Jaro Väisänen

Konvoluutioneuroverkko metallimusiikkia sisältävän polyfonisen ääniraidan rumpuiskutapahtumien havaitsemisessa

Tietotekniikan pro gradu -tutkielma

3. joulukuuta 2020

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Jaro Väisänen

Yhteystiedot: jaro.vaisanen@gmail.com

Ohjaaja: Paavo Nieminen

Työn nimi: Konvoluutioneuroverkko metallimusiikkia sisältävän polyfonisen ääniraidan rumpuiskutapahtumien havaitsemisessa

Title in English: Detecting Drum Onsets In Polyphonic Metal Music Using Convolutional Neural Networks

Työ: Pro gradu -tutkielma

Opintosuunta: Ohjelmistotekniikka

Sivumäärä: 64+2

Tiivistelmä: Tässä tutkielmassa tutkittiin automaattista rumputranskriptiota käyttäen konvoluutioneuroverkkoa iskutapahtumien havaitsemiseen ja transkriptioon ääniraidasta, jossa on säröä sisältävää metallimusiikkia rumpujen seassa. Iskutapahtumien havaitseminen on perustavanlaatuinen ja tärkeä vaihe musiikkianalyysissä, kuten automaattisessa transkriptiossa ja rytmin tunnistamisessa. Automaattinen rumputranskriptio on nuotinnuksen tekemistä äänitetystä rumpujen soitosta, missä iskutapahtumien havaitsemista voidaan kuvailla prosessin ensimmäiseksi vaiheeksi. Rumpuiskutapahtumista pyrittiin havaitsemaan bassorumpua ja virveliä, jotka ovat hi-hatin lisäksi useimmiten valitut rumpuinstrumentit transkription haasteellisuuden vuoksi. Konvoluutioneuroverkoilla tehtynä transkription tulokset ovat tänä päivänä parhaita muihin menetelmiin verrattuna etenkin silloin, kun rumpuiskujen seassa on polyfonista musiikkia. Tutkimuksessa luotiin konvoluutioneuroverkkototeutus polyfonisten ääniraitojen rumpuiskutapahtumien havaitsemista varten. Menetelmän kehitysvaiheessa aineistona käytettiin ENST-rumputietokantaa, joka on vastaavissa tutkimuksissa laajasti käytetty rumpuaineisto. Menetelmän testaamisessa käytettiin ENST-rumputietokannan lisäksi *Riddle Me This* -yhtyeen studioäänityssessiosta saatuja ääniraitoja, joilla voitiin todentaa menetelmän toimivuutta metallimusiikissa. Tutkimuksen tuloksista selvisi, että virveliiskutapahtumia saatiin havaittua paremmin aiempaan tutkimukseen verrattuna, ja bassorum-

mun iskutapahtumia vain hieman paremmalla tarkkuudella. Menetelmä oli sovellettavissa myös metallimusiikkiaineistoon eroteltujen perkussiivisten elementtien ansiosta.

Avainsanat: iskutapahtuma, automaattinen rumputranskriptio, konvoluutioneuroverkko, metallimusiikki, polyfoninen musiikki, rumpujen soitto, signaalinkäsittely, STFT

Abstract: This paper examines automatic drum transcription using Convolutional Neural Networks (CNN) to detect drum onsets from polyphonic metal music. Onset detection is a fundamental task particularly in rhythm related musical analysis. It can be described as the first step in automatic drum transcription, where the aim is to produce an accurate notation of a musical composition. As the most common drum instruments used in onset detection, bass drum and snare drum were the target instruments applied in this paper. CNNs are said to produce the most prominent results obtained from transcription tasks particularly in the case where drum instruments are among polyphonic music. A method for detecting drum onsets from polyphonic music was implemented and evaluated as part of this thesis. The method is evaluated and trained using the publicly available ENST Drum database, allowing comparison with previous studies. The comparison shows that the developed method was able to detect snare drum onsets with better accuracy, and bass drum onsets with slightly improved accuracy compared to the previous method. The method is then evaluated in the metal music context using polyphonic tracks obtained from studio sessions by the metal band *Riddle Me This*. Applicability to the metal music dataset was also verified.

Keywords: onset detection, automatic drum transcription, convolutional neural network, metal music, polyphonic music, playing the drums, signal processing, STFT

Kuviot

Taulukot

Sisällys

1 Johdanto

Tämän tutkimuksen tarkoituksena on selvittää, kuinka konvoluutioneuroverkot soveltuvat rumpuiskutapahtumien havaitsemiseen metallimusiikkia sisältävän polyfonisen ääniraidan seasta. Iskutapahtumien havaitseminen (engl. *onset detection*) on matalan tason tehtävä, jonka ajatellaan olevan ensimmäinen askel automaattisessa transkriptiossa (Schlüter ja Böck 2014; Jacques ja Roebel 2018). Automaattisessa rumputranskriptiossa yksityiskohtaisesti erotelluista rummuista halutaan saada ulos symbolinen notaatio, joka kuvataan usein nuotinnuksen eli nuottiviivaston muodossa (Wu ym. 2018). Transkriptio on käänteinen prosessi musiikin luonnissa, jossa symbolista notaatiota luodaan ja mallinnetaan aiemmin äänitetystä musiikista (Wu ym. 2018). Tutkimuksessa tarkastellaan yksiraita-aineistoa, jossa bassorumpu ja virveli ovat polyfonisen musiikin seassa. Moniraita-aineistossa kunkin rummun iskutapahtumat voidaan havaita erikseen, jolloin niistä tehty transkriptio saadaan koostettua varmemmalla tarkkuudella (Paulus 2009). Käytännönläheisemmissä yksiraitaäänitteissä rumpujen erotteleminen polyfonisen musiikin seasta on tutkimusalan keskeisimpiä ongelmia, jonka takia tutkimuksen kontribuutio on tärkeä alan kannalta (Wu ym. 2018).

Aihe on merkittävä MIR (engl. *Music Information Retrieval*) -tutkimuskentässä, jonka suurimpien kontribuutioiden ajatellaan olevan musiikillisessa koulutuksessa ja musiikin tuotannossa (Dittmar ym. 2012; Wu ym. 2018; Wu ja Lerch 2015). Musiikillisen koulutuksen hyötyjä on nähty paljon mm. videopelialalla (Dittmar ym. 2012), jossa esimerkiksi rytmillisiä taitoja vaativat pelit voisivat hyötyä automaattisesta transkriptiosta (Wu ym. 2018). Esimerkiksi *Guitar Hero*, *Rock Band* ja *PaRappa the Rapper* -musiikkipeleissä pelaajan käsinäppäryys, rytmiset taidot sekä silmän ja käden välinen koordinaatio kehittyvät (Grollmisch, Dittmar ja Gatzsche 2009; Dittmar ym. 2012). Automaattisen transkription onnistuminen monipuoliselle datajoukolle voisi mahdollistaa lähes rajattoman määrän kappaleita sekä lisätä syvyyttä peleihin. Automaattinen transkriptio auttaa myös henkilöitä, joilla ei vielä ole tarvittavia taitoja tehdä kuullusta musiikista transkriptiota itsenäisesti (Dittmar ym. 2012). Musiikin ja rytmien automaattinen transkriptio mm. tähtää soittovirheiden tunnistamiseen reaaliajassa, jolloin soittaja kykenee paremmin itse arvioimaan suoritustaan, kun ammattitaitoista opettajaa ei ole saatavissa (Dittmar ym. 2012). Äänitteestä saadun nuotinnuksen

kautta tietyn kappaleen harjoitteleminen helpottuu merkittävästi (Wu ym. 2018; FitzGerald 2004; Dittmar ym. 2012). Transkriptiosta olisi hyötyä myös musikologisessa tutkimuksessa, jossa mikrorytmisten piirteiden, kuten swingin, shufflen ja grooven tunnistaminen on mielekästä (Wu ym. 2018; Davies ym. 2013; Dittmar, Pfleiderer ja Müller 2015; Dittmar ym. 2017, esimerkkeinä). Älykkäiden musiikkijärjestelmien kehittyminen on tärkeää alan ja yksilöiden musiikillisten taitojen kehittymisen kannalta (Wu ym. 2018; Dittmar ym. 2012). Kaikkiin edellä kuvattuihin esimerkkeihin vaaditaan tarkkaa iskutapahtumien havaitsemista. Tämän tutkimuksen tarkoituksena on täydentää aukkoa harmonista distortiota (säröä) sisältävien ääniraitojen iskutapahtumien havaitsemisessa. Esimerkiksi metallimusiikissa esiintyvää särökitaralle ominaista kohinaa sisältävän ääniraidan transkriptiosta on niukasti tutkimusta saatavissa: ainoa metallimusiikkiin keskittyvä julkaisulehti on *Metal Music Studies (Journal)* [1], jonka lisäksi metallimusiikki esiintyy pienessä määrin osana automaattisen rumputranskription ENST-tietokantaa (Gillet ja Richard 2006).

Iskutapahtumien havaitsemisen ja automaattisen rumputranskription merkittävimmät haasteet ovat tällä hetkellä polyfonisen musiikin seassa olevien iskutapahtumien erottelemisessa. Ongelman ratkaiseminen olisi merkittävä käytännön musiikkidatan monipuolisessa sovellettavuudessa (Wu ym. 2018). Konvoluutioneuroverkoilla tehtyä rumpuiskutapahtumien havaitsemista kuvaillaan nykyisistä vaihtoehtoisista menetelmistä parhaimmaksi (Jacques ja Roebel 2018).

Tässä tutkielmassa tarkastellaan erityisesti konvoluutioneuroverkoilla tehtävää rumpuiskutapahtumien havaitsemista polyfonisen musiikin seasta. Tutkimuksen tavoitteena oli luoda menetelmä iskutapahtumien havaitsemiseen aiemmin hyväksi todettujen menetelmien perusteella, tarkastella niihin liittyviä haasteita ja kehityssuuntia sekä koestaa menetelmää metallimusiikkiaineistolla. Tutkimuksessa toteutettiin järjestelmä, jossa opetettiin konvoluutioneuroverkkoa havaitsemaan rumpuiskutapahtumia esikäsitellyistä audiosignaaleista.

Luvussa 2 esitellään tutkielmassa käytetyt käsitteet ja merkinnät. Luvussa 3 esitellään aiemmin tehty tutkimus ja syvennytään käytettyyn kirjallisuuteen. Luvussa 4 perehdytään tutkimuksessa käytettyyn aineistoon ja perustellaan sen tarkoituksenmukaisuus tutkimuksen kannalta. Luvussa 5 esitellään tutkimuksessa kehitetty toteutus iskutapahtumien havaitsemisel-

1. https://www.intellectbooks.com/metal-music-studies

le. Luvussa 6 esitellään tutkimuksen tulokset. Luvussa 7 kerrataan tutkimuksen pääkohdat yhteenvedon muodossa.

2 Käsitteet ja merkinnät

Tässä luvussa esitellään tutkielmassa käytetyt käsitteet ja merkinnät. Lerch (2012) kuvaa tutkielman kannalta relevantteja signaalinkäsittelyn käsitteitä, sillä kirja keskittyy erityisesti audiosisällön analysointiin ja digitaalisen signaalinkäsittelyn rooliin MIR-tutkimuskentässä. Kirja on kattava teos digitaalisen signaalinkäsittelyn keskeisistä asioista, ja on siis hyvin linjassa tutkielman aiheen kanssa. Myös Tan ja Jiang (2013) ja Smith (1997) esittävät signaalinkäsittelyn periaatteita, joita on käytetty paikoin selityksen tukena. Hemanth ja Estrela (2017), Goodfellow, Bengio ja Courville (2016) ja LeCun ja Bengio (1998) kuvaavat konvoluutioneuroverkkoihin liittyviä käsitteitä ja merkintöjä. Esitellyt käsitteet ja merkinnät pysyvät yhdenmukaisina tutkielman loppuun saakka, minkä takia ne on tarpeen tuoda esille tutkielman alkuvaiheessa.

2.1 Signaalinkäsittely

Signaalinkäsittelyn tavoitteet tässä tutkielmassa keskittyvät hyödyllisen informaation johtamiseen, audiosisällön analysointiin ja transkriptioon vaadittavan metadatan johtamiseen. Signaali kuvaa, kuinka jokin parametri vaihtelee toisen parametrin suhteen (Smith 1997). Ihmisen havaitsema audiosignaali voidaan kuvata jatkuvana ajan suhteen muuttuvana äänenpainetason funktiona, jota voidaan prosessoida tietyssä systeemissä haluttujen tavoitteiden saavuttamiseksi (Lerch 2012). Lähtökohtaisesti jokainen sovelluskohtaisesti räätälöityä algoritmia parantava signaalin esikäsittelyvaihe katsotaan suotuisaksi. Signaalinkäsittelyssä systeemi on jokin prosessi, joka tuottaa ulostulosignaalin sisääntulevan signaalin perusteella. Systeemin sisällä voidaan esimerkiksi suodattaa signaalia halutusti, jolloin siitä on mahdollista johtaa hyödyllistä metadataa.

2.1.1 Näytteistäminen

Digitaalisesti ei ole mahdollista operoida jatkuvien signaalien kanssa, joten signaali täytyy diskretisoida eli näytteistää. Näyte (engl. *sample*) saadaan signaalista näytteistämällä se tietyistä kohdista, jolloin signaali diskretisoituu ajan suhteen. Näytteenottoteoreeman mukaan

signaali voidaan muodostaa uudelleen ilman informaation menetystä ainoastaan silloin, kun näytteenottotaajuus f_S on vähintään kaksi kertaa korkeampi kuin korkein näytteistettävä taajuus f_{max} eli

$$f_S > 2 \cdot f_{max}.$$

Jos signaali sisältää puolitetusta näytteenottotaajuudesta korkeampia taajuuskomponentteja, kyseiset komponentit aiheuttavat laskostumista (engl. *aliasing*). Tällöin ei-haluttuja signaaleja esiintyy halutuilla taajuusalueilla (Tan ja Jiang 2013). Smith (1997) puolestaan esitti laskostumisen siten, että tarkasteltava siniaalto omaksuu itselleen taajuuksia, jotka kuuluvat toiselle siniaallolle. Laskostumista aiheuttavia korkeita taajuuksia on kuitenkin mahdollista poistaa laskostumissuotimilla (engl. *Antialias filter*) (Smith 1997).

2.1.2 Konvoluutio

Konvoluutio muodostaa matemaattisen viitekehyksen digitaaliselle signaalinkäsittelylle, ja se sallii järjestelmien analysoinnin aika-alueessa. Konvoluutio on matemaattinen operaatio, jossa kukin ulostulon arvo on ilmaistu painotetuilla kertoimilla kerrottujen sisääntulevien arvojen summana (Smith 1997). Konvoluution tuloksena kahdesta signaalista muodostetaan kolmas signaali. Konvoluutio voidaan kuvata jonkin aikainvariantin järjestelmän impulssivasteena, joka on järjestelmän vastaanottaman signaalin ulostuloimpulssi. Kun järjestelmän impulssivaste on tiedossa, tiedetään kuinka kyseinen lineaarinen järjestelmä reagoi mihin tahansa impulssiin. Jokainen suodatusprosessi on konvoluutio, mutta audiosisällön analyysin kannalta mielenkiintoista on tarkastella suotimia, joilla on äärellisen pituinen impulssivaste. Tällainen voi esimerkiksi olla korkeataajuussuodin (ts. alipäästösuodin, engl. *low-pass filter*), jota myös tässä tutkielmassa on käytetty signaalin esikäsittelyvaiheessa.

Sisääntuleva signaali voidaan hajottaa (engl. *decompose*) impulssien joukkoon, joista kukin voidaan nähdä skaalattuna ja siirrettynä (engl. *shift*) deltafunktiona $\delta[n]$, jossa n on näyte. Deltafunktio on normalisoitu impulssi, jossa se saa arvon 1 näytteelle n ja arvon 0 muualla. Tällöin kunkin impulssin ulostulo on skaalattu ja siirretty versio impulssivasteesta $h[n]$. Lopullinen ulostulosignaali saadaan summaamalla skaalatut ja siirretyt impulssivasteet. Jos kyseessä oleva järjestelmä on suodin, impulssivaste on tällöin suodinydin, konvoluutioydin

tai pelkkä ydin. Tässä tutkielmassa ei käytetä pelkkää ydin-termiä sen tulkinnanvaraisuuden vuoksi signaalinkäsittelyn ja neuroverkkojen välillä.

Konvoluution tulos deltafunktion kanssa on signaali itse:

$$x(i) = \delta(i) * x(i),$$

jossa kunkin yksittäisen painotetun näytteen tulos voidaan laskea itsensä kanssa yhteenlaskettujen näytteiden summana.

2.1.3 Ikkunoitu Fourier-muunnos (STFT)

Fourier-muunnoksella signaali saadaan hajotettua yksittäisiin taajuuskomponentteihin ja niiden voimakkuuksiin. Signaali siis muunnetaan aika-alueesta taajuusalueeseen, jolloin tuloksena on spektri (engl. *spectrum*). Diskreetti Fourier-muunnos eli DFT (engl. *Discrete Fourier Transform*) on yksi tärkeimmistä audiosignaalien prosessointi- ja analysointityökaluista digitaalisessa signaalinkäsittelyssä. Diskreetti Fourier-muunnos voidaan laskea tehokkaasti nopean Fourier-muunnoksen eli FFT:n (engl. *Fast Fourier Transform*) avulla, jolloin laskenta-aika putoaa $O(\kappa^2)$:sta $O(\kappa log \kappa)$:aan. Fourier-muunnos ei kuitenkaan sellaisenaan kerro aikainformaatiota tarkastelun kohteena olevista taajuuskomponenteista, minkä takia signaalia on tarpeen käydä läpi liikkuvalla aikaikkunalla. Yhden pituudeltaan $\kappa = i_e(n) - i_s(n) + 1$ olevan ikkunan DFT on nimeltään STFT (engl. *Short-time Fourier Transform*), joka voidaan määritellä seuraavasti:

$$X(k,n) = \sum_{i=i_s(n)}^{i_e(n)} x(i) \exp\left(-\mathrm{j}k \cdot (i - i_\mathrm{s}(n)) \frac{2\pi}{\kappa}\right),$$

jossa k on diskretisoidun taajuuden (engl. *frequency bin*) indeksi, n on ikkunan indeksi, $i_s(n)$ ja $i_e(n)$ ovat n:nnen ikkunan alku- ja loppuindeksit, ja $\mathrm{j} = \sqrt{-1} \in \mathbb{C}$ viittaa luvun kompleksiosaan.

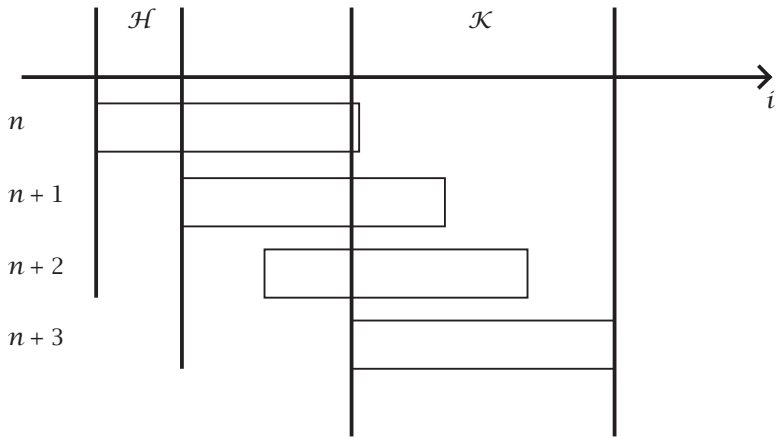
Ikkuna (engl. *window, frame, block*) on signaalin kyseisessä ajanhetkessä tarkastelun kohteena oleva jakso, joka sisältää n näytettä. Signaalin pilkkominen aikaikkunoihin sallii paremman muistiallokaation suositellulla ikkunakoolla, joka on $32 \leq \kappa \leq 16384$. Käytettäessä tällä välillä olevia ikkunakokoja laskentatehokkuus verrattuna yksittäisen näytteen prosessointiin on parempi, koska tällöin vältytään liiallisilta funktiokutsuilta ja vektorisoinneilta.

