Äänen fysikaalisesti perusteltuja augmentointimenetelmiä puheentunnistusjärjestelmää opetettaessa

Pro gradu
Turun yliopisto
Fysiikka
2024
LuK Akseli Wingström
Tarkastajat:
TkL Kenneth Oksanen
Dos. Matti Murtomaa

TURUN YLIOPISTO
Fysiikan ja tähtitieteen laitos

WINGSTRÖM, AKSELI: Äänen fysikaalisesti perusteltuja augmentointimenetelmiä puheentunnistusjärjestelmää opetettaessa

Pro gradu -tutkielma, 77 s.
Fysiikka
Huhtikuu 2024

Syvillä neuroverkoilla on viime vuosina saavutettu merkittäviä parannuksia useissa perinteisissä tekoälyyn kuuluvissa tehtävissä. Neuroverkkojen opettaminen vaatii kuitenkin edelleen valtavasti opetusdataa, jonka kerääminen ja annotoiminen on hidasta.

Puheentunnistusjärjestelmien ongelmana on ollut saatavilla olevan opetusdatan määrä sekä järjestelmien luotettavuus ympäristöissä, joissa järjestelmän syötteessä on mukana paljon häiriötä, kuten taustamelua tai säröilyä.

Augmentoinnilla tarkoitetaan valmiiksi annotoidun opetusdatan muokkaamista siten, että siihen lisätään erilaisia häiriöitä tai muita variaatioita, mutta sen alkuperäinen merkitys kuitenkin säilyy. Tämän avulla valmista opetusdataa voidaan hyödyntää uutena datana neuroverkon opetuksessa.

Opinnäytetyössä parannettiin puheentunnistusjärjestelmän robustisuutta kehittämällä ja optimoimalla fysikaalisesti perusteltuja augmentointimenetelmiä. Työssä keskityttiin hälyjen, säröjen, kaikujen ja taajuusvasteiden augmentointeihin.

Työssä suoritettiin akkumulaatio- sekä ablaatiotestejä, joissa puheentunnistusjärjestelmä opetettiin eri augmentaatiomenetelmiä ja niiden kombinaatioita käyttäen. Opetetut järjestelmät testattiin evaluointidatalla, josta laskettiin Levenshtein -editointietäisyys tulokseksi.

Editointietäisyyksistä arvioitiin Harrell-Davis -evaluointimenetelmällä lopullinen tulos testille, joita vertailtiin keskenään augmentointimenetelmien toimivuuden määrittämiseksi. Augmentointimenetelmien lisäksi vertailtiin opetusaikojen eroa ilman augmentointia ja augmentoinnin kanssa tapahtuneiden opetusten välillä.

Lopputuloksena jokainen augmentointimenetelmä paransi puheentunnistusjärjestelmän robustisuutta jo lyhyellä opetusajalla. Robustiuden parantamisen lisäksi augmentointimenetelmät nopeuttivat neuroverkkojen oppimista.

Asiasanat: augmentointi, data-augmentaatio, tekoäly, koneoppiminen, neuroverkko, puheentunnistusjärjestelmä, ASR, conformeri, spektrogrammi, impulssivaste, taajuusvastekäyrä, akkumulaatiotesti, ablaatiotesti, Levenshtein editointietäisyys

KIITOKSET

Pro gradu -lopputyö toteutettiin osana *Patria Aviation Oy*:n projektia.
Kiitän Patriaa kiinnostavan ja ajankohtaisen lopputyöaiheen mahdollistamisesta.

Suurin kiitokseni menee työni ohjaajille tekniikan lisensiaatti Kenneth Oksaselle sekä dosentti Matti Murtomaalle, joilta sain ammattimaista ohjausta alusta loppuun asti. Lisäksi heidän asiantuntemuksensa ansiosta työn eri osioihin ja asiasisältöihin sai tarvittaessa neuvoa ja apua nopeallakin aikataululla.

