEY, L., WARDEN, P., AGARWAL, A., BANBURY, C., BANZI, M., BENNETT, M., BROWN, B., ET AL. Widening access to applied machine learning with tinyml. *arXiv preprint arXiv:2106.04008* (2021).
- [39] RICHARDS, M., SCHEER, J., JA HOLM, W. *Principles of Modern Radar: Basic Principles*, vol. 1 of *Principles of Modern Radar*. SciTech Pub., 2010.
- [40] SHAO, Y., GUO, S., SUN, L., JA CHEN, W. Human motion classification based on range information with deep convolutional neural network. *Julkaisusarjassa 2017 4th International Conference on Information Science and Control Engineering (ICISCE)* (2017), IEEE, 1519–1523.
- [41] SHELKE, S., JA AKSANLI, B. Static and dynamic activity detection with ambient sensors in smart spaces. *Sensors* 19, 4 (2019), 804.
- [42] SHRESTHA, A., JA MAHMOOD, A. Review of deep learning algorithms and architectures. *IEEE Access* 7 (2019), 53040–53065.

- [43] SINGH, A., REHMAN, S. U., YONGCHAREON, S., JA CHONG, P. H. J. Sensor technologies for fall detection systems: A review. *IEEE Sensors Journal* 20, 13 (2020), 6889–6919.
- [44] SKOLNIK, M. *Introduction to radar systems: Third edition*. McGraw-Hill, 2001.
- [45] SKOLNIK, M. I. *Radar handbook*. McGraw-Hill Education, 2008.
- [46] SOKOLOVA, M., JAPKOWICZ, N., JA SZPAKOWICZ, S. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. *Julkaisusarjassa Australasian joint conference on artificial intelligence (2006)*, Springer, 1015–1021.
- [47] SOKOLOVA, M., JA LAPALME, G. A systematic analysis of performance measures for classification tasks. *Information processing & management* 45, 4 (2009), 427–437.
- [48] SUNDHOLM, M., CHENG, J., ZHOU, B., SETHI, A., JA LUKOWICZ, P. Smartmat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix. *Julkaisusarjassa Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing (2014)*, 373–382.
- [49] TENSORFLOW. Build and convert models. URL https://www.tensorflow.org/lite/microcontrollers/build_convert, viitattu 16.5.2023.
- [50] TENSORFLOW. Model conversion overview. URL <https://www.tensorflow.org/lite/models/convert/>, viitattu 26.5.2023.
- [51] TENSORFLOW. Post-training quantization. URL https://www.tensorflow.org/lite/performance/post_training_quantization, viitattu 29.5.2023.
- [52] TENSORFLOW. Tensorflow lite. URL <https://www.tensorflow.org/lite/guide>, viitattu 26.5.2023.
- [53] TRAFICOM. Radiotaajuusmääräys 4. URL https://www.finlex.fi/data/normit/48897/1_Radiotaajuusmaarays_4AD2023M_koonti.pdf, viitattu 5.6.2023.
- [54] TROMMEL, R., HARMANNY, R., CIFOLA, L., JA DRIESSEN, J. Multi-target human gait classification using deep convolutional neural networks on micro-

- doppler spectrograms. *Julkaisusarjassa 2016 European Radar Conference (EuRAD)* (2016), IEEE, 81–84.
- [55] VALTONEN, M., MAENTAUSTA, J., JA VANHALA, J. Tiletrack: Capacitive human tracking using floor tiles. *Julkaisusarjassa 2009 IEEE international conference on pervasive computing and communications* (2009), IEEE, 1–10.
- [56] WANG, J., CHEN, Y., HAO, S., PENG, X., JA HU, L. Deep learning for sensor-based activity recognition: A survey. *Pattern recognition letters* 119 (2019), 3–11.
- [57] WANG, X., KONG, L., KONG, F., QIU, F., XIA, M., ARNON, S., JA CHEN, G. Millimeter wave communication: A comprehensive survey. *IEEE Communications Surveys & Tutorials* 20, 3 (2018), 1616–1653.
- [58] YAMASHITA, R., NISHIO, M., DO, R. K. G., JA TOGASHI, K. Convolutional neural networks: an overview and application in radiology. *Insights into imaging* 9 (2018), 611–629.
- [59] ZHANG, Z., TIAN, Z., JA ZHOU, M. Latern: Dynamic continuous hand gesture recognition using fmcw radar sensor. *IEEE Sensors Journal* 18, 8 (2018), 3278–3289.