Äärimmäisissä tapauksissa ikkunakoko voisi olla $\kappa = 1$ tai koko ääniraidan kokoinen. Suositeltujen ikkunakokojen etenkin FFT:tä käytettäessä hyvän tehokkuuden saavuttamiseksi tulee olla 2:n potensseja, kuten 1024, 2048 tai 4096. Pienemmissä ikkunoissa näytteiden määrä on liian pieni tarpeellisen spektritiedon saamiseksi, ja isommissa ikkunoissa signaaliin alkaa tulla liikaa vaihtelua. Reaaliaikaisissa järjestelmissä täytyy kuitenkin pystyä prosessoimaan aikaikkuna korkeintaan samassa ajassa, jonka ikkunan näytteistäminen kestää, sillä tunnettuja ikkunoita ovat nykyisen ikkunan lisäksi vain aikaisemmin käsitellyt. Tässä tutkielmassa tarkastellaan ääniraitoja ns. offline-tilassa, jolloin kaikki näytteet ovat tunnettuja. Ikkunoiden välistä tilaa kutsutaan harppaukseksi $\mathscr{H}$ (engl. *hop size, stride, time shift*), joka kertoo näytteiden lukumäärän kunkin peräkkäisen Fourier-muunnetun ikkunan välillä. Harppauksen tulee olla pienempi tai yhtäsuuri kuin ikkunan pituus, jotta peräkkäiset ikkunat menevät päällekkäin. Ikkunan prosessointi voidaan määritellä

$$i_s(n) = i_s(n-1) + \mathscr{H},$$
$$i_e(n) = i_s(n) + \kappa - 1,$$

missä $i_s(n)$ on ikkunan aloittava näyte ja $i_e(n)$ ikkunan päättävä näyte. Ikkunoiden olisi suotavaa mennä toistensa kanssa päällekkäin, jotta saadaan mahdollisimman tarkkaa aika-taajuustietoa koko signaalista. Pienempi harppaus lisää tarkkuutta, mutta kasvattaa samalla kokonaisprosessointiaikaa.

Ikkunafunktiot, kuten Hamming- ja Hann-ikkunafunktiot pyrkivät ideaalitilanteessa tuomaan esiin yksittäisen impulssin taajuutta vähentämällä reunoilla olevien näytteiden amplitudia (Smith 1997). Tällöin kohina vähentyy huomattavasti, jolloin signaalin mielenkiintoisten piirteiden tarkastelu helpottuu. Tarkoituksena on pyrkiä minimoimaan pääkeilan leveys (engl. *Main lobe width*) ja sivukeilojen korkeus (engl. *Side lobe height*), jotka aiheuttavat laskostumista ympärillä oleviin taajuusalueisiin. Signaalin kertominen Hann-ikkunalla ennen diskreetin Fourier-muunnoksen tekemistä korostaa haluttujen taajuuksien huippupisteitä, mutta niistä tulee samalla leveämpiä. Ikkunafunktioiden käyttö pakottaa valitsemaan erottelukyvyn (engl. *resolution*) eli huippupisteen leveyden ja spektrivuodon (engl. *spectral leakage*) eli sivukeilojen amplitudin välillä (Smith 1997). Spektrivuoto määräytyy pääasiassa pääkeilan leveyden, lähimmän sivukeilan korkeuden sekä peräkkäisten sivukeilojen vaimentumisen kautta (Lerch 2012).

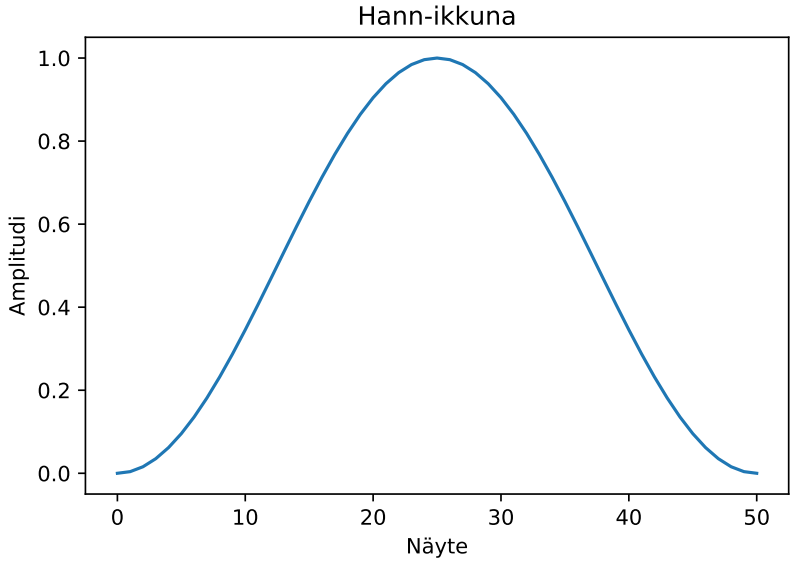


Kuvio 1: Signaalin jaetut kooltaan κ olevat ikkunat menevät toistensa kanssa päällekkäin, kun niitä siirretään harppauksen $\mathscr{H}$ verran näytteitä edeltävästä ikkunasta.

Ikkunafunktion käyttö kaventaa tarkasteltavan ikkunan sisässä olevan signaalin teräviä reunoja, jotka voivat tuoda taajuusalueeseen ei-toivottuja artifakteja. Signaalista pyritään siis saamaan jaksollisen kaltainen, jossa ikkunan reunat kaventuvat samoihin arvoihin. Tyypilliset ikkunafunktiot ovat symmetrisiä, joissa arvojen painotus tapahtuu funktion keskellä oleville näytteille. Tällaiset ikkunat mahdollistavat tarkastelun kohteena olevan impulssivasteen pidentämisen. Kuviossa 2 on nähtävissä tässä tutkielmassa käytetyn Hann-ikkunafunktion kuvaaja ja kuviossa 3 sen taajuusvaste.

2.1.4 Spektri ja spektrogrammi

Fourier-muunnoksesta ei kuitenkaan käy suoraan ilmi, kuinka signaalin taajuussisältö vaihtelee ajan suhteen; informaatio taajuussisällön vaihtelusta on selvitettävissä vaiheen (engl. *phase*) avulla. Tarkastelun kohteena olevan signaalin taajuuksien kirjoa ja niiden magnitudi-informaatiota kutsutaan spektriksi. Signaalin taajuussisällön vaihtelu voidaan selvittää jakamalla signaali ikkunoihin ja laskemalla spektri kullekin ikkunalle STFT:n avulla. STFT:n laskemista varten tulee määrittää ikkunaominaisuudet, kuten ikkunan pituus ja harppaus $\mathscr{H}$ eli ikkunoiden keskinäinen päällekkäisyys.

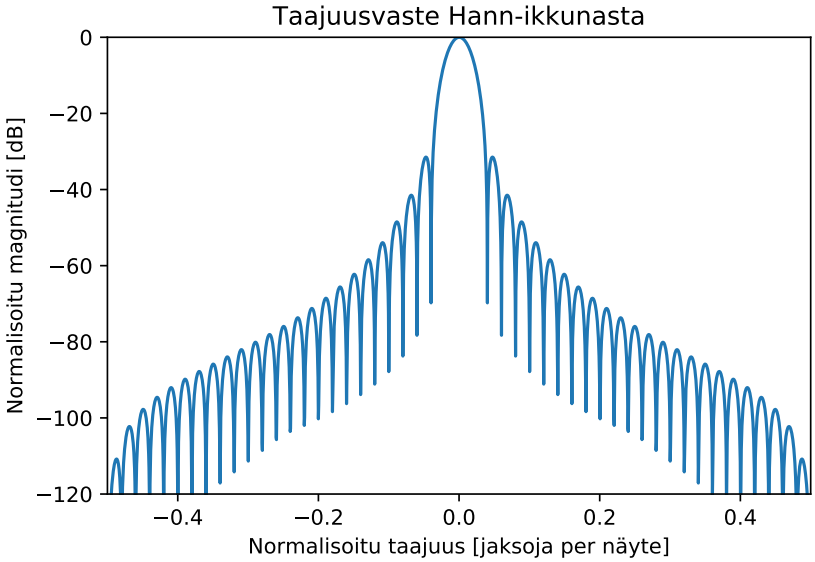


Kuvio 2: Hann-ikkunafunktio silottaa käsiteltävää signaalia reunoilta, jolloin kyseessä oleva taajuus korostuu funktion keskikohdassa laskostuen symmetrisesti.

Analysoitavan Fourier-muunnetun signaalin aika-taajuusominaisuuksista joudutaan valitsemaan joko suotuisamman taajuusresoluution tai aikaresoluution väliltä, sillä STFT:ssä käytetyn ikkunan kokonaisresoluutio on kiinnitetty (Smith 1997). Tätä valintaa kutsutaan epämääräisyysperiaatteeksi (engl. *Uncertainty principle*). Taajuusresoluutio kertoo, kuinka tiheässä Fourier-muunnetun spektrin näytteistetyt taajuudet ovat. Se voidaan määritellä kaavalla

$$f_Q = f_s/\kappa,$$

jossa f_s on näytteenottotaajuus. Taajuusresoluutiosta tulee täten tarkempi, kun ikkunakoko κ kasvaa. Pidempi kooltaan κ oleva ikkuna sallii tarkemman taajuusresoluution, mutta toimiakseen hyvin ikkuna vaatii jaksollisen ja muuttumattoman signaalin. Tällöin ikkunoista voidaan erotella lähellä olevia taajuuksia toisistaan tarkemmin, koska signaalin pääkeilasta pääteltävä taajuusinformaatio on keskittynyt kapeammalle alueelle. Tässä tutkielmassa tarkasteltava signaali ei ole muuttumaton, mikä tuo haastetta taajuuksien analysointiin. Pidemmän ikkunan spektrogrammi on kapeakaistainen (engl. *narrowband spectrogram*), jossa DFT-pisteitä on enemmän antaen tarkemman taajuusresoluution, mutta vähemmän tarkkuutta ajan suhteen. Lyhyempi ikkuna puolestaan sallii tarkemman aikaresoluution, jolloin esimerkiksi rumpuiskutapahtumien alkupisteiden havaitseminen onnistuu paremmin. Lyhyem-

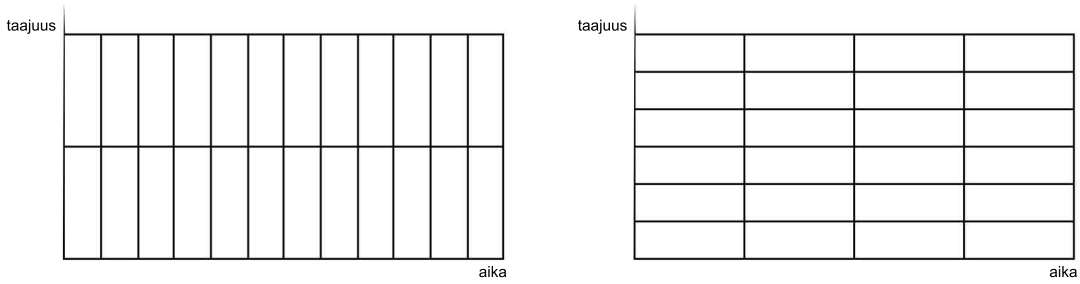


Kuvio 3: Signaalin taajuusvaste käyttäen 2048 kokoista ikkunaa ja Hann-ikkunafunktiota. Keskellä oleva pääkeila on optimaalisessa tilanteessa melko kapea, ja sivukeilat suhteellisen matalia normalisoidun magnitudin suhteen.

män ikkunan spektrogrammi on laajakaistaisempi (engl. *wideband spectrogram*), jossa DFT-pisteitä on vähemmän, mutta aikasiivuja on enemmän. Tällöin saadaan tarkkuutta siirtymien ajankohtiin. Kompromissia ajan ja taajuuden erottelukyvyn välillä on havainnollistettu kuviossa 4.

Audiosignaalin aika- ja taajuuskomponenttien visuaalista kombinaatiota kutsutaan spektrogrammiksi. STFT lasketaan kullekin päällekkäiselle dataikkunalle, jotka visualisoidaan (pseudo)kolmiulotteiseksi kuvaksi. Signaalin kuvattu aika esiintyy usein spektrogrammin vaaka-akselilla ja taajuudet pystyakselilla. Kolmiulotteisessa spektrogrammissa signaalin magnitudi (dB) on esitetty tummennettuina kohtina äänentason voimakkuudesta riippuen, missä kukin STFT on yksi sarake (kuvaajassa 5 pystyviiva).

Tässäkin tutkimuksessa käytetty mel-spektrogrammi on vaihtoehtoinen esitysmuoto perinteiselle spektrogrammille. Mel-spektrogrammissa audiosignaalin taajuudet on skaalattu mel-skaalaan, joka ilmentää paremmin ihmisen havaitsemia taajuuksia. Mel-spektrogrammi muodostetaan STFT:n pohjalta valitulla ikkunakoolla, ja koko spektri skaalataan tasaisesti valittuun määrään taajuusosia (esim. 80 tai 128). Tasaisuus mel-skaalassa tarkoittaa yhtä mittaisia



Kuvio 4: Spektri ilmenee käytetyn ikkunakoon mukaan tarkemmin joko aikaresoluution tai taajuusresoluution suhteen. Kuvassa vasemmalla aikaresoluutio on tarkempi, ja oikealla taajuusresoluutio on tarkempi.

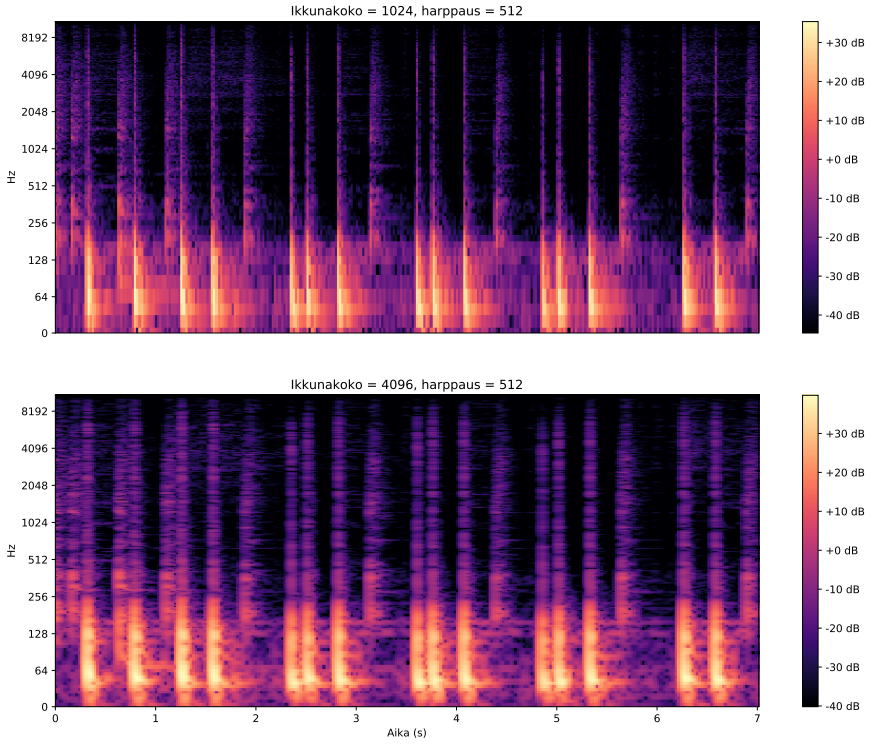
välejä ihmisen kuulemien taajuuksien perusteella.

2.2 Konvoluutioneuroverkot

Konvoluutioneuroverkot soveltuvat erityisesti kuvan luokitteluun niiden useiden syvien kerrosten, piirteiden valinnan (engl. *feature selection*) ja vastavirta-algoritmien (engl. *back-propagation*) ansiosta. Konvoluutioneuroverkot suorittavat perinteisten neuroverkkojen matriisikertolaskujen lisäksi konvoluutio-operaatioita. Tässä tutkielmassa luokittelua tehtiin mel-spektrogrammidatalla, jolla opetettiin neuroverkkoa havaitsemaan iskutapahtumia. Seuraavissa luvuissa käsitellään konvoluutioneuroverkkojen arkkitehtuuria, opettamista ja evaluointia.

2.2.1 Arkkitehtuuri

Konvoluutioneuroverkon arkkitehtuuri koostuu sisääntulokerroksesta (engl. *input layer*), jopa sadoista piirteiden havaitsemiskerroksista (engl. *feature detection layer*) ja täysin yhdistetyistä kerroksista (engl. *fully connected layer*). Piirteiden havaitsemiskerroksen tehtävänä on suorittaa joko konvoluutiota, yhdistämistä (engl. *pooling*) tai operoinnin muuttamista epälineaariseksi käyttämällä esimerkiksi oikaistua lineaarista yksikköä (engl. *Rectified Linear Unit (ReLU)*) aktivaatiofunktiona. Konvoluutiokerroksia seuraa yleensä yhdistämiskerros, ja



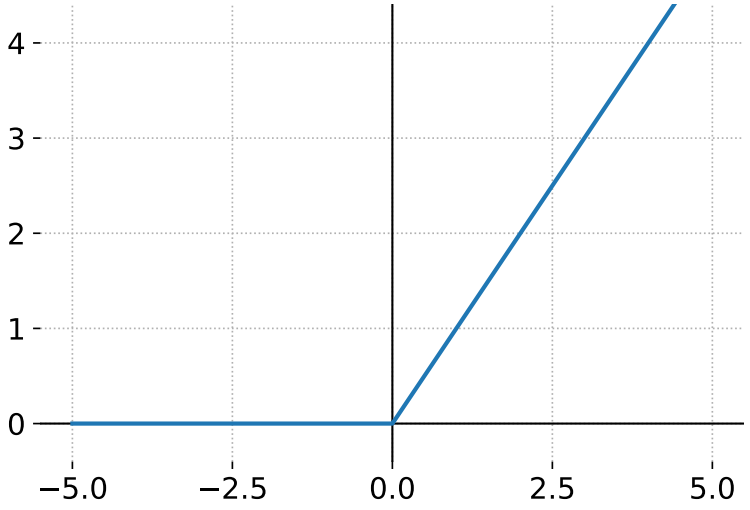
Kuvio 5: Bassorumpuraidan spektrogrammiesitykset, joissa ylemmän spektrogrammin 1024 näytteen kokoinen ikkuna antaa paremman aikaresoluution verrattuna alempaan spektrogrammiin, jossa 4096 näytteen kokoisella ikkunalla saadaan parempi taajuusresoluutio. Harppaus $\mathscr{H} = 512$.

näitä vuorotellaan keskenään osana piilotettujen kerrosten joukkoa. Aktivaatiofunktio merkitään yleensä osana konvoluutiokerrosta.

Konvoluutioneuroverkoissa vierekkäisten kerrosten neuronien yhteydet vahvistuvat kerrosten edetessä. Konvoluutiokerroksissa neuronit ovat kaksiulotteisessa taulukossa, josta saadaan muodostettua piirrekarttoja (engl. *feature map*) seuraavia kerroksia varten. Piirrekartat koostuvat reseptiivisistä kentistä (engl. *receptive field*), jotka ovat tiettyjä kohtia tarkasteltavasta kuvasta. Konvoluutiokerros koostuu useista piirrekartoista, jotta voidaan havaita useita piirteitä kustakin sijainnista. Havaitut yhteydet perustuvat konvoluutiomaskiin (engl. *kernel*), joka on yksi käyttäjän vaikuttamista hyperparametreistä. Konvoluutio käyttää esiase-

tuksia (engl. *bias*) neuronien laskennassa, joista saadaan ulostulo aktivaatiofunktiota varten. Konvoluutiokerroksissa kuva viedään suotimien läpi, jolloin kuvan tietyt piirteet aktivoituvat seuraavia kerroksia varten. Tällöin voidaan tunnistaa kuvasta mielenkiintoisia kohteita, kuten reunoja ja kulmia, ja myöhemmillä kerroksilla korkeamman tason tietoa toistuvista kuvioista. Yhdistämiskerrokset suorittavat epälineaarista alaspäin näytteistämistä (engl. *downsampling*), jolloin käsiteltävän datan määrä supistuu. Aktivaatiokerroksissa paljon käytetty ReLU (ks. kuvio 6) pitää positiiviset aktivoidut arvot tallella, mutta kuvaa negatiiviset arvot nollaan:

$$f(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases}$$



Kuvio 6: Paljon käytetty ReLU-aktivaatiofunktio kuvaa negatiiviset aktivoidut arvot nollaan.

ReLU on yksinkertaisuutensa vuoksi tehokas laskennallisesti, minkä takia sitä on käytetty myös tässä tutkielmassa piilotetuilla kerroksilla aktivaatiofunktiona.

Ennen ulostuloa sijaitseva luokittelukerros on korkean tason informaatiota johtava täysin kytketty kerros, jossa tulokset kuvataan johonkin luokka-arvoon ennaltamäärättyjen L vaihtoehtojen eli tunnisteiden (engl. *label*) joukosta. Mitä enemmän dataa neuroverkon opetusvaiheessa syötetään, sitä yleiskäyttöisempi neuroverkko on eri ongelmien ratkaisemisessa.

Laskenta-aika hyvin suurella datamäärällä kasvaa kuitenkin runsaasti, jolloin opettamiseen on tarpeen käyttää tehokkaampia laskentayksikköjä.

Neuroverkon arkkitehtuuria ja opetusalgoritmia säädetään hyperparametreillä, joiden kombinaatiot vaikuttavat merkittävästi opettamiseen ja lopputuloksiin. Käyttäjän säädettävissä olevia hyperparametrejä ovat mm. konvoluutiomaskien ja yhdistämiskerrosten suotimien koot, harppaus (engl. *stride*), laajenemisarvo (engl. *dilation rate*), kerrosten sisään- ja ulostulojen pehmustus (engl. *padding*), kerrosten järjestys, regularisointi, pudotus (engl. *dropout*), epookkien määrä, käytetty optimoija ja sen oppimistahti (engl. *learning rate*) sekä eräkoko (engl. *batch size*).