Kiitokset myös *Savox Communications Oy*:lle, josta Jussi Havakka sekä Ilkka Huhtakallio jakoivat mikrofoneihin liittyvää asiantuntemustaan. Lisäksi kiitän läheisiäni, opiskelukavereitani sekä kollegoitani, joilta olen saanut tukea ja kannustusta työn suorittamisen ohella.

Akseli Wingström
19.4.2024

Sisällys

Johdanto

Kuluneen vuosikymmenen aikana teknologian kehitys on ollut monilla aloilla merkittävää. Tietokoneiden nopea kehittyminen on mahdollistanut myös neuroverkkopohjaisen syväoppimisen (*engl. Deep Learning*, DL) huomattavan kehityksen. Neuroverkkojen kehittyessä sitä hyödyntävät ohjelmat, kuten automaattiset puheentunnistusjärjestelmät (*engl. Automatic Speech Recognition*, ASR), ovat kehittyneet merkittävästi ja mahdollistaneet kaupallisia tuotteita kuten Apple Siri, Amazon Alexa ja Google Assistant. Tämän opinnäytetyön lopputuloksen tarkoituksena on kehittää vastaavan puheentunnistusjärjestelmän robustisuutta eli luotettavuutta. [1]

Kehittymisestä huolimatta syväoppiminen vaatii nykyäänkin valtavasti dataa, ja edellä mainituilla yrityksillä onkin opetusmateriaalina tuhansien tuntien edestä puhekorpuksia, eli puhutun kielen nauhoitteita. Opetusmateriaalia voidaan lisätä kahdella tavalla; keräämällä uusia ääninäytteitä sekä augmentoimalla (*engl. augmentation*) jo olemassa olevia. Uusien ääninäytteiden kerääminen ja annotoiminen on kuitenkin todella kallista eikä senkään avulla varmisteta, että ääninäytteet kattaisivat kaikenlaiset puheäänet tai esimerkiksi fyysistä stressiä kokevat puhujat. [1]

Uusien ääninäytteiden yksioikoisen lisäämisen rinnalla syväoppimisen opetusmateriaalina käytetään yleisesti myös ääninäytteitä, joissa ääntä on augmentoitu. Augmentoinnilla tarkoitetaan alkuperäisen opetusmateriaalin varioimista jollakin tavalla uudeksi opetusdataksi, kuten ääninäytteessä puhedatan varioimista vastaamaan alkuperäistä puhetta taustameluisessa ympäristössä. [2]

Opinnäytetyössä tutkitaan, kehitetään ja evaluoidaan menetelmiä, joilla alkuperäisistä opetusdataan kuuluvista ääninäytteistä voidaan luoda uusia ääninäytteitä. Uusien ääninäytteiden tulee olla fysikaalisesti perusteltuja ja realistisilta kuulostavia. Lisäksi augmentointimenetelmien tulee olla tietokoneelle laskennallisesti tehokkaita, jotta neuroverkkojen opettaminen ei hidastu niiden vuoksi.

Kehitettävän puheentunnistusjärjestelmän tarkoitus on toimia myös spesifisti määritetyissä ympäristöissä, jotka ovat normaalisti puheentunnistusjärjestelmille haastavia. Tällaisissa ympäristöissä toimivat päivittäin monet viranomaistahot, kuten esimerkiksi moottoripyöräpoliisit.

Augmentointimenetelmät voidaan jakaa aiheuttajansa mukaan kolmeen eri luokkaan: ympäristön, laitteiston ja puhujan variaatioiden mukaisiin augmentointeihin. Tässä opinnäytetyössä keskitytään erityisesti augmentointeihin, jotka liittyvät hälyjen, säröjen, kaikujen sekä taajuusvasteiden muutoksiin. Augmentointimenetelmät toteutetaan käyttökelpoiseksi ohjelmakoodiksi, joiden toimivuutta tutkitaan sekä kuuntelemalla että opettamalla neuroverkkoja.

Ääninäytteiden kuuntelulla pyritään varmistamaan kattava varianssi, realistisia tilanteita vastaavat sekä aidoilta kuulostavat augmentaatiot. Neuroverkkojen opettelussa augmentointimenetelmiä vertaillaan testidatan avulla, jolloin tarkkuutta pystytään mittaamaan esimerkiksi puheentunnistuksessa WER:in (*engl. Word Error Rate*) tai tässäkin työssä käytetyn Levenshteinin editointietäisyyden avulla.

Augmentointimenetelmien suoranaisen kehittämisen lisäksi opinnäytetyössä suoritetaan mikrofonille taajuusvastemittauksia, jonka avulla perustellaan yksi augmentointitavoista.

Opinnäytetyössä kehitettyjen augmentointimenetelmien avulla parannetaan puheentunnistusjärjestelmän robustisuutta. Lopputuloksissa verrataan alkuperäisen ja augmentointimenetelmien avulla saatujen editointietäisyyksien erotusta. Lisäksi käytettyjen augmentointitapojen toimivuuden tai toimimattomuuden syitä pohditaan ja perustellaan fysikaalisesti.

1 Puheentunnistusjärjestelmän teoria

Tässä luvussa kerrotaan puheentunnistusjärjestelmän perusteista, jotta lukija saa perusteellisen ymmärryksen opinnäytetyön taustasta ja tarkoituksesta. Luvuissa 1.1 ja 1.2 käydään läpi perusteet puheentunnistusjärjestelmän opettamisesta ja luvussa 1.3 kerrotaan järjestelmän testaamisesta.

1.1 Tietokone oppimassa puhetta

Tietokoneille opetetaan puhetta spektrogrammien avulla. Spektrogrammeilla voidaan esittää ja analysoida ääntä. Tekoälylle voidaan opettaa esimerkkien avulla, miltä tietyt äänteet näyttävät spektrogrammeissa, jolloin se teoriassa oppii tunnistamaan kyseiset äänteet myös tulevaisuudessa. Äänteet tekoäly pystyy yhdistämään sanoiksi erilaisten optimointimenetelmien avulla. Sanat voidaan opettaa erikseen tekoälylle esimerkiksi sanakirjasta. [1]

Todellisuudessa puheen opettaminen tietokoneelle ei ole edellä mainitun yksinkertaista, koska ei ole yhtä ainoaa universaalia tapaa lausua sanoja. Ihmiset käyttävät eri murteita, lausuvat sanoja eri tavoilla, painottavat eri tavuja sekä puhuvat eri äänen korkeuksilla ja nopeuksilla. Toisin sanoen pelkästään puhetyylissä on monia muuttujia, jotka vaikuttavat spektrogrammin muotoon. Lisäksi spektrogrammiin, ja täten myös tekoälyn mahdollisuuteen tunnistaa sana, vaikuttaa muitakin tekijöitä. Tällaisia ovat esimerkiksi mikrofonin ominaisuudet tai huone, jossa puhe on nauhoitettu, josta esimerkkinä jo mainittu kaiun syntyminen. [3] [4]