Neuroverkon opetuksessa epookki on koko opetusaineiston yksi läpikäynti. Opetus voi kestää jopa useita tuhansia epookkeja, jotta takaisinkytkentäalgoritmi konvergoituu painotusten ja esiasetusten avulla hyväksyttävään tarkkuuteen. Neuroverkon optimoija säätää opetusta epookkien välillä annettujen hyperparametrien, kuten oppimistahdin ja eräkoon avulla. Oppimistahti määrittää, kuinka nopeasti malli sopeutuu ongelmaan ja sen yleistettävyyteen. Oppimistahtia lasketaan asteittain, ja sen alustavan arvon valinta on kokeiltava ongelmakohtaisesti. Liian pienellä oppimistahdilla oppiminen etenee hitaasti ja voi jäädä paikoilleen, ja vaatii yleensä enemmän epookkeja suoriutuakseen. Suuremmalla oppimistahdilla oppimistahdilla havainnoitu oppimiskäyrä heilahtelee rajusti. Oppimistahdin valinnassa suositellaan tarkasteltavan kuvaajia oppimiskäyristä ja vaihteluista, jotka auttavat valitsemaan ongelmaa varten sopivimman arvon (Goodfellow, Bengio ja Courville 2016).

Eräkoko on yksi tämän tutkimuksen keskeisistä hyperparametreistä, sillä kehitetyn menetelmän toimivuutta tarkastellaan eräkokokohtaisesti. Neuroverkkoa opetetaan erä (engl. *minibatch*) kerrallaan, jolloin neuroverkkomallin painotuksia ja parametrejä päivitetään satunnaisesti valittujen näytteiden perusteella. Eräkoko määrittää neuroverkon opetuksen läpi kerralla kulkevien näytteiden lukumäärän. Eräkoko on usein suhteellisen pieni opetusaineiston kokoon verrattuna, jolloin neuroverkon tekemää virhettä pyritään optimoimaan sopivin askelin. Käytettyjä eräkokoja ovat esimerkiksi 32, 64, 128 ja 256. Pienemmillä eräko'oilla voi olla regularisoivia vaikutuksia, sillä ne lisäävät opetusprosessiin kohinaa. Tällöin on usein perusteltua käyttää myös pienempää oppimistahtia, mikä lisää opetukseen tarvittavaa aikaa. Suuremmilla eräko'oilla malli näkee enemmän opetusnäytteitä kerralla, jolloin tappiofunk-

tiota (engl. *loss function*) optimoivan gradientin muutos on suurempi. Gradientti on vektori, joka määrittää minimoitavan tappiofunktion muutoksen suunnan ja suuruuden. Suuremmat eräkoot vaativat myös enemmän muistia laskentaan käytettävältä laitteistolta. Kun laskentaa suoritetaan rinnakkain GPU:illa, on tavanomaista käyttää toisen potenssin eräkokoja tehokkuuden lisäämiseksi.

2.2.2 Opettaminen ja evaluointi

Neuroverkon opettamisen ja evaluoinnin tavoitteena on tarkastella neuroverkon kykyä suoriutua ratkaistavasta ongelmasta ja reagoida niistä saatavaan palautteeseen, jotta mallia voidaan parantaa (Goodfellow, Bengio ja Courville 2016). Palautetta tarkastellaan metriikoiden muodossa, joiden perusteella voidaan tehdä harkittuja päätöksiä, jotta opettaminen onnistuisi paremmin ilman sokeita arvailuja. Palaute voi kieliä mm. suuremman datajoukon tarpeesta, mallin piirteistä tai regularisoinnin ja optimoinnin säätelystä. Neuroverkon opettamiseen liittyvien toistuvien ja lisääntyvien muutosten tekeminen on tärkeää, sillä toimivan mallin rakentaminen on ongelmakohtaisesti ratkottava iteratiivinen prosessi.

Opettamiseen ja evaluointiin käytetty aineisto koostuu opetus-, validointi- ja testiaineistosta. Mitä suurempi opetusaineisto on, sitä enemmän esimerkkejä malli näkee ja pystyy todennäköisemmin löytämään ongelmaan paremman ratkaisun. Validointiaineisto auttaa löytämään parhaan mallin, sillä opetusaineistolla vahvistetun mallin opettaminen pysäytetään validointiaineistoa vasten arvioitavan tappiofunktion perusteella ylisovittamisen välttämiseksi. Testiaineiston tehtävänä on todentaa mallin yleistettävyys datassa, jota se ei ole aiemmin nähnyt. Jos testidataa on liian vähän, mallin kykyä toimia yleiskäyttöisenä luokittelijana ei pystytä todentamaan luotettavasti. Tällöin evaluoinnista johdettu tarkkuus on heikko. Testiaineisto ei saa sisältää samoja näytteitä kuin opetus- ja validointiaineisto. Jos testiaineisto suoriutuu heikosti, voi olla tarpeen lisätä testijoukon ja koko aineiston määrää, tai säätää regularisoivia hyperparametrejä (Goodfellow, Bengio ja Courville 2016).

3 Aiemmin tehty tutkimus

Tässä luvussa esitellään aiemmin tehtyä tutkimusta, tutustutaan käytettyyn lähdekirjallisuuteen, sekä perustellaan aiemman tutkimuksen valossa tämän tutkielman aiheen olevan uutta tieteellistä tietoa. Kirjallisuuskatsauksen lopuksi esitellään ja avataan tarvittavat käsitteet. Kirjallisuutta ja aiemmin tehtyä tutkimusta esitellään osittain aikajärjestyksessä ja tutkimusten tapahtumaketjujen kehittymisen kannalta oleellisessa järjestyksessä, josta ilmenee automaattisen rumputranskription keskeisimmät elementit, tutkimuksen kehitys ja tulokset sekä konvoluutioneuroverkkojen paikka tutkimuskentässä. Iskutapahtumien havaitsemiseen liittyen tarkastellaan erityisesti polyfonisen musiikin seassa esiintyviä rumpuiskutapahtumia.

3.1 Automaattinen rumputranskriptio

Automaattinen musiikin transkriptio on käänteinen prosessi, jossa musiikin symbolista notaatiota luodaan ja mallinnetaan aiemmin äänitetystä musiikista (Wu ym. 2018, 1457). Automaattisen rumputranskription järjestelmät havaitsevat äänisignaalissa olevia iskutapahtumia tai mielenkiintoisia impulsseja, joista voidaan havainnollistaa tietyn rumpuinstrumentin paikka nuottiviivastolle (Wu ym. 2018, 1457). Onnistuneesta transkriptiosta on paljon hyötyä MIR-tutkimuskentässä, ja sen avulla voidaan johtaa korkeamman tason informaatiota, kuten kappaleen tempoa ja genreinformaatiota (Vogl, Dorfer ja Knees 2017). Seuraavaksi esittelen haasteita ja menetelmiä automaattiseen rumputranskriptioon liittyen, jotta tutkimusalan nykyinen tilanne ja tutkielman tavoitteet tulevat ilmi.

3.1.1 Haasteet

Automaattinen rumputranskriptio on paljon tutkittu aihe, jossa riittää haasteidensa vuoksi vielä runsaasti jatkotutkimusta (Wu ym. 2018; Benetos ym. 2013). Yhdet ensimmäisistä digitaalisen signaalinkäsittelyn transkriptiojärjestelmistä on kehitetty 70-luvulla, jolloin esimerkiksi Moorer (1975) jätti vielä väitöskirjassaan perkussiiviset instrumentit pois niihin liittyvien haasteiden vuoksi. Perkussiivisten instrumenttien äänityksessä signaaliin tulee usein paljon jälkikaikua huoneen heijastuksista, mikä lisää signaalissa esiintyvän vaiheen ja ampli-

tudin tärinää (engl. *jitter*). Väitöskirjassa esitelty musiikkisignaalin analysointi keskittyi tunnistamaan vain kyseessä olevia nuotteja; instrumentteja ei yritetty tunnistaa. Tuohon aikaan ilmenneiden haasteiden vuoksi transkriptioon jouduttiin tekemään rajoituksia: musiikkiotteissa ei saanut olla kahta itsenäistä ääntä enempää, vibratoja tai glissandoja ei saanut esiintyä, nuottien piti olla vähintään 100 ms pituisia, eikä nuotin perustaajuus (engl. *fundamental frequency*) saanut sattua samalle hetkelle samalta kuulostavan eri taajuisen nuotin huiluäänen (engl. *harmonic*) kanssa. Tämän seurauksena oktaavit ja monet intervallit jäivät pois analyysistä. Esimerkiksi samankaltaisten symbaalien korkeat taajuudet olisivat kyseisessä järjestelmässä voineet mennä pahasti sekaisin, sillä niiden äänet koostuvat laajoilla taajuuskaistoilla tapahtuvasta kohinasta eikä tietyistä sävelkorkeuksista (engl. *pitch*). Perkussiivisilla äänillä ei myöskään ole tunnistettavia tasaisesti muuttuvia osittaistaajuuskomponentteja (engl. *partial tone*) (Lerch 2012), joiden löytyminen musiikkisignaalista oli välttämätöntä transkription tekemiseen Moorerin 1975 menetelmässä.

Sittemmin Schloss (1985) väitöskirjassaan kehitti automaattisen transkription järjestelmään parannuksia keskittyen perkussiivisiin instrumentteihin. Väitöskirjan tavoitteena oli transkriptiojärjestelmän kehittämisen lisäksi analysoida ja ymmärtää musiikissa esiintyvää rytmiä ja muita korkean tason MIR-kohteita tietokoneen avulla. Tutkitut haasteet liittyivät ajoituksen ja rytmillisen monimutkaisuuden käsittelemiseen täyteläisessä musikaalisessa kattauksessa. Väitöskirja esitteli aiempiin tutkimuksiin nähden uusia haasteita. Äänet eivät ole aina jaksollisia, joten iskutapahtumien luonteisiin jouduttiin keskittymään täsmällisemmin. Ensimmäistä kertaa iskutapahtumia havaittiin sävelkorkeuden sijaan iskujen luonteiden perusteella. Transkriptioon vaikuttaa merkittävästi rumpuiskun lyöntiajankohdan lisäksi myös se, kuinka rumpua lyödään (Schloss 1985). Rummun tonaalisuus vaikuttaa perkussiivisen iskun ilmenemiseen spektrogrammissa, josta se on neuroverkon tulkittavissa. Helpommin lähestyttävä vaihtoehto tässä on länsimainen musiikki, jossa perkussiivisilla instrumenteilla on vähemmän harmonisia piirteitä (Schloss 1985). Iskutapahtumien alkupisteistä, pituuksista, amplitudeista, ja muista musiikille tärkeistä ominaispiirteistä koostuva nuottilista luotiin iskujen ajoituksista ja ominaisarvoista. Näiden jälkeen voitiin analysoida korkeamman tason tietoa, kuten rytmiä, musiikkilajia, rakennetta ja jaksollisuutta. Schlossin kehittämä järjestelmä käsittelee myös aiemmin kuvatut haasteet, joita Moorer (1975) joutui asettamaan rajoituksiksi väitöskirjassaan. Schloss (1985) ja Moorer (1975) käyttivät vielä DFT:ta ja FFT:tä

äänisignaalin muuntamiseen taajuuskomponenttimuotoon, mutta myöhemmin STFT alkoi vakiintua käytettäväksi menetelmäksi, koska se on toimivampi polyfonisessa kontekstissa (Wu ym. 2018; Jacques ja Roebel 2018). STFT:n avulla voidaan ottaa signaalin taajuuskomponenttiesitykset eri ikkunako'oilla tietystä temporaalisesta sijainnista, jolloin analyysi on tarkempaa ja tehokkaampaa (Lerch 2012). Oleellisena haasteena Schloss (1985) korostaa automaattisen segmentoinnin tärkeyttä musikaalisen materiaalin manipuloinnissa, sekä tarvetta hyvälle signaalin äkillisten voimakkuuksien muutosten tunnistusalgoritmille iskutapahtumien havaitsemisessa. Iskutapahtumien havaitsemiseen perehdytään tarkemmin luvussa 3.2.

Ensimmäiset rumputranskriptiojärjestelmät perkussiivisten lähteiden erotteluun polyfonisesta musiikista ovat kehitelleet Goto ja Muraoka (1994) 90-luvun alkupuolella (FitzGerald 2004). Paulus (2009) kuvailee väitöskirjassaan, että haastetta automaattisessa rumputranskriptiossa tuottaa erityisesti yksittäisten iskujen ja rumpujen erotteleminen äänitetystä soitosta, jossa signaali täyttyy muista sisääntulevista äänistä. Tällöin rumputranskriptiota tehdään polyfonisen musiikin seasta. Transkriptio pystytään nykyisin melko hyvin tuloksin suorittamaan pelkälle rumpuraidalle, mutta polyfonisen musiikin seasta tehty rumputranskriptio ei vielä tuota tarpeeksi tyydyttäviä tuloksia (Vogl, Dorfer ja Knees 2017; Dittmar ja Gärtner 2014). Vaikka perkussiiviset ja melodiset instrumentit ovat perusominaisuuksiltaan hyvin erilaisia, esimerkiksi bassorummun äänikomponentit menevät useimmiten päällekkäin bassokitaran kanssa, virveli kitaran ja pianon kanssa, ja symbaalit kohinalle omaisten luonteidensa takia laajan taajuusalueen kanssa (Wu ym. 2018). Etenkin polyfonisen musiikin kontekstissa perkussiivisten instrumenttien transkriptioon liittyen mielekästä on tarkastella ainoastaan iskutapahtumien alkupisteitä (engl. *onset*); nuotin tai iskun päättymiskohdalla (engl. *offset*) ei ole perkussiivisten instrumenttien luonteen takia suurta merkitystä rytmillisessä analyysissä (Benetos ym. 2013). Goto ja Muraoka (1994) saivat ensimmäisessä polyfonisen musiikin rumputranskriptiojärjestelmässään havaittua esimerkiksi avonaisen hihatin päättymiskohdan hyvällä menestyksellä totuusarvoon (engl. *ground truth*) verrattuna. Polyfonisesta musiikista johdetuilla transkriptioilla voidaan myös johtaa korkeamman tason musiikki-informaatiota, kuten kappaleen tempoa, tahteja ja rytmitietoa (Jacques ja Roebel 2018; Vogl, Dorfer ja Knees 2017).

Gillet ja Richard (2008) kuvailevat hiljaisten iskujen, kuten virvelin haamuiskujen ja muiden vastaavien luonteeltaan heikompien nuottien olevan vaikeita löytää polyfonisen musiikin seasta. Haamuiskuilla voi kuitenkin olla merkittävä osuus grooven ilmenemisessä, ja niiden puuttuminen voi olla haitallista transkriptiolle. Haasteellisiksi Gillet ja Richard (2008) kuvailevat myös vispilöiden ja mallettien iskujen transkription varsinkin polyfonisen musiikin seasta.

Transkriptioaineiston keruussa eli musiikin äänitysvaiheessa huoneen akustiikan vaikutukset signaaliin voivat olla huomattavat esimerkiksi lisääntyneen kaiun takia. Äänittäjä mitä todennäköisimmin muokkaa signaalia tasapäistämällä (engl. *equalize*) haluttuja taajuusalueita kuuntelijalle mieluisaksi ja suodattamalla mikrofonisignaalia, jolloin tehdyt signaalimuutokset voidaan mallintaa konvoluutiona yhden tai useamman impulssivasteen kanssa (Wu ym. 2018). Tavallisesti signaalia käsitellään vielä ainakin epälineaarisilla efekteillä, kuten dynaamisella kompressoinnilla ja säröllä. Käytetyssä datajoukossa ei tulisi kuitenkaan olla liikaa pelkästään jälkikäsiteltyjä signaaleja, sillä neuroverkon oppimisvaiheessa tapahtuisi muuten ylisovittamista tietynlaisten äänitysolosuhteiden suhteen. Datajoukkoa on tarpeen laajentaa tuottamalla runsaasti vaihtelua esimerkiksi kaikuun, säröön ja dynaamiseen prosessointiin. (Wu ym. 2018). Vogl, Dorfer ja Knees (2017) kuvailevat, että etenkin syviä neuroverkkoja käyttävissä menetelmissä luotettavuutta tulisi lisätä datajoukon augmentaatiolla ja pudottamisella (engl. *dropout*), jotka vähentävät ylisovittamista. Eri tyyppiset rummut (vrt. latino- ja rock-rummut) soivat erilaisilla taajuusalueilla, minkä takia yleistettävyyden vuoksi datajoukkojen tulisi olla mahdollisimman suuria ja monipuolisia (Gillet ja Richard 2008).

Akustisen rytmisoitinsignaalin transkription tavoitteena on ensisijaisesti tarkan nuottinotaation automaattinen luominen, mutta myös tarkoitus ymmärtää rytmin käsitystä ja sen tuottamista, sekä ymmärtää paremmin säveltäjän ja esiintyjän välistä suhdetta (Schloss 1985). Länsimaisessa populäärimusiikissa rummut ja perkussiot ovat keskeisessä elementissä, joka useimmiten määrittää rytmin muodon, tarkoituksen sekä kyseessä olevan musiikkityylin (Wu ym. 2018). Schloss (1985, 48) kuvailee väitöskirjassaan länsimaisessa musiikissa perkussiivisten instrumenttien olevan harvemmin soolo-osuuksina; ne ovat useimmiten polyfonisen musiikin seassa. Sen sijaan afrikkalaisessa musiikissa rytmin merkitys musiikille ja musikaaliselle ilmaisuvoimalle korostuu merkittävästi. MIR-tutkimuskentässä rytmiikan ko-

konaisvaltaisessa tutkimisessa huomioon olisi hyvä ottaa usein käytetyn länsimaisen musiikin lisäksi myös afrikkalainen musiikki. Rummun kalvo ja runko vaikuttavat merkittävästi äänen sointiväriin ja sävelkorkeuteen, minkä takia akustiset piirteet ja eri tyyppiset rummut tulisi ottaa huomioon rytmiikkaan liittyvässä tutkimuksessa (Schloss 1985).

Signaalinkäsittelyä varten akustinen signaali pitää ensin näytteistää, jolloin se diskretisoidaan ajan suhteen digitaalisesti käsiteltävään muotoon, ja kvantisoida (engl. *quantization*), jolloin se diskretisoidaan signaalin amplitudin mukaan (Lerch 2012). Tietokoneen kyky havaita rumpuiskutapahtumia laskennallisesti sallisi mm. rytmin ja rytmiikan laajemman analysoinnin (Wu ym. 2018).

Nykyisen tutkimustilanteen valossa automaattisen rumputranskription suurimmiksi nähtävät haasteet ovat datajoukkojen niukkuudessa, ADT-yhteisön (engl. *Automatic Drum Transcription*) kehittymisessä, rumpuinstrumenttien (kuten tomien ja muiden symbaalien) lisäämisessä, dynaamisten yksityiskohtien tunnistamisessa, signaalin esi- ja jälkikäsittelymenetelmissä, kielimallien integroimisessa sekä yleisesti kappaleen tai musiikkinäytteen kokonaisuuden kattavassa transkriptiossa. (Wu ym. 2018). Aihetta ovat tutkineet myös Mesaros ym. (2010) ja Rabiner (1989) sekä Paulus ja Klapuri (2009) käyttäen erityisesti Markovin piilomallia (engl. *Hidden Markov Model (HMM)*), joka on transkription lisäksi mm. puheentunnistuksessa paljon käytetty tilastollinen malli.

Vielä ei ole olemassa loppukäyttäjille suunnattua sovellusta, joka kykenisi luotettavasti transkriboimaan useita soittimia sisältävää äänitettyä musiikkia. Tuoreimpienkaan järjestelmien suorituskyky ei yllä lähelle ammattitaitoisen ihmisen tekemään transkriptiota, vaikka ihminenkin on toiminnassaan erehtyväinen (Benetos ym. 2013). Jacques ja Roebel (2018) kuvailevat transkription ongelman jokseenkin ratkaistuksi monofonisille signaaleille, mutta haasteita polyfonisten signaalien transkriptiossa riittää runsaasti. Päällekkäiset signaalit ja nuotit tekevät transkriptiosta yhä monimutkaisemman lisääntyneiden äänilähteiden takia.

3.1.2 Menetelmät

Automaattisen rumputranskription ongelmien ratkaisumenetelmät on usein jaoteltu korkealla tasolla kahteen sukuun: segmentointiin ja luokitteluun (engl. *Segment and Classify*) sekä

erotteluun ja havaitsemiseen (engl. *Separate and Detect*) (Paulus ja Klapuri 2009; Jacques ja Roebel 2018, esimerkkeinä). Ensin mainitussa menetelmässä äänisignaali jaetaan segmentteihin ja yritetään muodostaa toistettava käsitys segmentin sisällöstä. Jälkimmäisessä menetelmässä erotellaan instrumentit toisistaan eri kanaviin, joista pyritään havaitsemaan iskutapahtumia (Jacques ja Roebel 2018). Tässä tutkielmassa käytetty menetelmä perustuu osittain erottelun ja havaitsemisen menetelmään, jossa pyritään havaitsemaan yksittäiset rumpuinstrumentit polyfonisen musiikin seasta.

Ensimmäiset automaattisen rumputranskription suunnittelumallit jaettiin karkeasti äänitapahtumien hahmontunnistukseen ja erotteluun perustuviin järjestelmiin (FitzGerald ja Paulus 2006). Näistä ensimmäinen segmentoi signaalin tarkoituksenmukaisiin tapahtumiin ja tunnistaa näiden sisällöt hahmontunnistuksen menetelmillä (engl. *Pattern Recognition Approaches*). Tapahtumiin perustuvat hahmontunnistuksen menetelmät voidaan jakaa karkeasti neljään vaiheeseen. Ensimmäinen on signaalin segmentointia joko löytämällä potentiaaliset iskutapahtumat tai generoimalla säännöllinen temporaalinen verkko signaalin päälle. Segmentoinnin jälkeen johdetaan toiminnot ja piirteet kustakin segmentistä, jonka sisällöt voidaan luokitella johdettujen piirteiden perusteella. Lopuksi yhdistetään segmenttien aikaleimat niihin liittyvien sisältötietojen kanssa, joiden avulla voidaan tuottaa transkriptio (FitzGerald ja Paulus 2006). Erotteluun perustuvissa järjestelmissä (engl. *Separation-Based Approaches*) äänivirrat pyritään erottelemaan kunkin rumpuinstrumentin mukaan, joista voidaan etsiä iskutapahtumat (FitzGerald ja Paulus 2006). Erottelujärjestelmien moniraitaäänitteissä eri rumpusetin osat vuotavat ei-toivottua kohinaa kuhunkin sensoriin (mikrofoniin), mikä heikentää yksinkertaisimpien menetelmien, kuten ICA:n (engl. *Independent Component Analysis*) suorituskykyä (Paulus 2009; FitzGerald ja Paulus 2006). Esimerkiksi Gillet ja Richard (2008) käyttivät ICA:ta taajuuskaistasuodatetun stereosignaalin prosessoinnissa rumpujen erottelemisessa polyfonisesta musiikista.

Myöhemmin Paulus (2009) ja Gillet ja Richard (2008) ovat esittäneet automaattisen transkription suunnittelumalleille täsmällisemmän jaottelun, joka koostuu neljästä kategoriasta. Suunnittelumallit jaoteltiin segmentointiin ja luokitteluun (engl. *Segment and Classify Approach*), erotteluun ja havaitsemiseen (engl. *Separate and Detect Approach*), sopeutuvien mallien täsmäämiseen (engl. *Match and Adapt Approach*) sekä Markovin piilomalleihin pe-

rustuviin tunnistusmenetelmiin (engl. *HMM-based Recognition Approach*).

Lisääntyneen rumputranskriptioon liittyvän tutkimuksen myötä Wu ym. (2018) esittivät tuoreimpana jaotteluna 6-osaisen suunnittelumallijoukon automaattisen transkription suorittamiseen. Joukko koostuu toimintojen esitystavoista (engl. *Feature Representation*), tapahtumien segmentoinnista (engl. *Event Segmentation*), aktivaatiofunktiosta (engl. *Activation Function*), toimintojen muuntamisesta (engl. *Feature Transformation*), tapahtumien luokittelusta (engl. *Event Classification*) ja kielimallista (engl. *Language Model*). Suunnittelumalleista kutakin voi käyttää työkalumaisesti toisistaan riippumatta halutussa järjestyksessä.

Signaalista halutaan käyttöön oleellisin mahdollinen informaatio, jolloin sen esitystapaa on tarpeen muuntaa esimerkiksi STFT:llä taajuusalueeseen. Wu ym. (2018) kuvailevat toimintojen olevan paremmin käytettävissä myöhäisemmässä prosessoinnissa, kun äänitteen rumpuja on saatu prosessoitua mielekkäämpään muotoon esimerkiksi taajuuskaistasuotimilla, jolloin tietyt rumpuinstrumentit ovat korostettuja.

Tapahtumien segmentoinnissa signaalista pyritään havaitsemaan iskutapahtumien temporaaliset sijainnit ennen jatkoprosessointia. Segmentointia tehdään lasketun spektrivuofunktion avulla, jolla pystytään havaitsemaan äkilliset muutokset signaalissa, kuten iskutapahtumien alkupisteet. Spektrivuo ilmentää spektrin kirjon voimakkuuden vaihtelua vertailemalla spektrin voimakkuutta suhteessa edelliseen ikkunaan (Ganguly ja Sharma 2017; Lerch 2012). Iskutapahtumien havaitsemisessa ainoastaan positiivisia voimakkuuden muutoksia on mielekästä tarkastella, mikä kertoo syttyvästä nuotista tai rumpuiskusta. Tässä tutkielmassa spektrivuota on käytetty STFT-ikkunoille signaalin huippujen analysoinnissa, ja sen matemaattinen kaava on

$$v_{\mathrm{SF}}(n) = \frac{\sqrt{\sum_{k=0}^{\kappa/2-1}(|X(k,n)| - |X(k,n-1)|)^2}}{\kappa/2},$$

jossa $X(k,n)$ on yhden ikkunan STFT, ja $(|X(k,n)| - |X(k,n-1)|)^2$ peräkkäisten ikkunoiden kompleksilukujen amplitudien erotus. Lasketun spektrivuon tuloksena on arvo välillä $0 \leq v_{\mathrm{SF}}(n) \leq A$, jossa A on suurin mahdollinen spektrin voimakkuus. Maksimiarvo määräytyy signaalille tehtyjen esikäsittelytoimenpiteiden ja taajuusmuunnosten perusteella (Lerch

2012).

Aktivaatiofunktiota muodostaessa pyritään kuvaamaan aiemmin saadut STFT-esitystavan toiminnot aktivaatiofunktioina, jotka ilmentävät eri rumpuiskujen aktivointitasoja. Aktivaatiofunktioita voidaan johtaa mm. NMF- (engl. *Non-Negative Matrix Factorization*), PLCA- (engl. *Probabilistic Latent Component Analysis*) sekä DNN-menetelmillä (engl. *Deep Neural Networks*) (Wu ym. 2018). Tässä tutkielmassa käytettiin konvoluutioneuroverkkoja aktivaatiofunktioiden johtamiseen. Toimintojen muuntaminen osana kyseistä suunnittelumallia on vastaavanlainen vaihe aktivaatiofunktion muodostamisen kanssa. Tällöin toimintojen esitystapaa pyritään typistämään esimerkiksi PCA:n (engl. *Principle Component Analysis*) avulla (Wu ym. 2018).

Tapahtumien luokittelussa pyritään assosioimaan ajan suhteen havaitut musikaaliset tapahtumat tietyn rumpuinstrumentin kanssa (Wu ym. 2018). Tässä tutkielmassa keskitytään havaitsemaan ainostaan bassorumpua ja virveliä. Gillet ja Richard (2008) kuvailevat haasteita löytyvän piirteiden poiminnassa, jossa ilman sävelkorkeutta olevia rumpuinstrumentteja pitäisi pystyä erottelemaan toisistaan signaalista. Rumpuinstrumenttien tunnistamista varten piirteitä voidaan yrittää muodostaa temporaalisten piirteiden, energian jakautuneisuuden, spektrin piirteiden, Fourier-muunnoksesta johdettujen piirteiden (engl. *cepstral features*) sekä aistillisten piirteiden (engl. *perceptual features*) avulla. Signaalin energiasta voidaan hyödyntää sen kokonaisenergiaa, rumpuinstrumenttispesifiä suodatusenergiaa tai suodatettuja oktaavitaajuusalueita. Aistillisia piirteitä voivat olla signaalista ilmenevä suhteellinen äänekkyys, terävyys ja hajonta. Esimerkiksi symbaalit hajoavat useille taajuusalueille (Wu ym. 2018).

Kielimallia käytetään korkeamman tason MIR-ongelmien ratkaisemisessa, jossa peräkkäisistä tapahtumista pyritään todennäköisyysmallien avulla ilmentämään jotain ilmiötä, kuten musiikkigenreä, tempoa tai rytmiä. Useimmiten kuitenkin vain osaa näistä kuudesta suunnittelumallista on tarpeen käyttää automaattisessa rumputranskriptiossa. (Wu ym. 2018)

Viimeisin kattava katsaus automaattisen rumputranskription tutkimuskenttään on tehty vuonna 2018 (Wu ym. 2018). Tässä tutkielmassa iskutapahtumien havaitseminen keskittyy DTM-rumputranskriptiotehtäviin, jossa DTM on rumputranskriptio muiden melodisten soittimien ollessa läsnä (engl. *Drum Transcription in the presence of Melodic Instruments*). DTD on

pelkän rumpuraidan transkriptio (engl. *Drum Transcription of Drum-only Recordings*), jolle saadaan yleisesti ottaen paljon parempia tuloksia DTM:ään verrattuna (Vogl, Dorfer ja Knees 2017; Wu ym. 2018, esimerkkeinä).

3.1.3 Rumputranskriptiotehtävät

Seuraavat kategorisoidut rumputranskriptiotehtävät (engl. *Drum transcription tasks*) muodostavat automaattisen rumputranskription kokonaisuuden (Wu ym. 2018).

- Rumpuäänien luokittelu (DSC, engl. *Drum Sound Classification*)
- Samankaltaisten rumpuäänien tunnistaminen (DSSS, engl. *Drum Sound Similarity Search*)
- Rumputekniikan luokittelu (DTC, engl. *Drum Technique Classification*)
- Pelkän rumpuäänityksen transkriptio (DTD, engl. *Drum Transcription of Drum-only Recordings*)
- Rumputranskriptio muiden perkussioiden läsnä ollessa (DTP, engl. *Drum Transcription in the Presence of Additional Percussion*)
- Rumputranskriptio muiden melodisten soittimien lomassa (DTM, engl. *Drum Transcription in the presence of Melodic Instruments*)

Wu ym. (2018) esittävät katsauksessaan, että DTD-tehtäville pystytään saamaan luotettavia tuloksia, mutta DTP- ja DTM-tehtävät kaipaavat vielä paljon kehittävää tutkimusta (Gillet ja Richard 2008).

3.1.4 Konvoluutioneuroverkot

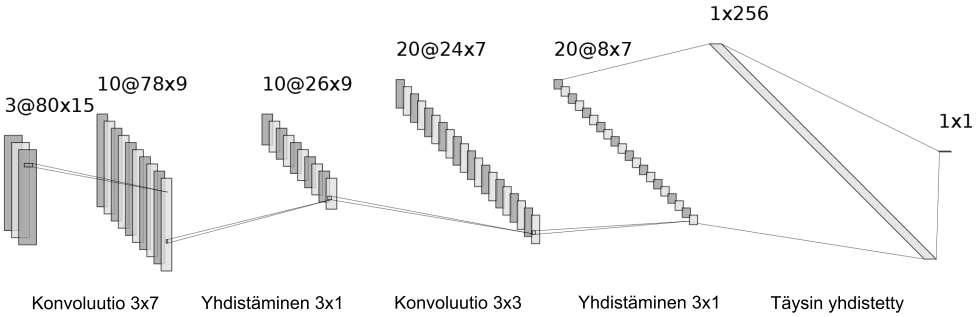
LeCun ja Bengio (1998) esittelivät kuvien luokitteluun ja puheen analysointiin soveltuvat konvoluutioneuroverkot, jotka saattavat yhteen kolme arkkitehtonista ideaa taatakseen jonkin asteisen siirron ja vääristymän muuttumattomuuden paikallisilla tulkinta-alueilla (engl. *local receptive field*) eli spektrogrammianalyysissä käytetyillä suodinten ko'oilla, jaetuilla painotuksilla (parametrien jakamisessa) sekä spatiaalisessa alinäytteistämisessä (engl. *spatial sub-sampling*). Kyseiset ideat ovat käytössä mm. reunojen tunnistamisessa spektrogrammidatasta. Näitä piirteitä hyödynnetään peräkkäisillä kerroksilla korkeamman tason tiedon johtamista varten (LeCun ja Bengio 1998).

Konvoluutioneuroverkkojen sanotaan olevan tällä hetkellä parhaimpia muihin menetelmiin verrattuna iskutapahtumien havaitsemisessa automaattisessa rumputranskriptiossa (Jacques ja Roebel 2018; Schlüter ja Böck 2014; Jacques ja Roebel 2019). Konvoluutioneuroverkot soveltuvat erityisen hyvin etenkin kuva- ja spektrogrammidatan hyödyntämiseen (Lerch 2012; Pons, Lidy ja Serra 2016). Schlüter ja Böck (2014) käyttivät suorakulmaisia suodatusmuotoja spektrogrammien analysointiin, jossa tärkeää oli havaita iskutapahtumia. Pons, Lidy ja Serra (2016) kuvailevat temporaalisten suodinten soveltuvan erityisesti iskutapahtumien havaitsemiseen. Taajuuden tarkastelussa suodatusikkuna oli konvoluutioneuroverkossa kapeampi sen sijaan, että oltaisiin käytetty neliön muotoista ikkunaa, jolloin aika ja taajuus olisivat yhtä merkitseviä komponentteja. Konvoluutioneuroverkot vaikuttavat lisäksi olevan yleisesti parempia aiemmin paljon käytettyihin NMD-menetelmiin (engl. *Non-negative Matrix Deconvolution*) verrattuna ja etenkin silloin, kun kutakin havaittavaa rumpuinstrumenttia varten opetetaan oma konvoluutioneuroverkko. Toisaalta tällöin riski ylisovittamiselle kasvaa, mikä voi tapahtua esimerkiksi tiettyjen bassorumpuäänitteiden tapauksessa. (Jacques ja Roebel 2018).

Pons, Lidy ja Serra (2016) kuvailevat konvoluutioneuroverkkojen sopivan paremmin paikallisten komponenttien, kuten instrumentin sointivärin tai musikaalisten yksikköjen mallintamiseen. Takaisinkytkeytyvät neuroverkot (engl. *Recurrent Neural Network (RNN)*) puolestaan soveltuvat paremmin pitkäaikaisten riippuvuuksien, kuten musiikin rakenteen tai toistuvien harmonioiden mallintamiseen. Konvoluutioneuroverkko oli osuva valinta tämän tutkielman menetelmäksi, sillä signaalinkäsittelyä tehdään paikallisesti lyhyille STFT-muunnetuille aikaikkunoille.

Konvoluutioneuroverkkoarkkitehtuurit voivat olla keskenään hyvin samankaltaisia riippuen siitä, ovatko ne tarkoitettu käytettäväksi yleispätevään iskutapahtumien havaitsemiseen vai täsmällisempään tehtävään, kuten pianoiskujen havaitsemiseen (Jacques ja Roebel 2018). Eroava tekijä voi olla datan rakenne, kuten Wangin ja Schlüterin tutkimuksissa käytettyjen spektrogrammien välillä. Wang ja Yan (2017) käyttivät tutkimuksessaan vain yhtä Q-vakiomuunnettua spektrogrammia suurella taajuuskaistalukumäärällä, kun Schlüter ja Böck (2014) ja Jacques ja Roebel (2018) puolestaan käyttivät kuvankäsittelymenetelmille omainaista kolmen kanavan (RGB, *"Red", "Green"and "Blue"*) lähestymistapaa. RGB-kanavien

sijaan käytettiin kolmesta eri STFT-esityksestä laskettuja mel-taajuuskaistaspektrogrammeja, joissa kussakin käytettiin yhtä monta taajuuskaistaa. Tässä tutkielmassa kuvion 10 mukaista neuroverkkoarkkitehtuuria on lähdetty jäljentämään näiden hyväksi todettujen menetelmien kautta, jossa käytetyt STFT-ikkunakoot olivat 1024 (23 ms), 2048 (46 ms) ja 4096 (96 ms), ja mel-taajuuskaistojen lukumäärä 80. Käytetty harppaus oli 441 (10 ms) näytteenottotaajuuden ollessa 44100 Hz. Tässä tutkielmassa kehitettyä toteutusta on kuvattu luvussa 5.



Kuvio 7: Tutkielmassakin pohjana käytetty konvoluutioneuroverkkoarkkitehtuuri, jota käyttivät Jacques ja Roebel (2018). Täysin yhdistetyn kerroksen 256 yksikköä ovat aktivointiin käytettyjä ReLU-yksiköitä, joista saadaan johdettua lopullinen havaittu rumpuinstrumentti.

Schlüter ja Böck (2014) havaitsivat konvoluutiokerrosten konvoluutiomaskien painotusten (engl. *weight*) olevan erilaisia harmonisten ja perkussiivisten iskutapahtumien havaitsemisessa, jolloin tiettyjen rumpuinstrumenttien luokittelu on mahdollista konvoluutioneuroverkoilla. Harmonisten ja perkussiivisten lähteiden erottelun merkitystä iskutapahtumien havaitsemisen kannalta on kuvattu tarkemmin luvussa 3.2.3.

3.2 Iskutapahtumien havaitseminen

Iskutapahtumien havaitseminen (engl. *onset detection*) on matalan tason tehtävä ja ensimmäinen tarpeellinen vaihe automaattisessa rumputranskriptiossa (Schlüter ja Böck 2014; Vogl ym. 2017; Roebel, Jacques ja Aknin 2018). Benetos ym. (2013) kuvailevat iskutapahtumia äkillisiksi energian muutoksiksi signaalissa, harmonisen sisällön muutoksena tai etenkin perkussiivisten instrumenttien tapauksessa ennustamattomana tapahtumakomponenttina, jota seuraa vakaa alue. Vakaa alue aiheutuu perkussiivisten instrumenttien lyhyestä soinnista,

jossa nuotti tai isku päättyy suhteellisen nopeasti verrattuna harmoniseen sisältöön. Onnistunut iskutapahtumien havaitseminen vähentää merkittävästi transkriptioalgoritmin käyttämää prosessointiaikaa, koska algoritmia ei silloin tarvitse suorittaa koko signaalille (Jacques ja Roebel 2018). Iskutapahtumien havaitsemista kuvaillaan ensimmäiseksi vaiheeksi myös tarkoituksena ymmärtää musiikissa piileviä jaksollisia piirteitä ja aksentteja, jotka määrittävät rytmin (Benetos ym. 2013, 8).

3.2.1 Haasteet

Iskutapahtumien havaitsemisen yksi suurimmista haasteista automaattisessa rumputranskriptiossa on tarve hyvin tarkoille iskutapahtuma-annotaatioille (Jacques ja Roebel 2019; Schlüter ja Böck 2014, esimerkkeinä). Toleranssi-ikkuna (engl. *tolerance window*) kertoo kuinka harvakseltaan musiikkiotteen iskutapahtumien sallitaan sijaita, jotta ne voidaan havaita omiksi tapahtumikseen (esim. 16- ja 32-osa nuottien erottelu) (Wu ym. 2018). Varsinkin metallimusiikille ominaista on tiheät iskutapahtumat ja nopea tempo (Williams 2014). Litovsky ym. (1999) kuvailevat ihmisen pystyvän erottamaan vähintään 8-10 ms väliset yksinkertaiset tapahtumat omiksi iskuikseen, minkä takia vähintään 20 ms toleranssi iskutapahtuman merkitsemisessä oikeaksi on hyväksyttävä arvo. Bello ym. (2005) kuvailevat hyödylliseksi käyttää eri aikaresoluutioita eri taajuuskaistoille. Aikaresoluution tulee kuitenkin olla tarpeeksi tarkka, mikä taas epämääräisyysperiaatteen mukaan tarkoittaa heikompaa taajuusresoluutiota. Transkriptiotutkimuksissa käytetyt toleranssi-ikkunat voivat olla kooltaan jopa 50 ms (Paulus 2009), mutta suurin osa löytämieni lähteiden perusteella vaikuttivat olevan 20-30 ms välillä, (Dittmar ja Gärtner 2014; Vogl, Dorfer ja Knees 2017; Vogl ym. 2017; Jacques ja Roebel 2018; Schlüter ja Böck 2014, esimerkkeinä).

Gillet ja Richard (2008) kuvailevat joidenkin instrumenttien iskutapahtumien tunnistamisen olevan hyvin haasteellista niiden herkkyyden takia muiden instrumenttien läsnä ollessa. Mikäli esimerkiksi bassorummun kanssa soi samaan aikaan jokin korkeataajuuksinen instrumentti, tieto matalista taajuuksista saattaa hukkua, sillä kyseisen ajanhetken spektri on keskittynyt korkeisiin taajuuksiin (engl. *spectral centroid*). Jacques ja Roebel (2018) ovat ratkaisseet heikkojen bassorumpuiskujen havaitsemisen ongelmaa normalisoimalla signaalidataa ainoastaan aika-alueen suhteen. Normalisointi korostaa äkillisiä signaalienergian

muutoksia, jotka voidaan mahdollisesti luokitella iskutapahtumiksi. Tämänkaltainen normalisointi kuitenkin muuttaa taajuusalueiden välisiä energiasuhteita, mutta bassorummun matalien taajuusalueiden tapauksessa iskutapahtumat ovat havaittavissa paremmin normalisoinnin jälkeen.