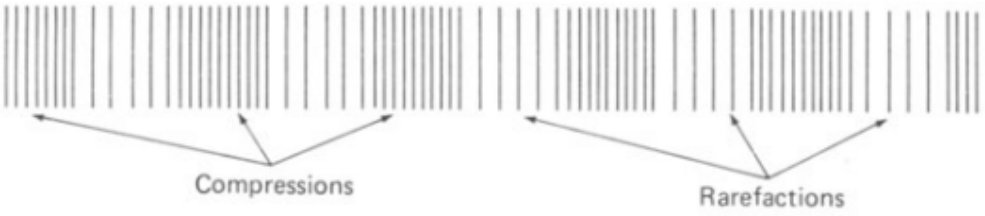
Puheentunnistusjärjestelmän todennäköisyyttä tunnistaa satunnainen sana pystytään parantamaan lisäämällä opetusdataan esimerkkejä, joilla kasvatetaan tekoälylle opetettujen sanojen ja äänteiden määrää. Isommalla sanavarastolla on todennäköisempää, että tekoälylle on opetettu myös satunnainen sana, joka halutaan tunnistaa. Lisäksi aiemmin mainittujen muuttujien, kuten esimerkiksi kaiun, vaikutusta järjestelmän todennäköisyyteen tunnistaa sana voidaan parantaa lisäämällä kaiuin augmentoituja esimerkkejä opetusdataan. Opetusdataa voidaan siis kasvattaa kahdella tavalla: lisäämällä täysin uusia ääninäytteitä sekä augmentoimalla valmiiksi olemassa olevia ääninäytteitä. Opetusdatan muodostamisesta kerrotaan tarkemmin luvussa 2.

Luvussa 1.1.1 käydään läpi perusteita miten ihminen tai tietokone voi ymmärtää ääntä, sekä miten ääntä tallennetaan mikrofoneilla. Luvussa 1.1.2 kerrotaan spektrogrammien perusteista, kuten siitä, miten spektrogrammeja voidaan muodostaa ja miten niistä voidaan lukea eri äänteitä. Tekoälystä ja neuroverkoista puheentunnistusjärjestelmän taustalla kerrotaan luvuissa 1.1.3 sekä 1.1.4. Lopuksi luvuissa 1.2 sekä 1.3 kerrotaan, miten järjestelmää optimoidaan sekä testataan.

1.1.1 Paineen muutoksista spektrogrammiksi

Äänen kuulemiseen tarvitaan äänilähde, väliaine sekä vastaanotin. Väliaine on yleensä ilmaa ja se kuljettaa ääniaaltoa eteenpäin.

Äänilähde muodostaa väliaineeseen ääniaallon, joka on pitkittäistä aaltoliikettä. Ääniaallot muodostuvat kuvan 1a mukaisista ilmanpaineen tihentymistä ja harventumista, joita voidaan myös kuvailla kuvan 1b mukaisesti siniaaltona. [5]



(a) Ilmanpaineen tihentymät (Compressions) ja harventumat (Rarefactions).

(b) Ilmanpaineen vaihtelut aaltomuodossa. Nollatasona normaali ilmanpaine.

Kuva 1: Ääniaallon eteneminen ilmassa. [5]

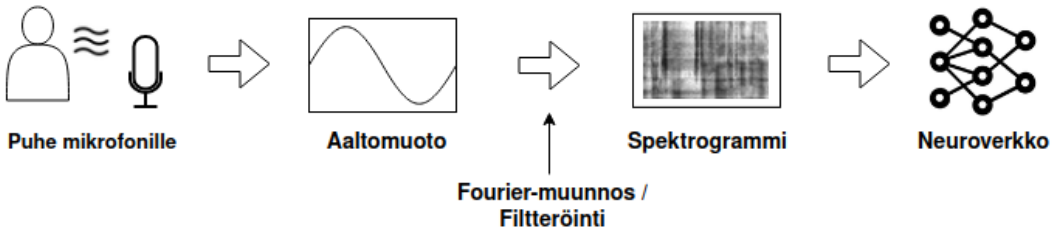
Ilmanpaineen tihentymät ja harventumat muodostavat syklin. Kokonaiseen sykliin kuluvaa aikaa sanotaan jaksonajaksi T. Taajuudeksi

$$f = \frac{1}{T},$$

kutsutaan sekunnissa tapahtuvia syklejä. Taajuudesta saadaan yksiköksi $1/\mathrm{s} = \mathrm{Hz}$ (hertsi). Jaksonaika T määrittää siis taajuuden f, joka taas kertoo kuultavan äänen korkeuden. Normaalit äänentaajuudet ovat 20 - 20 000 Hz, mutta puheen tapauksessa olennaiset taajuudet painottuvat matalammille taajuuksille. [5]

Ihmisellä mainittuna vastaanottimena toimii korva. Ilmassa tulevat paineen vaihtelut päätyvät aluksi tärykalvolle, joka välittää värähtelyjä taas eteenpäin kuuloluille. Kuuloluut voimistavat värähtelyjä eteenpäin simpukan basilaariselle kalvolle. [6]

Basilaarisessa kalvossa ääntä erotellaan taajuuden perusteella; korkeataajuiset äänet saavat basilaarisen kalvon värähtelemään eri tavalla kuin matalataajuiset äänet. Värähtelyiden perusteella värekarvat tuottavat erilaisia sähköisiä signaaleja reseptoreille. Lopuksi signaali päätyy kuulohermolle aivojen tulkittavaksi. [5] [6]



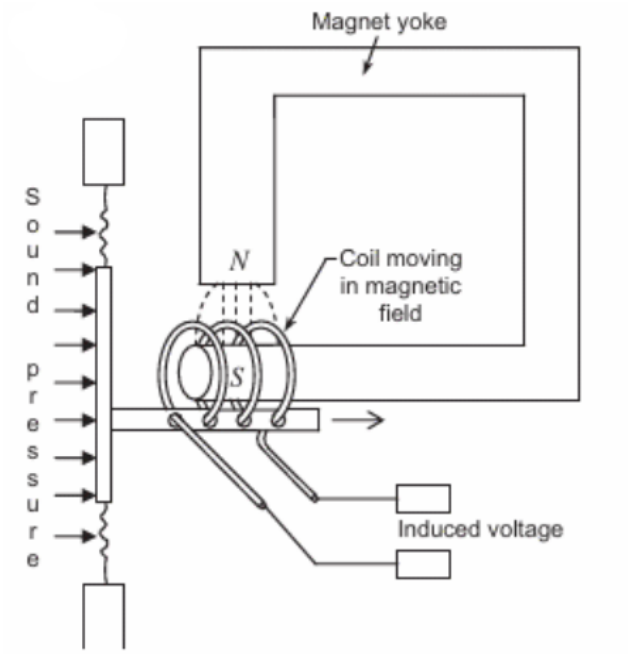
Kuva 2: Havainnekuva äänisignaalin prosessista ennen neuroverkoille syöttämistä.

Edellä kuvattiin, miten ihminen ymmärtää puhetta aivojen avulla monen eri välivaiheen kautta. Kuvassa 2 esiteltynä sama prosessi puheen kuulemisesta tekoälyn tapauksessa, joka on myös monivaiheinen tapahtuma.

Tekoälylle puheen vastaanottimena toimii korvan sijasta mikrofoni. Puhe tallennetaan mikrofonin avulla, joka havaitsee ilmanpaineaaltojen muutokset. Erilaiset mikrofonit nauhoittavat ääntä eri tavoin. Tutustutaan seuraavaksi mikrofoneihin ja niiden toimintamenetelmiin.

Kuten ihmisillä korvat, myös mikrofonit muuntavat ääniaallot sähkösignaaleiksi. Mikrofonit voivat toimia esimerkiksi kelan tai kondensaattorin avulla. [7]