3.2.2 Menetelmät

Useimmiten transkriptiossa signaalin esikäsittelyvaiheen päätteeksi on saatu riittävästi tietoa transkriptiota hyödyttävistä piirteistä, joiden avulla voidaan suorittaa iskutapahtumien havaitsemista. Tällöin lasketaan iskutapahtumafunktio (ODF, engl. *Onset Detection Function*)[1], jonka paikallisten maksimiarvojen katsotaan olevan iskutapahtumia, mikäli arvo ylittää algoritmille annetun kynnysparametrin (Jacques ja Roebel 2018). Funktio määrittää signaalin piirteiden muutokset ikkunakohtaisesti (engl. *frame*). Iskutapahtumafunktiot lasketaan yleensä signaalien taajuusalueesta käyttäen taajuuskaistakohtaista voimakkuutta (engl. *magnitude*) tai vaihetta, joiden avulla saadaan laskettua spektrivuo (engl. *spectral flux*), vaihehajonta (engl. *phase deviation*) sekä kompleksialueen havaitsemisfunktioita (Benetos ym. 2013; Bello ym. 2005).

FitzGerald (2004) esittää väitöskirjassaan iskutapahtumien havaitsemiseen yleisesti käytetyn rakenteen, jossa sisääntuleva signaali prosessoidaan ensin taajuuskaistasuodatuksen läpi (engl. *Filterbank Processing*). Kunkin vahvistetun suodatuksen tulosteet konvolvoidaan 100 ms kokoisella puoli-Hanning-ikkunalla (engl. *half-Hanning*), minkä tuloksena saadaan laskettua verhokäyrät (engl. *Amplitude envelope*). Verhokäyrillä kuvataan amplitudin muutoksia ajan suhteen, mikä auttaa havaitsemaan nopeita muutoksia signaalissa (Lerch 2012). Schlüter ja Böck (2014) puolestaan käyttivät sileäkäyräistä Hamming-ikkunointia kolmella eri ikkunakoolla (23 ms, 46 ms ja 93 ms), mikä on konvoluutioneuroverkon opetuksessa tarkempi menetelmä kustakin ikkunakoosta saadun aika- ja taajuustietojen kokoamisen jälkeen. Suodatus, vahvistus ja ikkunointi vastaa prosessia, jonka ihmisen korva suorittaa vastaanotetuille ääniaalloille [2] (FitzGerald 2004). Verhokäyrät syötetään prosessissa iskuta-

1. Toiselta nimeltään *novelty function*, joka viittaa johonkin uuteen tapahtumaan audiosignaalissa (Lerch 2012).

2. Ns. spektrin analysointia basilaarimembraanin avulla (Smith 1997).

pahtumakomponenttien havaitsemisvaiheeseen, jossa tarkastellaan amplitudin muutosta signaalin tasoon nähden. Parametrina annetun kynnysarvon ylittävät iskutapahtumat merkataan talteen. Schlüter ja Böck (2014) katsovat iskutapahtumiksi kaikki tapahtumat, joiden paikalliset maksimiarvot ylittävät annetun kynnysarvon. Eri ikkunoiden koot siten opettavat neuroverkkoa hieman vaihtelevilla paikallisilla maksimeilla. Iskutapahtumakomponenttien intensiteettiä arvioidaan taajuuskaistasuodinten keskitaajuuden perusteella, ja mikäli komponentti sijaitsee korkeintaan 50 ms päässä intensiteetiltään vahvemmasta komponentista, se jätetään huomioimatta (FitzGerald 2004). Tämän jälkeen eri taajuuskaistojen iskutapahtumakomponentit yhdistetään, jolloin ne muodostavat koko signaalista saadut iskutapahtumat. Opetetun neuroverkon arvioinnissa Schlüter ja Böck (2014) huomioivat iskutapahtumat vain, mikäli ne ovat 25 ms sisällä merkatusta iskutapahtumakohteesta.

Iskutapahtumien havaitsemisessa osa tutkimuksesta keskittyy huippukohtien löytämiseen iskutapahtumafunktiosta, ja osa neuroverkkojen käyttöön kyseisen funktion muodostamisessa (Jacques ja Roebel 2018, 80-81). Schlüter ja Böck (2014) käyttivät iskutapahtumien havaitsemisessa konvoluutioneuroverkkoja, jotka mielletään vuosien 2017 ja 2018 MIREX-evaluointien[3] tulosten mukaan parhaimmaksi menetelmäksi tänä päivänä. Vuonna 2018 lähes yhtä tarkkoihin tuloksiin pääsi Roebel, Jacques ja Aknin (2018) kehittämällään myös konvoluutioneuroverkkoja hyödyntävällä keinotekoisesti jatkettujen aineistojen menetelmällä.

3.2.3 Harmonisten ja perkussiivisten lähteiden erottelu

Tässä luvussa syvennytään perkussiivisten lähteiden erottelemiseen harmonisista lähteistä, eli rumpujen erottelemiseen polyfonisesta musiikista. Jacques ja Roebel (2018) kuvailevat musiikin koostuvan karkeasti harmonisista ja perkussiivisista instrumenteista. Gillet ja Richard (2008) näkivät rumputranskriptiojärjestelmien hyötyvän lähteiden erottelusta, sillä harmonisten instrumenttien kontribuutio signaalissa halutaan poistaa tai vähintäänkin minimoida. On siis perusteltua erotella nämä lähteet niiden erilaisten piirteidensä vuoksi.

Aiemmin esitellyistä rumputranskriptiotehtävistä tarkemman tarkastelun kohteeksi otetaan DTM, jota on tutkinut muun muassa Paulus (2009) ja Gillet ja Richard (2008). Harmonisia ja

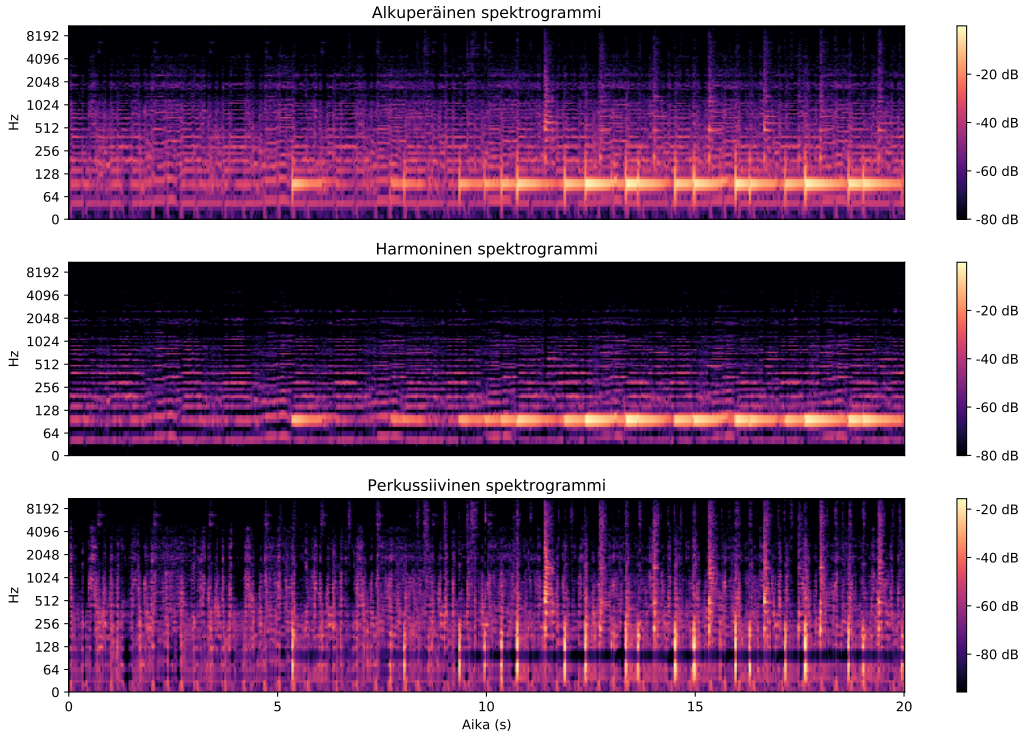
3. https://nema.lis.illinois.edu/nema_out/mirex2017/results/aod/summary.html

perkussiivisia lähteitä on usein perusteltua erotella signaalin esikäsittelyvaiheessa parempien tulosten saamiseksi iskutapahtumien havaitsemisessa, sillä sisääntuleva polyfoninen signaali sisältää usein eri instrumenttien tapahtumia samoilla taajuusalueilla (Elowsson ja Friberg 2013). Tässä tutkielmassa käytettyä lähteiden erottelua tehtiin käyttäen mediaanisuodatusta (engl. *median filtering*) ja reuna-arvoihin liittyvää lisätarkastelua (engl. *margin-based extension*) (Driedger, Müller ja Disch 2014). Erottelumenetelmien Python-toteutus on nähtävissä librosa-audioanalyysikirjastossa [4].

Hajotettujen (engl. *diffusion*) harmonisten ja perkussiivisten spektrogrammien tarkastelussa keskeisintä on erot niiden rakenteissa: harmonisissa spektrogrammeissa sisääntuleva signaali on muodoltaan horisontaalinen, ja perkussiivisissa spektrogrammeissa sisääntuleva signaali muodostaa vertikaalisia rakenteita (Ono ym. 2008) (ks. kuvio 8). Näitä hajotuksesta saatuja paranneltuja esitystapoja vertaillaan alkuperäisen spektriesityksen (engl. *spectral representation*) kanssa, jonka alkiot kartoitetaan joko harmoniseen tai perkussiiviseen paranneltuun spektrogrammiin osuvuuden perusteella (Driedger, Müller ja Disch 2014). Tällä tavalla tehtyä iteratiivista spektrogrammien hajotusta edelsi yksinkertaisempi mediaanisuodatusmenetelmä, joka oli laskentateholtaan tehokkaampi (Fitzgerald 2010). Mediaanisuodatusta käytetään erikseen horisontaalisille ja vertikaalisille suunnille lähteiden erottelussa. Kuitenkin pelkällä mediaanisuodatuksella tehty lähteiden erottelu ei tuota tarpeeksi tarkkoja spektrogrammeja, sillä hajontaa tapahtuu harmonisista lähteistä perkussiivisiin spektrogrammeihin ja päinvastoin. Tätä on paranneltu hajontaa tarkentavalla erottelukertoimella (engl. *separation factor*), jolla saadaan vähennettyä harmonisen taustan, kohinan ja esimerkiksi särökitaroiden vuotamista perkussiiviseen spektrogrammiin (Driedger, Müller ja Disch 2014).

Aikaisemmin Helén ja Virtanen (2005) ovat esittäneet artikkelissaan menetelmän rumpujen erottelemiseen polyfonisesta musiikista käyttäen NMF:ää sekä tukivektorikoneita (SVM). Kyseinen menetelmä on myös paljon käytetty, mutta tässä tutkielmassa se jää maininnan tasolle vertailukelpoisena vaihtoehtona, koska konvoluutioneuroverkoista on saatu lupaavia tuloksia (Jacques ja Roebel 2018; Schlüter ja Böck 2014; Wu ym. 2018).

4. https://librosa.github.io/librosa_gallery/auto_examples/plot_hprss.html



Kuvio 8: Kuviossa ilmennetään STFT-muunnetusta polyfonista metallimusiikkia sisältävästä rumpusignaalista erotellut harmoniset ja perkussiiviset lähteet. Spektrogrammin laskennassa käytetty ikkunakoko $\kappa = 2048$, harppaus $\mathscr{H} = 512$ ja erottelukerroin $margin = 2$. Harmonisessa spektrogrammissa on nähtävissä horisontaaliset rakenteet ja perkussiivisessa spektrogrammissa vertikaaliset rakenteet.

3.2.4 Metallimusiikki

Yksi metallimusiikin määrittävistä piirteistä on siinä esiintyvät säröefektit (engl. *distortion*), joita varten (kitara)signaalia on käsitelty terävöittämällä ääniaaltojen huippuja (Herbst 2017). Tyypillisissä metallimusiikkiotteissa äänten dynaaminen vaihteluväli on usein kompressoitu kattamaan useat taajuusalueet korkealla volyymitasolla, mikä puolestaan sallii signaalin voimakkuuden vahvistamisen (engl. *gain*) miksatussa lopputuotoksessa (Williams 2014). Jo yksinään tämän takia rumpujen transkriptio ja erottelu harmonisen metallimusiikin seasta on haastavaa, sillä säröstä aiheutuvaa kohinaa ilmenee laajoilla taajuusalueilla.

4 Data

Tutkimuksessa käytetty data koostuu vertailukelpoisesta ENST-rumputietokannasta ja metallimusiikkiin erikoistuneista *Riddle Me This* -yhtyeen studioäänityksistä.

4.1 ENST-rumputietokanta

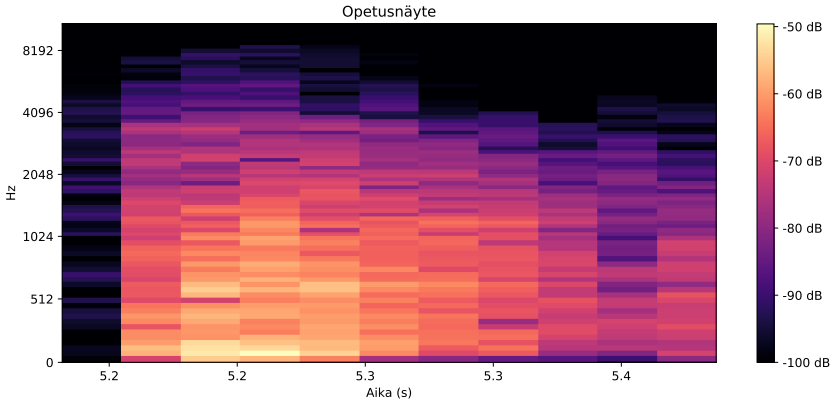
Tämän tutkielman menetelmän kehitykseen käytetty aineisto on saatu julkisesta tutkimuskäyttöön tarkoitetusta ENST-rumputietokannasta (engl. *ENST-Drums: an extensive audio-visual database for drum signals processing*) (jäljempänä ENST). ENST koostuu kolmen ammattilaisrumpalin soittamista äänitteistä, jotka on muodostettu 8-kanavaisista moniraitaäänitteistä ja kahdesta videokanavasta (Gillet ja Richard 2006). Tässä tutkielmassa hyödynnetään ainoastaan moniraitaäänitteitä. ENST-tietokantaa on käytetty laajasti useissa tutkimuksissa (Jacques ja Roebel 2018; Wu ym. 2018; Southall, Stables ja Hockman 2017, esimerkkeinä), minkä takia se on tässäkin tutkimuksessa erittäin vertailukelpoinen aineisto. ENST on luotu tarkoituksena saattaa monipuolinen rumpuaineisto julkiseen tutkimuskäyttöön.

Kokonaisuudessaan ENST sisältää noin 75 minuuttia äänitettyä materiaalia kultakin kolmelta rumpalilta. ENST:n ääniraidat ovat kokonaisvaltaisesti merkattu totuusarvoisilla aikaleimatuilla rumpuiskutapahtumilla, jotka kattavat bassorummun, virvelin ja hi-hatin lisäksi useita symbaaleja sekä tomeja. Kukin kolmesta rumpalista soitti omilla rumpuseteillään, erilaisilla lyömävälineillä sekä omalla tyylillään ilman kirjoitettuja nuotteja, mikä tuki mahdollisimman luonnollisen ja monipuolisen datan keräämistä. Tietokannan annotaatiot verifioitiin useamman henkilön toimesta, jotta saatiin mahdollisimman luotettavat aikaleimat kullekin iskutapahtumalle.

Tässä tutkielmassa ENST-tietokannasta käytettiin ääniraitoja, joissa oli polyfonista musiikkia äänitetyn rumpujen soiton seassa. Nämä raidat on merkattu tietokannassa tunnisteella *'minus-one'*, joka tarkoittaa niiden sisältävän polyfonista musiikkia.

Tietyn rummun iskutapahtumia tunnistavan neuroverkon opetusta varten ääniraidoista muo-

dostettiin mel-spektrogrammeja, jotka pilkottiin 12 aikaikkunan pituisiin lohkoihin. Jacques ja Roebel (2018) käyttivät tutkimuksessaan 15 aikaikkunan pituisia lohkoja, mutta tätä tutkielmaa varten päädyin käyttämään hieman pienempiä lohkoja, jotta iskutapahtumat olisivat selkeämmin havaittavissa peräkkäisistä pilkotuista spektrogrammilohkoista. Iskutapahtuman pituudella tai päättymisajankohdalla ei ole painoarvoa perkussiivisissa iskutapahtumissa. Yhden datanäytteen koko pystyakselilla on 80 mel-taajuuskaistaa logaritmisesti skaalattuna, ja vaaka-akselilla 12 aikaikkunaa. Kuviossa 9 ilmennetään yhtä neuroverkon opetuksessa käytettyä iskutapahtumanäytettä.



Kuvio 9: Neuroverkon opetuksessa käytetty iskutapahtumanäyte virvelin iskusta.

Opetus- ja testiaineisto jaettiin kappaleiden mukaan sekoitetun datajoukon kesken siten, että testiaineiston kooksi otettiin 10 % koko aineistosta. Opetus- ja testiaineistoon voi mahdollisesti jäädä saman kappaleen sisällä olevia aikaikkunoita, mutta samoja näytteitä ei esiinny opetus- ja testiaineistoissa. Opetusvaiheessa käytettäväksi validointiaineistoksi valikoitiin 20 % opetusaineiston loppupäästä.

4.2 Studioäänitykset

Tutkielmassa hyödynnettiin myös itseäänitettyä rumpuaineistoa; metallimusiikkiaineistoa käytettiin menetelmässä, jonka toimivuus validoitiin ENST-aineistolla. Vietin yhteensä neljä päivää Keravalla Rami Nykäsen studiossa, jossa äänitin rumpuja *Riddle Me This* -yhtyeen esikoisalbumia varten. Äänitettyä soittoa sessioista saatiin 11 kappaleen verran, joka on ajallisesti noin 48 minuuttia. Nykäseltä saatiin neljä moniraitaäänitettä tutkimuksen varsinais-

ta metallimusiikkiaineistoa varten. Kustakin kappaleesta käytettävissä olevia raitoja olivat bassorumpu, virveli, koko rumpusetti sekä polyfoninen taustaraita. Taustaraita sisälsi särökitaroita, bassoa, laulua sekä muita orkestraalisia ja elektronisia instrumentteja. Yksittäisiä rumpuinstrumentteja käytettiin iskutapahtumien aikaleimojen löytämiseen, mikä on yksiraitaäänitteiden tapauksessa jo olemassa oleviin äänenkäsittelykirjastoihin toteutettu, helposti ratkaistava ongelma (ks. luku 3.1.1). Äänitettyä datajoukkoa käsiteltiin Python-skripteillä ja Madmom-kirjastolla, jolloin datajoukko saatiin jatkettua tutkittavaan muotoon ja merkattua iskutapahtumat automaattisesti konvoluutioneuroverkkoa käyttävän iskutapahtumahavaitsimen [1] avulla. Rumpuraidan päälle lisättiin polyfonista taustamusiikkia, jossa äänenpainetaso painottui rumpuraitaan taustaraidan ollessa hiljemmalla.

1. https://github.com/CPJKU/madmom/blob/master/bin/CNNOnsetDetector

5 Toteutus

Tässä luvussa esitellään tutkimuksessa kehitettyä menetelmää iskutapahtumien havaitsemiseen. Rumpuiskutapahtumia havaittiin konvoluutioneuroverkolla, jonka lähtökohtana oli tuoreimpia käytettyjä automaattisen rumpu- ja musiikkitranskription menetelmiä, (Jacques ja Roebel 2018; Schlüter ja Böck 2014; Wu ym. 2018, esimerkkeinä). Toteutettu iskutapahtumien havaitseminen ja neuroverkon opetus koostuvat datan sisäänlukemisesta, monivaiheisesta esikäsittelystä, konvoluutioneuroverkkomallin muodostamisesta, neuroverkon opetusvaiheesta ja evaluoinnista sekä tulosten raportoinnista.