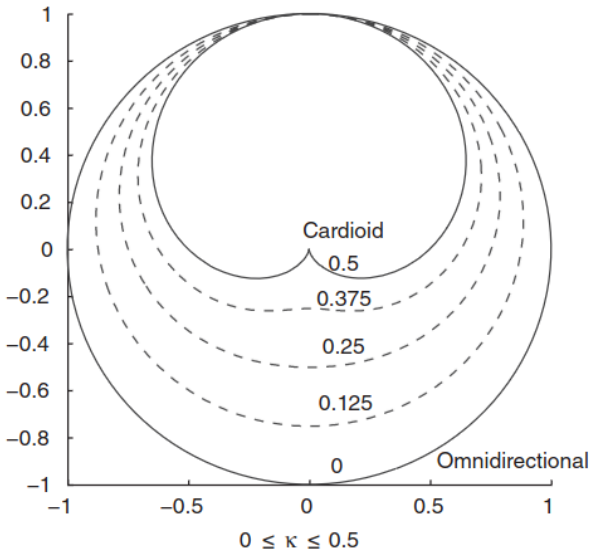
Kelalla toimivaa mikrofonia kutsutaan dynaamiseksi mikrofoniksi. Kuvassa 1 esitellyt ääniaallot liikuttavat dynaamisessa mikrofonissa välikalvoa (*engl. diaphragm*), joka taas liikuttaa kelaa, johon indusoituu magneetilla luodun magneettikentän ansiosta jännite. Välikalvon lisäksi mikrofonissa on sille toimivan taajuusalueen parantamiseksi muita akustisia piirejä. Kuvassa 3 on esiteltynä kuvaillun dynaamisen mikrofonin kaaviokuva. [7] [8]



Kuva 3: Dynaamisen mikrofonin kaaviokuva. [8]

Myös mikrofoneista on olemassa monia erilaisia tyyppejä, joista kuva 3 edustaa vain yhtä. Mikrofonista riippuen vasteeseen vaikuttavat myös lämpötila, ilmanpaine ja -kosteus sekä myös niiden sisäiset ominaisuudet. Tyypilliset dynaamiset mikrofonit ovat hyviä havaitsemaan ääniä taajuuksien 40 - 16 000 Hz väliltä suhteellisen tarkasti. [7]

Mikrofoneille toinen tyypillinen ominaisuus on suuntakuvio (*engl. polar pattern*), joka kertoo mikrofonin vasteesta äänilähteen sijaintiin nähden. Yleisin suuntakuvio on pallonmuotoinen (*engl. omnidirectional*), jossa ääniaallot saapuvat mikrofonille yhtä voimakkaasti mikrofonin ympäriltä. Toinen tyypillinen kuvio on hertta (*engl. cardioid*), jossa ääni vastaanotetaan paremmin edestäpäin. Mainitut suuntakuviot esiteltynä kuvassa 4. [9]



Kuva 4: Mikrofonien tyypillisimmät pallo- ja herttasuuntakuviot. [9]

Mikrofoneissa amplitudin tallennuksesta käytetään termiä kvantisointi (*engl. quantization*) ja sitä mitataan biteissä. Yleisesti käytettyjä määriä ovat 16 ja 24 -bittiä. Lisäksi mikrofonit tallentavat ääntä eri näytteenottotaajuuksilla (*engl. sample rate*), joka taas kertoo, kuinka usein mittaus tapahtuu sekunnissa. Yleisesti käytetty näytteenottotaajuus on 41,100 kHz, jota käytetään esimerkiksi CD -levyissä. [10]

Nyquistin teoreeman mukaan näytteenottotaajuuden tulee olla kaksinkertainen verrattuna maksimi äänenkorkeuteen, jotta informaatiota ei hukata mikrofonissa. Nyquistin teoreemasta käytetään myös nimitystä Nyquist-Shannon teoreema. Aiemmin normaaliksi äänentaajuuksiksi mainittiin 20 - 20 000 Hz, joten teoreeman mukaan esimerkiksi CD -levyissä käytetty 44,100 kHz riittää tallentamaan normaalit äänentaajuudet ilman menetettyä informaatiota. [10] [11]

Nykyään käytetään myös ammattilaiskäytössä 96,000 kHz näytteenottotaajuutta, vaikka ihminen ei käytännössä pystykään huomaamaan eroa sen ja 44,100 kHz näytteenottotaajuuden kanssa. [10] [11]