5.1 Datan esikäsittely

Käsiteltävä data koostuu yksikanavaisista wav-ääniraidoista ja niitä vastaavista aikaleimatuista iskutapahtumatunnisteista. Ääniraitojen näytteenottotaajuus on 44100 Hz. Ääniraidan signaali saatetaan visuaaliseen muotoon laskemalla STFT audiosignaalille, josta muodostuu ensimmäinen spektrogrammi ja kutakin ikkunaa vastaavat aikaleimat. STFT lasketaan käyttämällä 2048 näytteen kokoista ikkunaa, 441 näytteen kokoista eli ajallisesti 10 ms mittaista harppausta ja Hann-ikkunafunktiota. STFT-spektrogrammi hajotetaan harmoniseen ja perkussiiviseen spektrogrammiin, joista talteen otetaan pelkkä perkussiivinen spektrogrammi. Hajotuksessa käytetty erottelukerroin on leveä (1.0, 5.0), mikä antaa paremmin eristetyn perkussiivisen spektrogrammin. Perkussiivinen spektrogrammi skaalataan mel-skaalaan 80 mel-taajuuskaistalla. Mel-spektrogrammin muodostaminen *librosa*-kirjastolla vaatii, että sisääntuleva spektrogrammi muutetaan kompleksitasosta reaaliluvuiksi (McFee ym. 2015). Mel-spektrogrammi skaalataan lopuksi desibeleihin, joka logaritmisen skaalansa ansiosta havainnollistaa spektrogrammin arvoja hyvin.

Esikäsitelty spektrogrammi pilkotaan peräkkäisiin 12 aikaikkunan kokoisiin lohkoihin, jotka paritetaan niitä vastaavien aikaleimojen kanssa. Yksi pilkottu spektrogrammi vastaa yhtä neuroverkolle syötettävää näytettä, joka sisältää tunnisteen iskutapahtumasta tai sen puuttumisesta. Pilkottu spektrogrammi sisältää iskutapahtuman vain, jos sen aikaleima on 30 ms sisällä totuusarvoisesta aikaleimasta. Samaa 30 ms kynnysarvoa olivat käyttäneet myös

Jacques ja Roebel (2018). Schlüter ja Böck (2014) puolestaan käyttivät 25 ms kynnysarvoa yleisessä iskutapahtumien havaitsemisessa.

5.2 Konvoluutioneuroverkko

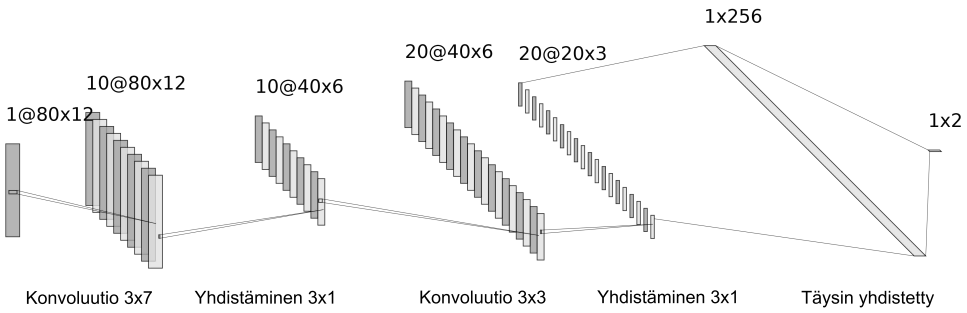
Tutkimuksessa käytetyn konvoluutioneuroverkon malli muodostettiin hyväksi todetun arkkitehtuurin perusteella, jota käyttivät myös Jacques ja Roebel (2018) ja Schlüter ja Böck (2014). Kyseistä mallia käyttämällä oltiin saatu parhaimpia tuloksia muihin menetelmiin verrattuna, joten se oli hyvä lähtökohta myös tälle tutkimukselle. Malli rakennettiin käyttäen TensorFlow-alustaa [1] ja Keras-kirjastoa [2].

Toisin kuin Jacques ja Roebel (2018) toteuttivat mallinsa sisääntulokerroksen, tätä tutkimusta varten sisääntulokerros muodostettiin kolmen kanavan sijaan vain yhdestä kanavasta käyttäen yhtä STFT-ikkunakokoa (2048). Sisääntulokerrokseen syötettävä data on muotoa (80, 12, 1), jossa 80 on mel-taajuuskaistojen lukumäärä ja 12 aikalohkojen lukumäärä. Kolmas arvo kuvaa kanavien lukumäärää eli ulottuvuutta. Toteutusta voisi tulevaisuudessa jatkaa tekemällä kolmikanavaisen sisääntulokerroksen.

Mallin piilotetut kerrokset koostuvat kahdesta konvoluutiokerroksesta, jotka on liitetty ReLU-aktivaatioihin ja maksimiarvoihin perustuviin yhdistämiskerroksiin (engl. *max-pooling*) sekä erien normalisointiin (engl. *batch normalization*). Konvoluutiokerrosten jälkeinen täysin kytketty kerros sisältää 256 ReLU-aktivaatioyksikköä, erien normalisoinnin ja pudotuksen (engl. *dropout*) 50 % todennäköisyydellä, mikä auttaa estämään ylisovittamista. Malli päättyy ulostulokerrokseen, jossa on käytetty sigmoid-aktivaatiofunktiota. Ulostulon arvona on joko iskutapahtuma tai ei iskutapahtumaa; neuroverkkoa opetetaan kerrallaan yhtä rumpuinstrumenttia varten. Kuvaajassa 10 ilmennetään tutkimuksessa käytetyn konvoluutioneuroverkon arkkitehtuuria.

Neuroverkon optimointia ohjaava tappiofunktio mittaa ennustetun ulostuloarvon ja kohdetunnisteen välistä divergenssiä. Tappiofunktiona käytettiin kategorista ristientropiaa (engl. *categorical cross-entropy*), joka soveltuu hyvin myös binääristen tunnisteiden kanssa (Jacques

1. https://www.tensorflow.org/
2. https://keras.io/



Kuvio 10: Tutkimuksessa käytetty konvoluutioneuroverkkoarkkitehtuuri, jossa ulostulokerroksen arvona on joko tunnistettu iskutapahtuma tai ei iskutapahtumaa.

ja Roebel 2018). Mikäli arveltiin näytteen sisältävän iskutapahtuman, kategorisoitiin näyte arvolla 1, ja ilman iskutapahtumaa olevat näytteet arvolla 0. Neuroverkon optimoijana käytettiin Adamia, jossa kokeillut oppimistahdit (engl. *learning rate*) olivat 0,0001 ja aikataulutettu hyperbolinen tangenttifunktio, jonka maksimioppimistahti oli 0,001. Pieni oppimistahti auttoi neuroverkkoa oppimaan tehokkaammin luokkien epätasapainoisuudesta aiheutuneesta haasteesta. Käytetyt eräkoot olivat 64, 256, 512 ja 1024, jotka valikoituivat optimaalisiksi kokeiltujen 32:n ja 1024:n väliltä. Validointiaineiston piti nähdä tarpeeksi näytteitä eräkohtaisesti, jotta voitiin ohjata neuroverkon opetusta oikeaan suuntaan.

Neuroverkolle asetettiin ehto ennenaikaiselle pysäytykselle. Kun validointitappio alkaa nousta suhteessa tappioon, on saavutettu teoreettinen ihannekohta mallin opetuksessa, jolloin neuroverkon opetuksen katsotaan olevan valmis. Opetuksessa pyrittiin tasapainottamaan epätasapainoisia luokkia asettamalla luokille painotukset niiden esiintyvyyden mukaan. *Sklearn*-kirjaston *compute_class_weight*-funktio [3] laskee painotukset opetusaineiston luokkien esiintyvyyksien mukaan.

Opettaminen tapahtuu käyttämällä grafiikkasuorittimia (GPU), jotka soveltuvat tutkielman kannalta hyvin konvoluutio-operaatioiden tehokkaaseen laskemiseen. Lopullisia tuloksia varten opetusajot suoritetaan Jyväskylän yliopiston GPU-palvelimella (*Supermicro 4028GR*), jossa käytettävissä on useita rinnakkaisia GPU:ita. Kyseinen palvelin sisältää kahdeksan *NVidia Tesla P100* GPU:ta, joissa kussakin on 16 Gt muistia, ja kaksi *Intel Xeon E5-2640 v4*

3. https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

prosessoria. Jyväskylän yliopiston yleinen laskentainfrastruktuuri perustettiin vuonna 2016 Informaatioteknologian tiedekunnan ja Matemaattis-luonnontieteellisen tiedekunnan toimesta.

5.3 Evaluointi

Neuroverkon opetuksen aikana mallia validoidaan validointidataa vasten, jonka kooksi valitaan 20 % opetusaineiston loppupäästä opetuksen alkaessa. Validointidataa käytetään opetettavan mallin evaluointiin, jonka perusteella päivitetään mallin oppimista. Malli näkee validointidataa kunkin epookin lopussa, mutta oppimisen sijaan validointidatasta lasketaan validointitappiota ja -tarkkuutta. Käytetty validointidata on sama joka epookilla. Validointidatan käyttäminen mm. ehkäisee ylisovittamista, sillä neuroverkon opetus voidaan pysäyttää, kun validointitappio alkaa kasvamaan suhteessa tappioon. Opetuksen jälkeen opetettua mallia arvioidaan ja todennetaan sen toimivuus testidatalla, jota neuroverkko ei ole aiemmin nähnyt. Testidatan osuus koko aineistosta on 10 %, joka sisältää vertailuun käytettävät totuusarvoiset tunnisteet.

Totuusarvoisia tunnisteita vertaillaan ennustettuihin tunnisteisiin, minkä perusteella voidaan laskea sekaannusmatriisi, täsmällisyys ja tunnistuskyky. Jacques ja Roebel (2018) kuvaavat täsmällisyyden (engl. *precision*) P kertovan, kuinka suuri osa positiivisista tunnistuksista osui todellisesti oikeaan. Se kuvaa luokittelijan kykyä olla merkkaamatta positiivisia luokkia eli iskutapahtumia negatiivisiksi. Tunnistuskyky (engl. *recall*) R kertoo, kuinka suuri osa todellisista positiivisista tunnistettiin oikein. Se kuvaa luokittelijan kykyä löytää kaikki positiiviset näytteet aineiston joukosta.

$$P = \frac{T_p}{T_p + F_p} \quad R = \frac{T_p}{T_p + F_n},$$

missä T_p on oikein luokiteltujen positiivisten havaintojen lukumäärä (engl. *true positive*), F_p on väärin luokiteltujen positiivisten havaintojen lukumäärä (engl. *false positive*) ja F_n on väärin luokiteltujen negatiivisten havaintojen lukumäärä (engl. *false negative*). F-arvo (engl.

F-measure) on kompromissi, jossa otetaan huomioon sekä täsmällisyys että tunnistuskyky:

$$F = \frac{2PR}{P+R}.$$

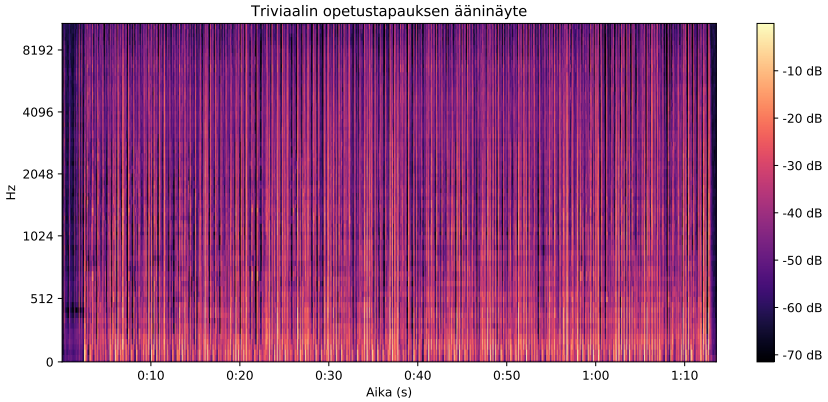
F-arvoa on mielekästä tarkastella, koska se ottaa huomioon kummatkin neuroverkkomallin evaluoinnista johdetut metriikat. Pelkästään täsmällisyyden tai tunnistuskyvyn tarkastelu tässä tutkimuksessa voisi tuottaa epäluotettavia tuloksia. Neuroverkon alkuarvojen satunnaisuuden vuoksi yksi ajo ei riitä takaamaan luotettavia metriikoita. Luotettavien tulosten saamiseksi opetusajoja suoritetaan peräkkäin *Multistart*-algoritmeille ominaisesti $N \cdot M$ kertaa, jolloin neuroverkko opetetaan erikseen kullekin rumpuinstrumentille 64, 256, 512 ja 1024 eräko'oilla. Ajetuista M kerrasta otetaan talteen paras tulos, joka valitaan pienimmän validointitappion perusteella. Parhaista N tuloksista lasketaan keskiarvot ja keskihajonnat metriikoille, jotka raportoidaan tutkielman tuloksiksi eräkokokohtaisesti.

5.4 Toimivuuden todentaminen triviaalilla datajoukolla

Toteutetun menetelmän ja evaluointikehyksen toimivuus varmistettiin ennen varsinaiseen opetukseen ryhtymistä triviaalilla datajoukolla, jonka tiedetään ennakkoon olevan menetelmän opittavissa helposti. Triviaalin datajoukon opetuksessa voitiin nähdä oppimisen ja validoinnin toimivan halutunlaisesti tarkasteltujen metriikoiden ja kuvaajien avulla. Yhdellä ääniraidalla tehdyn opetuksen tarkoituksena oli saada malli ylioppimaan aineisto, jolloin voitiin ilmentää haluttuja piirteitä tappiosta (engl. *loss*) ja tarkkuudesta (engl. *accuracy*). Tämä oli tarpeen todentaa, jotta voitiin varmistua lopullisten tulosten perustuvan valideihin päätelmiin. Datajoukkona käytettiin ENST-rumputietokannasta otettua *drummer_3_132_minus-one_charleston_sticks*-ääniraitaa sen tasaisuutensa vuoksi. Ääniraita on 73 sekunnin mittainen charleston-kappale, jossa bassorumpu ja virveli vuorottelevat selkeästi. Kuviossa 11 näkyy kappaleesta generoitu spektrogrammi.

Neuroverkon opetus lopetettiin asettamalla ehto ennenaikaiselle pysäytykselle. Kun validointitappio alkoi nousta suhteessa tappioon, oltiin saavutettu teoreettinen ihannekohta mallin opetuksessa. Kuvioista 12 selviää ennenaikaisen pysäytyksen toimivuus, kasvava validointitappio sekä laskeva validointitarkkuus triviaalilla datajoukolla.

Taulukossa 1 on kuvattu triviaalin opetustapauksen tuloksia bassorummun ja virvelin isku-



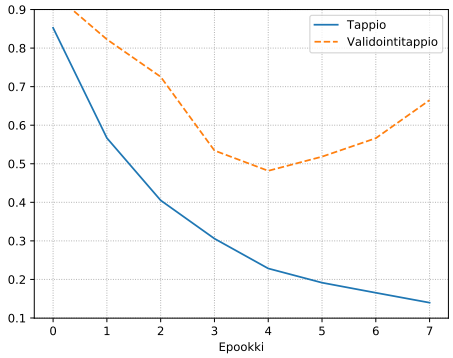
Kuvio 11: Logaritmiseen desibeliasteikkoon skaalattu mel-spektrogrammi triviaalissa opetuksessa käytetystä charleston-ääniraidasta.

tapahtumien havaitsemiselle. Kustakin instrumentista on raportoitu täsmällisyys, tunnistuskyky ja F-arvo. Metriikat laskettiin ottamalla keskiarvo ja keskihajonta viidestä parhaasta ajosta, jotka valittiin 15 ajon joukosta. Tämä toistettiin kullekin rumpuinstrumentille 64, 256 ja 512 eräko'oilla. Neuroverkkoja opetettiin yllä kuvatuilla asetuksilla ja hyperparametreillä 75 epookin ajan.

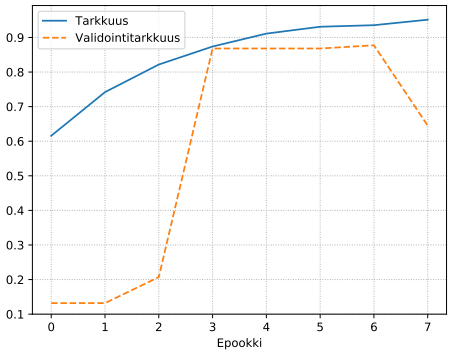
Virveli-iskutapahtumat ovat vaikeammin havaittavissa, sillä ne kattavat laajemman taajuusalueen bassorumpuun verrattuna. Ne sekoittuvat helpommin polyfoniseen musiikkiin, minkä takia harmonisten ja perkussiivisten lähteiden erottelun tärkeys korostuu datan esikäsittelyvaiheessa. Bassorummun iskutapahtumien havaitsemisen tarkkuutta ja tappiota on kuvattu kuviossa 13. Kuvaajissa olevaa mallia opetettiin 75 epookin ajan käyttäen 64 eräkokoa. Kun eräkoko oli 256, malli ei ylisovittunut yhtä nopeasti. Triviaalilla opetustapauksella pienempi eräkoko toimi paremmin verrattuna isompiin eräkokoihin. Kun datajoukko on laajempi, isompi eräkoko toimii paremmin, sillä opetusta ja validointia varten tarvitaan enemmän näytteitä kerralla (ks. taulukko 2). Kuvaajia 256 eräkoolla opetetusta mallista bassorummun iskutapahtumien havaitsemisessa on nähtävissä liitteessä 18.

Triviaalin datan tulokset ja kuvaajat olivat lähtöoletuksen mukaiset, mikä antoi luottoa siirtyä varsinaisten tulosten tuottamiseen realistisella aineistolla. Koodit Python-toteutuksesta ovat julkisesti saatavissa GitHubissa [4]. Kehityskohteita ja tulevaisuuden näkymiä on kuvattu

4. https://github.com/jarovaisanen/DrumOnsetDetectionCNN



(a) Tappio

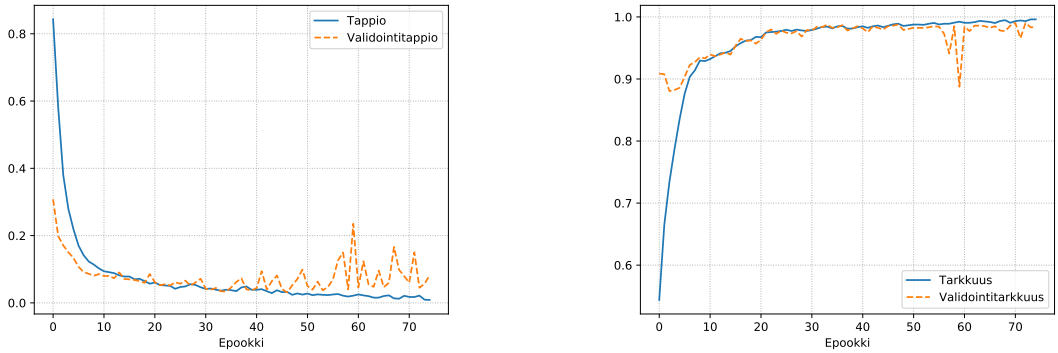


(b) Tarkkuus

Kuvio 12: Kuvaajilla ilmennetään opetuksessa seurattua tappiota ja tarkkuutta ennenaikaisessa pysäytyksessä. Kun validointitappio alkaa kasvamaan suhteessa tappioon, neuroverkon opetus on syytä lopettaa ylisovittamisen välttämiseksi. Mallia opetettiin yhden kappaleen datajoukolla kahdeksan epookin ajan käyttäen 128 eräkokoa.

Taulukko 1: Triviaalin opetustapauksen oppimistulokset kullekin rumpuinstrumentille 64, 256 ja 512 eräko'oilla.

Eräkoko	Metriikka	BD	SD
64	P	0.93 ± 0.037	0.858 ± 0.037
	R	0.901 ± 0.037	0.811 ± 0.060
	F	0.912 ± 0.018	0.827 ± 0.042
256	P	0.886 ± 0.057	0.785 ± 0.028
	R	0.922 ± 0.039	0.876 ± 0.035
	F	0.898 ± 0.031	0.819 ± 0.022
512	P	0.772 ± 0.013	0.731 ± 0.010
	R	0.905 ± 0.026	0.832 ± 0.017
	F	0.821 ± 0.009	0.767 ± 0.010



(a) Tappio

(b) Tarkkuus

Kuvio 13: Kuvaajilla ilmennetään opetuksessa seurattua tappiota ja tarkkuutta bassorummun iskutapahtumien havaitsemisessa. Kuvaajissa näkyvät piikit aiheutuvat opetuksessa käytettävien vaihtelevien optimointierien (engl. *minibatch*) laadusta. Mallia opetettiin yhden kappaleen datajoukolla 75 epookin ajan käyttäen 64 eräkokoa.

luvuissa 6 ja 7.

6 Tulokset

Tässä luvussa esitellään tutkimuksen tuloksia, haasteita ja kehityskohteita. Tulokset on raportoitu vertailukelpoisesta ENST-aineistoon perustuvasta iskutapahtumien havaitsemisesta sekä metallimusiikkiaineistoa vasten suoritetusta iskutapahtumien havaitsemisesta. Metriikat konvoluutioneuroverkkojen opetuksesta ja evaluoinnista on kuvattu taulukoihin ja kuvaajiin. Tutkimuksessa päästiin parempiin tuloksiin aikaisempaan Jacquesin ja Roebelin 2018 kehittämään menetelmään verrattuna.

6.1 Menetelmän tulokset

Tutkimuksessa ENST-aineiston avulla kehitetty menetelmä toimi Jacquesin ja Roebelin menetelmään verrattuna paremmin virveli-iskutapahtumien havaitsemisessa, ja hieman paremmin bassorummun iskutapahtumien havaitsemisessa. Neuroverkkomalleja opetettiin 175-250 epookin ajan ennenaikaisesta pysäytyksestä riippuen kullekin rumpuinstrumentille 64, 256, 512 ja 1024 eräko'oilla. Kunkin rumpuinstrumentin ja eräkoon 35 opetetusta mallista valikoitiin kolme parasta, joista laskettiin keskiarvot ja keskihajonnat lopullisiin tuloksiin. Tuloksia on vertailtu taulukossa 2. Kummankaan vertaillun menetelmän tuloksissa ei ole otettu huomioon ajan suhteen tehtävää normalisointia, jolla Jacques ja Roebel (2018) olivat saaneet bassorummun havaitsemisessa parempia tuloksia (F-arvo 5,3 % parempi). Ajan suhteen tehtävä normalisointi korostaa havaittavan energian muutosta, joka voi olla merkki iskutapahtumasta.

Taulukosta 2 ilmenee opetusajoista saatujen metriikoiden keskiarvot ja keskihajonnat eräko'oittain bassorummun ja virvelin iskutapahtumien havaitsemiselle. Kehitetty menetelmä havaitsi bassorummun iskutapahtumat vain hieman paremmin aikaisempaan menetelmään verrattuna, mutta virvelin iskutapahtumat havaittiin selvästi paremmalla tarkkuudella. Triviaaleista tuloksista (Taulukko 1) poiketen kunkin rumpuinstrumentin iskutapahtumien havaitsemisessa suuremmat eräkoot antoivat parempia tuloksia. Suurempi eräkoko toimii siis paremmin isomman aineiston kanssa, vaikka kummassakin käytettiin samaa oppimistahtia. Tuloksissa päästiin parempiin lukemiin aiempaan menetelmään verrattuna, vaikka Jacques ja

Taulukko 2: Koko aineiston oppimistulokset kullekin rumpuinstrumentille 64, 256, 512 ja 1024 eräko'oilla Jacquesin ja Roebelin menetelmään verrattuna.

Menetelmä	Metriikka	BD	SD
Eräkoko 64	P	0.682 ± 0.005	0.593 ± 0.002
	R	0.836 ± 0.006	0.756 ± 0.003
	F	0.729 ± 0.004	0.617 ± 0.004
Eräkoko 256	P	0.719 ± 0.010	0.633 ± 0.007
	R	0.806 ± 0.010	0.742 ± 0.006
	F	0.753 ± 0.005	0.666 ± 0.007
Eräkoko 512	P	0.773 ± 0.011	0.669 ± 0.011
	R	0.763 ± 0.012	0.702 ± 0.010
	F	0.767 ± 0.007	0.683 ± 0.006
Eräkoko 1024	P	0.816 ± 0.009	0.725 ± 0.010
	R	0.734 ± 0.010	0.668 ± 0.004
	F	0.768 ± 0.006	0.691 ± 0.005
Jacques ja Roebel (2018, Taulukko 2)	P	0.775	0.579
	R	0.750	0.670
	F	0.762	0.621

Roebel (2018) eivät olleet eritelleet datan esikäsittelyvaiheita eikä koodia toteutuksesta ollut saatavissa.

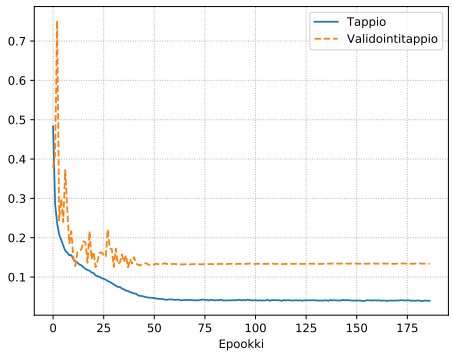
Jacques ja Roebel (2018) evaluoivat neuroverkkomallejaan kolminkertaisella ristiinvalidoinnilla verrattuna tämän tutkimuksen evaluointimenetelmään, jossa testiaineistoksi valikoitiin 10 % kaikkien näytteiden joukosta. Pienestä harppaus-arvosta ($\mathscr{H} = 441$) johtuen opetusnäytteitä oli 218 553 ja testinäytteitä 24 284. Suuren näytemäärän takia ristiinvalidoinnille ei ollut tarvetta tässä menetelmässä.

Kuten Jacques ja Roebel (2018) totesivat, konvoluutioneuroverkon opettaminen yksittäisten rumpuinstrumenttien iskutapahtumien havaitsemista varten voi aiheuttaa ylisovittamista yleiseen perkussiivisten iskutapahtumien havaitsijaan verrattuna. Käyttämällä aikataulutettua oppimistahtia (engl. *learning rate schedule*) saatiin kuitenkin vähennettyä ylisovittamista merkittävästi. Malli on alttiimpi ylisovittamiselle suurempia eräkokoja käytettäessä, sillä pienemmillä eräko'oilla optimointia ohjataan eri suuntiin pienemmin askelin. Kuviossa 14 ilmennetään ENST-aineistolla opetettua neuroverkkomallia bassorummun iskutapahtumien havaitsemisessa 512 eräkoolla. Virvelin iskutapahtumien havaitsemisen haasteellisuutta on kuvattu kuviossa 15. Virvelirummun laajojen taajuusalueiden vuoksi mallin on vaikea oppia havaitsemaan iskuja hyvällä tarkkuudella. Kuvaajia neuroverkkomallien opetuksesta eri eräko'oilla on nähtävissä liitteessä A (ks. kuviot 19 ja 20).

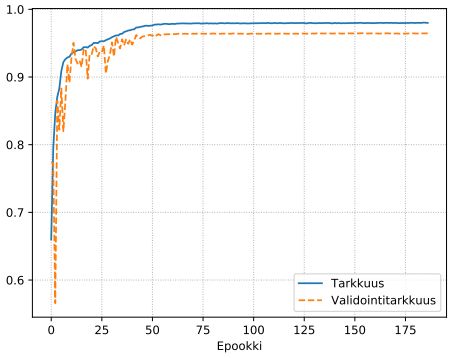
6.2 Metallimusiikki

Kehitettyä menetelmää käytettiin myös metallimusiikkiaineiston kanssa. Aineisto oli kuitenkin pieni ja homogeeninen verrattuna ENST-aineistoon, minkä seurauksena pelkällä metallimusiikkiaineistolla tehdyn opetuksen tulokset olivat erityisen hyviä. Pienestä aineistosta johtuen malleista oli myös helpommin havaittavissa ylisovittamista. Taulukkojen 3 ja 4 tuloksista nähdään, että ENST-aineistolla todennetun menetelmän datan esikäsittely ja perkussiivisten elementtien korostaminen spektrogrammeissa toimivat hyvin myös metallimusiikissa.

Taulukkoon 3 on kuvattu ENST-aineistolla opetettujen ja metallimusiikkiaineistolla testattujen mallien tulokset. Testiaineisto sisälsi vain yhden rumpalin soittamia rumpuja, joissa

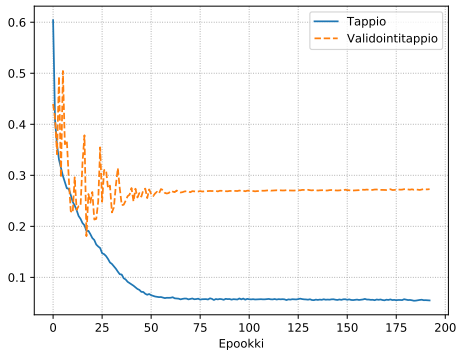


(a) Tappio

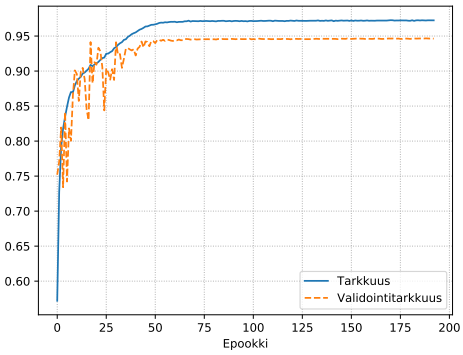


(b) Tarkkuus

Kuvio 14: Kuvaajilla ilmennetään opetuksessa seurattua tappiota ja tarkkuutta bassorummun iskutapahtumien havaitsemisessa. Aikataulutettua oppimistahtia käyttämällä neuroverkon optimointi on nopeampaa. Mallia opetettiin ENST-aineistolla 187 epookin ajan käyttäen 512 eräkokoa.



(a) Tappio



(b) Tarkkuus

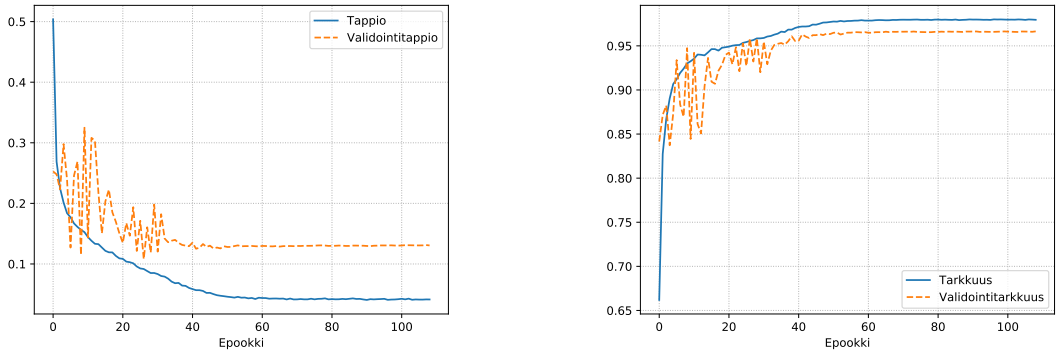
Kuvio 15: Kuvaajilla ilmennetään opetuksessa seurattua tappiota ja tarkkuutta virvelin iskutapahtumien havaitsemisessa. Kuvioon 14 verrattuna validointitappio vaihtelee enemmän suhteessa opetuksen tappioon. Mallia opetettiin ENST-aineistolla 193 epookin ajan käyttäen 512 eräkokoa.

Taulukko 3: ENST-aineistolla opetettujen ja metallimusiikkiaineistolla testattujen ajojen tulokset kullekin rumpuinstrumentille 64, 256 ja 512 eräko'oilla.

Eräkoko	Metriikka	BD	SD
64	P	0.794 ± 0.004	0.562 ± 0.003
	R	0.951 ± 0.008	0.832 ± 0.009
	F	0.852 ± 0.002	0.525 ± 0.010
256	P	0.841 ± 0.008	0.579 ± 0.010
	R	0.959 ± 0.006	0.861 ± 0.014
	F	0.889 ± 0.007	0.574 ± 0.026
512	P	0.863 ± 0.003	0.635 ± 0.007
	R	0.947 ± 0.002	0.897 ± 0.011
	F	0.9 ± 0.003	0.682 ± 0.010

miksaus oli jokaisen raidan kesken samankaltainen. Opetusaineisto sisälsi 218 553 näytettä ja testiaineisto 24 283 näytettä, jolloin testiaineisto oli 10 % koko aineiston koosta. Neuroverkkomalleja opetettiin 100-150 epookin ajan ennenaikaisesta pysäytyksestä riippuen kullekin rumpuinstrumentille 64, 256 ja 512 eräko'oilla. Kunkin rumpuinstrumentin ja eräkoon yhdeksästä opetetusta mallista valikoitiin kolme parasta, joista laskettiin keskiarvot ja keskihajonnat lopullisiin tuloksiin.

Tässä soveltavassa tapauksessa bassorummun iskutapahtumat onnistuttiin havaitsemaan erityisen hyvällä tarkkuudella etenkin 256 ja 512 eräko'oilla. Virveli-iskuissa suurempi eräkoko antoi parempia tuloksia. Bassorummun taajuudet asettuvat hyvin erottuville matalille taajuusalueille, jolloin neuroverkkomallin on helpompi oppia tunnistamaan niitä virveliiskuihin verrattuna. Opetus- ja testiaineiston bassorumpujen eroavaisuudella ei ollut heikentävää vaikutusta tuloksiin. Virveli-iskutapahtumia havaittiin heikommin taulukon 2 tuloksiin verrattuna, sillä testauksessa käytetyn metallimusiikkiaineiston virveli erosi selvästi ENST-aineistossa esiintyvistä virveleistä. Testauksessa käytetyn metallimusiikkiaineiston homogeenisyyden vuoksi tulokset eivät ole vertailukelpoisia yleiskäyttöisenä iskutapahtumien havaitsijana. Kuviossa 16 ilmennetään ENST-aineistolla opetetun ja metallimusiikkiaineistolla



(a) Tappio

(b) Tarkkuus

Kuvio 16: Kuvaajilla ilmennetään opetuksessa seurattua tappiota ja tarkkuutta bassorummun iskutapahtumien havaitsemisessa. Mallia opetettiin 109 epookin ajan käyttäen 512 eräkokoa.

testatun bassorummun iskutapahtumien havaitsemista.

Taulukkoon 4 on kuvattu tulokset pelkällä metallimusiikkiaineistolla tehdylle rumpuiskutapahtumien havaitsemiselle. Metallimusiikkiaineisto oli paljon yksipuolisempi ja lähes kolme kertaa pienempi kuin ENST-aineisto. Opetusaineisto sisälsi 85 446 näytettä ja testiaineisto 9 495 näytettä, joka oli 10 % koko aineiston koosta. Neuroverkkomalleja opetettiin 100-150 epookin ajan ennenaikaisesta pysäytyksestä riippuen kullekin rumpuinstrumentille 64, 256 ja 512 eräko'oilla. Kunkin rumpuinstrumentin ja eräkoon 20 opetetusta mallista valikoitiin viisi parasta, joista laskettiin keskiarvot ja keskihajonnat lopullisiin tuloksiin.

Metallimusiikkiaineiston homogeenisyyden vuoksi tarkastellut metriikat ovat erityisen korkeat; neuroverkkomalli suoriutui alkuperäisten oletusten mukaan liian hyvin. Eräkoolla oli merkitystä ainoastaan virveli-iskutapahtumien havaitsemisessa, kun käytetty eräkoko oli 64 (ks. kuvio 21). Pieneen erään ei mahtunut tarpeeksi monipuolisia näytteitä, jotta olisi päästy yhtä korkeisiin tuloksiin kuin isommilla eräko'oilla. Metallimusiikkiin kohdistuvien tulosten perusteella datajoukkoa olisi ollut tarpeen laajentaa, jotta opetettu neuroverkkomalli toimisi paremmin yleisenä iskutapahtumien havaitsijana myös metallimusiikissa. Jacques ja Roebel (2019) totesivat, että konvoluutioneuroverkon käyttämän opetusaineiston laajentaminen augmentoidulla datalla paransi neuroverkkomallin evaluoinnissa täsmällisyyttä ja tunnistuskykyä. Neuroverkko siis oppii enemmän piirteitä, mutta kykenee myös valitsemaan ne parem-

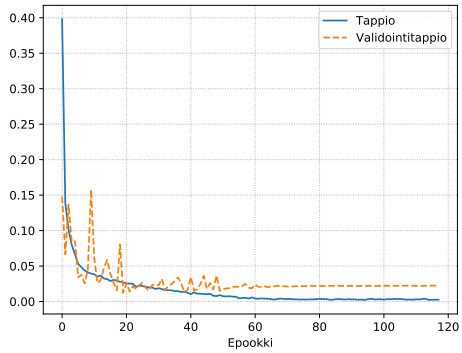
Taulukko 4: Metallimusiikkiaineistolla opetettujen ja evaluoitujen ajojen tulokset kullekin rumpuinstrumentille 64, 256 ja 512 eräko'oilla.

Eräkoko	Metriikka	BD	SD
64	P	0.968 ± 0.002	0.846 ± 0.012
	R	0.992 ± 0.001	0.967 ± 0.003
	F	0.98 ± 0.001	0.896 ± 0.007
256	P	0.981 ± 0.003	0.952 ± 0.007
	R	0.986 ± 0.002	0.949 ± 0.004
	F	0.984 ± 0.002	0.95 ± 0.005
512	P	0.981 ± 0.002	0.971 ± 0.007
	R	0.986 ± 0.003	0.951 ± 0.004
	F	0.984 ± 0.001	0.961 ± 0.005

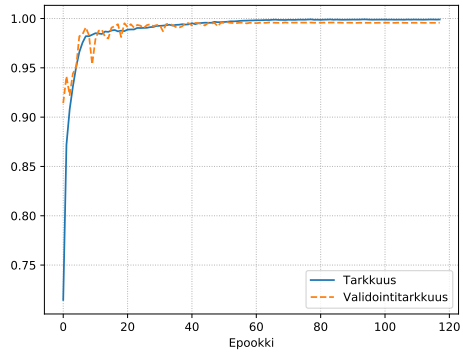
malla tarkkuudella. Data-augmentaatiostrategia ja siinä käytetyt parametrit tulee kuitenkin valita huolellisesti (Jacques ja Roebel 2019).

6.3 Haasteet

Tulosten analysoinnin aikana ilmeni haasteina olevan muun muassa datajoukon pieni koko, tunnisteiden epätasapainoisuus ja aikaisempien menetelmien datan esikäsittelyvaiheiden puuttuminen. Merkattuja iskutapahtumia oli reilusti vähemmän kuin tarkastelupisteitä, joissa iskutapahtumaa ei ollut. Vaikka luokkia tasapainotettiin, dataa oli kokonaisuudessaan liian vähän näin haastavan ongelman ratkaisemiseksi. Epätasapainoisuuden ratkaisemiseksi voitaisiin tulevaisuudessa esimerkiksi syöttää enemmän iskutapahtumia sisältävää raakadataa, tai suorittaa tasapainottamista joko ylinäytteistämällä (engl. *oversampling*) tai alinäytteistämällä (engl. *undersampling*). Ylinäytteistämisessä datajoukko laajenee, sillä iskutapahtumia sisältäviä näytteitä lisätään suhteessa muuhun dataan. Ylinäytteistäminen voi myös aiheuttaa ylisovittamista, sillä iskutapahtumanäytteitä lisättäisiin olemassa olevien iskutapahtumien perusteella. Tällöin datajoukkoon tulee toistuvia näytteitä. Alinäytteistämisessä datajoukko pienenee, sillä näytteitä, joissa iskutapahtumaa ei ole, poistetaan aineistosta paremman tasa-



(a) Tappio



(b) Tarkkuus

Kuvio 17: Kuvaajilla ilmennetään opetuksessa seurattua tappiota ja tarkkuutta bassorummun iskutapahtumien havaitsemisessa. Mallia opetettiin 118 epookin ajan käyttäen 256 eräkokoa.

painon saavuttamiseksi.

Jacques ja Roebel (2018) ja Schlüter ja Böck (2014) eivät eritelleet datan esikäsittelyvaiheita tarpeeksi tarkalla tasolla, eikä koodia toteutuksista ollut saatavissa, joten pohja datan esikäsittelylle luotiin melko vähäisten saatavissa olleiden tietojen perusteella.

Kuten Paulus ja Klapuri (2009) huomauttivat, evaluoinnin haasteena oli mahdollinen tilanne, jossa neuroverkkoa opetettiin kahdella musiikkilajilla (esim. rock ja jazz), mutta evaluointi suoritettiin kolmannella musiikkilajilla (esim. salsa). Tämän välttämiseksi päädyin jakamaan opetus- ja evaluointiaineiston (testi- ja validointiaineiston) pilkotuista spektrogrammeista sen sijaan, että jakaminen olisi tehty kappaleiden eli eri musiikkilajien välillä. Tällöin kokonaisdatajoukko on monipuolisempi, mutta opetus- ja evaluointiaineisto saattavat sisältää pienen osan samankaltaisia spektrogrammeja, mikäli kappaleen sisäiset iskutapahtumanäytteet (kuten virveli-iskut) ovat keskenään yhdenmukaisia.

Aiemman tutkimuksen puutteellisen toistettavuuden vuoksi vertailtuihin tuloksiin tulee suhtautua tarkkaavaisesti. Koska koodeja ei ollut saatavissa, tuloksiin johtaneet evaluointikehykset ja datan esikäsittelyvaiheet saattavat olla keskenään erilaiset. Jokaista yksityiskohtaista askelta ei pystytty toistamaan varmuudella, minkä seurauksena tässä tutkimuksessa toteutettua menetelmää tulee arvioida harkitusti. Tämä tutkimus on kuitenkin tehty toistettavaksi

ja jatkokehitys mahdolliseksi täsmällisesti raportoidun menetelmän ansiosta, sillä toteutetut koodit ja käytetyt parametrit ovat julkisesti saatavissa avoimen lähdekoodin repositoriosta.

7 Yhteenveto

Tässä tutkielmassa tutkittiin polyfonisen musiikin seassa olevien rumpuiskutapahtumien havaitsemista konvoluutioneuroverkoilla. Validin tutkimuksen toteuttamiseksi jäljennettiin aiemmin toteutettuja menetelmiä. Aiemmista tutkimuksista puuttuneiden datan esikäsittelyvaiheiden raportointi vaikeutti tämän tutkimuksen menetelmän toteuttamista, mutta tulokset yksittäisten rumpuiskutapahtumien havaitsemiselle olivat silti parempia aikaisempaan tutkimukseen verrattuna. Toteutettua menetelmää sovellettiin metallimusiikissa esiintyvien rumpuiskutapahtumien havaitsemiseen. Laajempaa, toistettavaa metallimusiikkiaineistoa ei ollut saatavissa, joten aineistona käytettiin *Riddle Me This* -yhtyeen studioäänityksiä. Menetelmä soveltui hyvin metallimusiikkiin, mutta monipuolisempi säröelementtejä sisältävä aineisto olisi tarpeen vastaavanlaisten musiikkilajien iskutapahtumien havaitsemista varten.

Menetelmiä iskutapahtumien havaitsemiseen on tarpeen kehittää automaattisen rumputranskription tutkimusalan edistämistä varten. Konvoluutioneuroverkot soveltuvat tehtävään hyvin tutkimustulosten ja MIREX-evaluointien perusteella, minkä takia menetelmää olisi tarpeen laajentaa kattamaan myös muita rumpuinstrumentteja. Tutkielmassa toteutetulla opetusalgoritmilla opetettu neuroverkkomalli voitaisiin ottaa käyttöön Madmom-kirjastossa osana erillistä iskutapahtumien havaitsija -moduulia. Madmom [1] on Pythonille kirjoitettu laajasti käytetty avoimen lähdekoodin audioprosessointi- ja MIR-kirjasto, joka operoi eri tehtäviin tarkoitetuilla prosessoreilla (Böck ym. 2016). Prosessoreita voidaan käyttää joko itsenäisesti tai ketjutettuna erinäisissä MIR-tehtävissä, kuten musikaalisten tapahtumien havaitsemisessa, rytmin ja tahdinosien tunnistamisessa, tempon arvioinnissa ja transkriptiossa.

Tutkimuksessa päästiin asetettuihin tavoitteisiin toteutetulla menetelmällä ja saaduilla tuloksilla, mistä on kiittäminen erityisesti Jyväskylän yliopiston IT-infrastruktuuria. Yleisen laskentainfrastruktuurin käyttö mahdollisti neuroverkkojen opettamisen ja evaluoinnin tehokkailla GPU-palvelimilla.

1. https://github.com/CPJKU/madmom

Lähteet

Bello, J. P., L. Daudet, S. Abdallah, C. Duxbury, M. Davies ja M. B. Sandler. 2005. “A tutorial on onset detection in music signals”. *IEEE Transactions on Speech and Audio Processing* 13 (5): 1035–1047.

Benetos, Emmanouil, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff ja Anssi Klapuri. 2013. “Automatic music transcription: Challenges and future directions”. *Journal of Intelligent Information Systems* 41 (joulukuu). https://doi.org/10.1007/s10844-013-0258-3.

Böck, Sebastian, Filip Korzeniowski, Jan Schlüter, Florian Krebs ja Gerhard Widmer. 2016. “madmom: a new Python Audio and Music Signal Processing Library”. *CoRR* abs/1605.07008. arXiv: 1605.07008. http://arxiv.org/abs/1605.07008.

Davies, Matthew, Guy Madison, Pedro Silva ja Fabien Gouyon. 2013. “The Effect of Microtiming Deviations on the Perception of Groove in Short Rhythms”. *Music Perception: An Interdisciplinary Journal* 30 (5): 497–510. ISSN: 0730-7829. https://doi.org/10.1525/mp.2013.30.5.497. eprint: https://mp.ucpress.edu/content/30/5/497.full.pdf. https://mp.ucpress.edu/content/30/5/497.

Dittmar, Christian, Estefanía Cano, Jakob Abeßer ja Sascha Grollmisch. 2012. “Music Information Retrieval Meets Music Education”. Teoksessa *Multimodal Music Processing,* toimittanut Meinard Müller, Masataka Goto ja Markus Schedl, 3:95–120. Dagstuhl Follow-Ups. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN: 978-3-939897-37-8. https://doi.org/10.4230/DFU.Vol3.11041.95. http://drops.dagstuhl.de/opus/volltexte/2012/3468.

Dittmar, Christian, ja Daniel Gärtner. 2014. “Real-Time Transcription and Separation of Drum Recordings Based on NMF Decomposition”. Teoksessa *17th Int. Conference on Digital Audio Effects (DAFx-14),* 1–8. Syyskuu.

Dittmar, Christian, Martin Pfleiderer, Stefan Balke ja Meinard Müller. 2017. “A Swingogram Representation for Tracking Micro-Rhythmic Variation in Jazz Performances”. *Journal of New Music Research* 47, numero 2 (elokuu): 97–113. https://doi.org/10.1080/09298215.2017.1367405.

Dittmar, Christian, Martin Pfleiderer ja Meinard Müller. 2015. “Automated Estimation of Ride Cymbal Swing Ratios in Jazz Recordings”. Teoksessa *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015,* toimittanut Meinard Müller ja Frans Wiering, 271–277. http://ismir2015.uma.es/articles/143%5C_Paper.pdf.

Driedger, Jonathan, Meinard Müller ja Sascha Disch. 2014. “Extending Harmonic-Percussive Separation of Audio Signals.” Teoksessa *Proceedings of the 15th International Society for Music Information Retrieval Conference,* 611–616. Taipei, Taiwan: ISMIR, lokakuu. https://doi.org/10.5281/zenodo.1415226.

Elowsson, Anders, ja Anders Friberg. 2013. “Modelling Perception of Speed in Music Audio”. Teoksessa *Proceedings of the Sound and Music Computing Conference 2013, SMC 2013,* 735–741. Stockholm, Sweden, heinäkuu. ISBN: 978-3-8325-3472-1. https://doi.org/10.5281/zenodo.850321.

FitzGerald, Derry. 2004. “Automatic drum transcription and source separation”. Tohtorinväitöskirja, Technological University Dublin. https://doi.org/10.21427/D7002Z.

Fitzgerald, Derry. 2010. “Harmonic/Percussive Separation using Median Filtering”. *13th International Conference on Digital Audio Effects (DAFx-10)* (Graz, Austria) (syyskuu).

FitzGerald, Derry, ja Jouni Paulus. 2006. “Unpitched Percussion Transcription”. Teoksessa *Signal Processing Methods for Music Transcription,* toimittanut Anssi Klapuri ja Manuel Davy, 131–162. Boston, MA: Springer US. ISBN: 978-0-387-32845-4. https://doi.org/10.1007/0-387-32845-9_5.

Ganguly, Antra, ja Manisha Sharma. 2017. “Detection of pathological heart murmurs by feature extraction of phonocardiogram signals”. *Journal of Applied and Advanced Research* 2 (heinäkuu). https://doi.org/10.21839/jaar.2017.v2i4.94.

Gillet, Olivier, ja Gaël Richard. 2006. “ENST-Drums: an extensive audio-visual database for drum signals processing.” Teoksessa *Proceedings of the 7th International Conference on Music Information Retrieval,* 156–159. Victoria, Canada: ISMIR, lokakuu. https://doi.org/10.5281/zenodo.1415902.

———. 2008. “Transcription and Separation of Drum Signals From Polyphonic Music”. *Audio, Speech, and Language Processing, IEEE Transactions on* 3 (huhtikuu): 529–540. https://doi.org/10.1109/TASL.2007.914120.

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

Goto, M., ja Y. Muraoka. 1994. “A sound source separation system for percussion instruments”. *The Transactions of the Institute of Electronics, Information and Communication Engineers D-II* Vol.J77-D-II, numero 5 (toukokuu): 901–911.

Grollmisch, S., C. Dittmar ja G. Gatzsche. 2009. “Concept, implementation and evaluation of an improvisation based music video game”. Teoksessa *2009 International IEEE Consumer Electronics Society’s Games Innovations Conference,* 210–212. Elokuu. https://doi.org/10.1109/ICEGIC.2009.5293599.

Helén, M., ja T. Virtanen. 2005. “Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine”. Teoksessa *2005 13th European Signal Processing Conference,* 1–4. Syyskuu.

Hemanth, D. J., ja Vania Vieira Estrela. 2017. *Deep Learning for Image Processing Applications.* Advances in Parallel Computing, volume 31. IOS Press. ISBN: 9781614998211. http://search.ebscohost.com.ezproxy.jyu.fi/login.aspx?direct=true&db=nlebk&AN=1791226&site=ehost-live.

Herbst, Jan. 2017. “Historical development, sound aesthetics and production techniques of metal’s distorted electric guitar”. *Metal Music Studies* 3 (maaliskuu): 1–17. https://doi.org/10.1386/mms.3.1.23_1.

Jacques, Celine, ja Axel Roebel. 2018. “Automatic drum transcription with convolutional neural networks”. Teoksessa *21th International Conference on Digital Audio Effects, Sep 2018, Aveiro, Portugal.* Aveiro, Portugal, syyskuu. https://hal.archives-ouvertes.fr/hal-02018777.

———. 2019. “Data Augmentation for Drum Transcription with Convolutional Neural Networks”. Teoksessa *Published in Proceedings of the 27th European Signal Processing Conference (EUSIPCO), 2019,* 1–5. Syyskuu. https://doi.org/10.23919/EUSIPCO.2019.8902980.

LeCun, Yann, ja Yoshua Bengio. 1998. “Convolutional Networks for Images, Speech, and Time Series”. Teoksessa *The Handbook of Brain Theory and Neural Networks,* 255–258. Cambridge, MA, USA: MIT Press. ISBN: 0262511029.

Lerch, Alexander. 2012. *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics.* 1st. Wiley-IEEE Press. ISBN: 111826682X.

Litovsky, Ruth, H. Steven Colburn, William Yost ja Sandra Guzman. 1999. “The precedence effect”. *The Journal of the Acoustical Society of America* 106 (marraskuu): 1633–1654. https://doi.org/10.1121/1.427914.

McFee, Brian, Colin Raffel, Dawen Liang, Daniel Ellis, Matt Mcvicar, Eric Battenberg ja Oriol Nieto. 2015. “librosa: Audio and Music Signal Analysis in Python”. Teoksessa *Proceedings of the 14th Python in Science Conference,* 18–24. Tammikuu. https://doi.org/10.25080/Majora-7b98e3ed-003.

Mesaros, Annamari, Toni Heittola, Antti Eronen ja Tuomas Virtanen. 2010. “Acoustic event detection in real life recordings”. Teoksessa *Proceedings of the 18th European Signal Processing Conference, Eusipco 2010, Aalborg, Denmark, 23-27 August 2010,* 1267–1271. Elokuu.

Moorer, James A. 1975. “On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer”. Tohtorinväitöskirja, Stanford University. https://ccrma.stanford.edu/files/papers/stanm3.pdf.

Ono, N., K. Miyamoto, J. Le Roux, H. Kameoka ja S. Sagayama. 2008. “Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram”. Teoksessa *2008 16th European Signal Processing Conference,* 1–4.

Paulus, Jouni. 2009. “Signal Processing Methods for Drum Transcription and Music Structure Analysis”. Tohtorinväitöskirja, Tampere University of Technology, joulukuu.

Paulus, Jouni, ja Anssi Klapuri. 2009. “Drum Sound Detection in Polyphonic Music with Hidden Markov Models”. *EURASIP Journal on Audio, Speech, and Music Processing* 2009 (tammikuu): 1–9. https://doi.org/10.1155/2009/497292.

Pons, J., T. Lidy ja X. Serra. 2016. “Experimenting with musically motivated convolutional neural networks”. Teoksessa *2016 14th International Workshop on Content-Based Multimedia Indexing (CBMI),* 1–6. https://doi.org/10.1109/CBMI.2016.7500246.

Rabiner, L. R. 1989. “A tutorial on hidden Markov models and selected applications in speech recognition”. *Proceedings of the IEEE* 77, numero 2 (helmikuu): 257–286. https://doi.org/10.1109/5.18626.

Roebel, Axel, Céline Jacques ja Achille Aknin. 2018. “MIREX 2018: Training CNN Onset Detectors With Artificially Augmented Datasets”. *19th International Society for Music Information Retrieval Conference,* numero 5, 1–3.

Schloss, W. A. 1985. “On the Automatic Transcription of Percussive Music - From Acoustic Signal to High-Level Analysis”. Tutkielma, Stanford University, toukokuu. https://ccrma.stanford.edu/files/papers/stanm27.pdf.

Schlüter, Jan, ja Sebastian Böck. 2014. “Improved musical onset detection with Convolutional Neural Networks”. Teoksessa *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 6979–6983. Toukokuu. https://doi.org/10.1109/ICASSP.2014.6854953.

Smith, Steven W. 1997. *The Scientist and Engineer’s Guide to Digital Signal Processing.* USA: California Technical Publishing. ISBN: 0966017633.

Southall, Carl, Ryan Stables ja Jason Hockman. 2017. “Automatic Drum Transcription for Polyphonic Recordings Using Soft Attention Mechanisms and Convolutional Neural Networks.” Teoksessa *Proceedings of the 18th International Society for Music Information Retrieval Conference,* 606–612. Suzhou, China: ISMIR, lokakuu. https://doi.org/10.5281/zenodo.1415616.

Tan, Li, ja Jean Jiang. 2013. *Digital Signal Processing: Fundamentals and Applications.* 2nd. USA: Academic Press, Inc. ISBN: 0124158935.

Wang, Qi, ja Yonghong Yan. 2017. "A Two-Stage Approach to Note-Level Transcription of a Specific Piano". *Applied Sciences (Switzerland)* 7 (syyskuu): 1–19. https://doi.org/10.3390/app7090901.

Williams, Duncan. 2014. "Tracking timbral changes in metal productions from 1990 to 2013". *Metal Music Studies* 1 (lokakuu): 39–68. https://doi.org/10.1386/mms.1.1.39_1.

Vogl, R., M. Dorfer ja P. Knees. 2017. "Drum transcription from polyphonic music with recurrent neural networks". Teoksessa *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 201–205. Maaliskuu. https://doi.org/10.1109/ICASSP.2017.7952146.

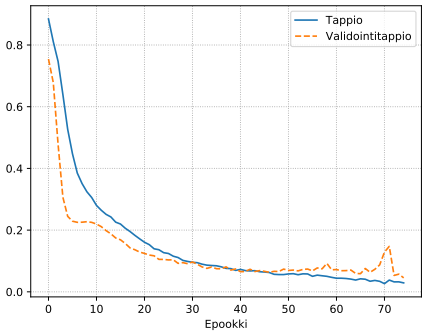
Vogl, Richard, Matthias Dorfer, Gerhard Widmer ja Peter Knees. 2017. "Drum Transcription via Joint Beat and Drum Modeling Using Convolutional Recurrent Neural Networks." Teoksessa *Proceedings of the 18th International Society for Music Information Retrieval Conference,* 150–157. Suzhou, China: ISMIR, lokakuu. https://doi.org/10.5281/zenodo.1415136.

Wu, C., C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Müller ja A. Lerch. 2018. "A Review of Automatic Drum Transcription". *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26, numero 9 (syyskuu): 1457–1483. https://doi.org/10.1109/TASLP.2018.2830113.

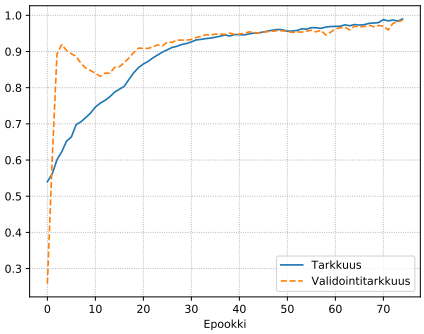
Wu, C., ja A. Lerch. 2015. "Drum transcription using partially fixed non-negative matrix factorization". Teoksessa *2015 23rd European Signal Processing Conference (EUSIPCO),* 1281–1285. Elokuu. https://doi.org/10.1109/EUSIPCO.2015.7362590.

Liitteet

A Kuvaajia konvoluutioneuroverkon opetuksesta

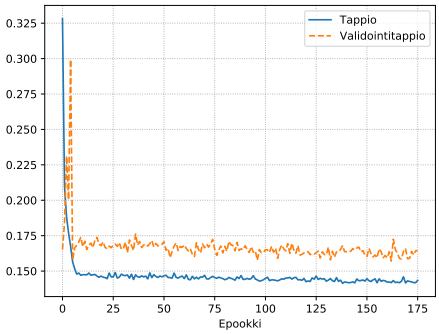


(a) Tappio

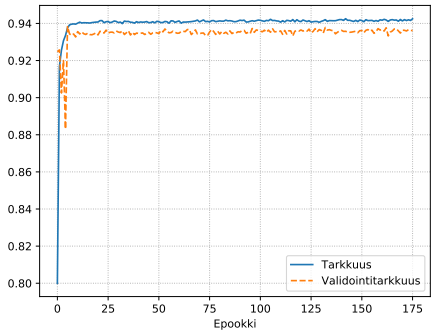


(b) Tarkkuus

Kuvio 18: Kuvaajilla ilmennetään opetuksessa seurattua tappiota ja tarkkuutta bassorummun iskutapahtumien havaitsemisessa. Kuvaajat ovat sileämpiä isommilla eräko'oilla. Mallia opetettiin yhden kappaleen datajoukolla 75 epookin ajan käyttäen 256 eräkokoa.

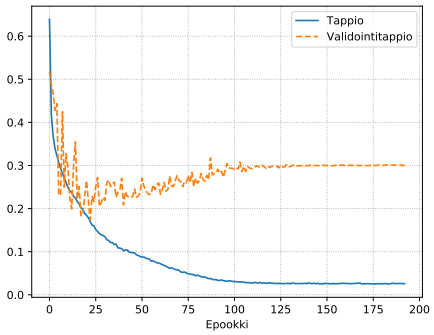


(a) Tappio

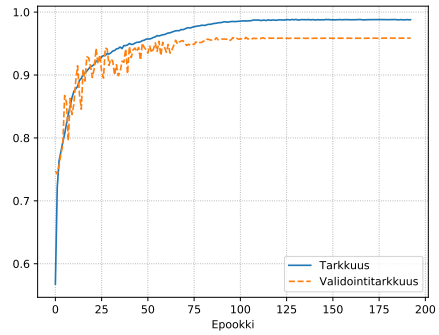


(b) Tarkkuus

Kuvio 19: Kuvaajilla ilmennetään opetuksessa seurattua tappiota ja tarkkuutta bassorummun iskutapahtumien havaitsemisessa. Pienemmällä eräkoolla mallin optimointi sisältää enemmän vaihtelua. Mallia opetettiin ENST-aineistolla 176 epookin ajan käyttäen 64 eräkokoa.

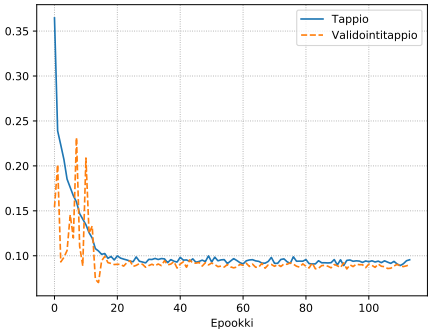


(a) Tappio

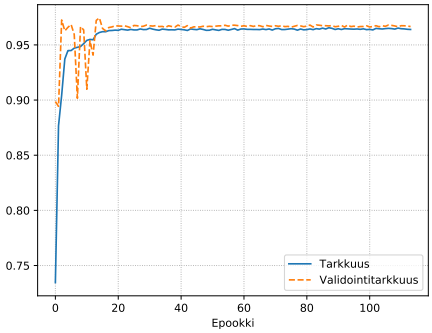


(b) Tarkkuus

Kuvio 20: Kuvaajilla ilmennetään opetuksessa seurattua tappiota ja tarkkuutta virvelin iskutapahtumien havaitsemisessa. Suuremmalla eräkoolla validointitappio alkaa nousta myöhemmin. Mallia opetettiin ENST-aineistolla 193 epookin ajan käyttäen 1024 eräkokoa.



(a) Tappio



(b) Tarkkuus

Kuvio 21: Kuvaajilla ilmennetään opetuksessa seurattua tappiota ja tarkkuutta virvelin iskutapahtumien havaitsemisessa. Kuvaajassa näkyvät piikit aiheutuvat opetuksessa käytettävien vaihtelevien optimointierien laadusta, joiden jälkeen validointitappio ja -tarkkuus asettuvat. Mallia opetettiin homogeenisellä, ylisovittamiselle alttiilla metallimusiikkiaineistolla 114 epookin ajan käyttäen 64 eräkokoa.