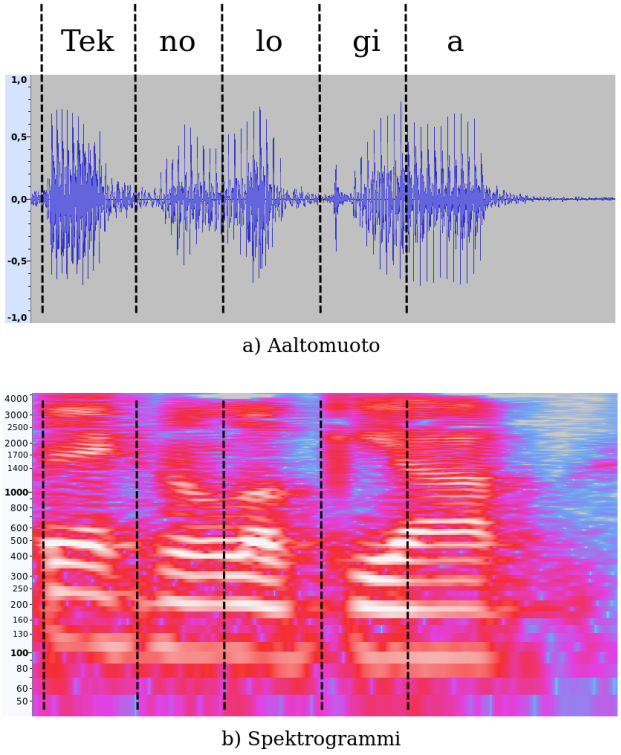
Jokaiselle mikrofonille ominaiset edellä mainitut tekijät vaikuttavat siihen millaisena ääni tallentuu. Kun kahdella eri mikrofonilla äänitetään sama lause, voi se näyttää spektrogrammeissa erilaiselta jo pelkästään mikrofonien eriävien suuntakuvioiden tai näytteenottotaajuuksien vuoksi. Tämä aiheuttaisi mahdollisesti merkittävää epäluotettavuutta puheentunnistusjärjestelmässä, jos sitä ei pyrittäisi huomioimaan millään tavalla.

Kuvan 2 mukaisesti äänisignaalia voidaan myös filtteröidä, jolla tarkoitetaan esimerkiksi äänen rajaamista kattamaan tarkemmin ihmisen kuuloalueen taajuuksia. Tämän opinnäytetyön puheentunnistusjärjestelmässä painotetaan erityisesti matalia taajuuksia, koska sillä on havaittu järjestelmien toimivuuden huomattava parantuminen esimerkiksi meluisissa ympäristöissä [12].

Filtteröinnissä voitaisiin käyttää myös erilaisia kaistasuodattimia (*engl. band-pass filter*), joiden avulla voitaisiin esimerkiksi poistaa kohinaa signaalista. Kuitenkin neuroverkot oppivat paremmin filtteröimään tiettyjä taajuuksia itse, joten kaistasuodattimien käyttö ei ole välttämätöntä.

1.1.2 Spektrogrammit

Kuvan 2 mukaisesti ääni esitetään neuroverkoille spektrogrammina. Spektrogrammit ovat tapa esittää äänidataa, joiden avulla voidaan havaita eri äänteitä ja foneemeja. Tutustutaan seuraavaksi spektrogrammeihin käymällä aluksi esimerkin avulla läpi, miltä ne näyttävät. Esimerkkinä kuvassa 5 esitettynä aalto- ja spektrogrammimuodot "teknologia" -sanasta.



a) Aaltomuoto

b) Spektrogrammi

Kuva 5: Esimerkki "Teknologia" -sanalle äänisignaalin esitystavoista. Kuvat `Audacity` -sovelluksesta.

Kuvan 5 aaltomuotoa (a) ja spektrogrammia (b) on melko yksinkertaista lukea, mutta niiden perusteella on vaikea ymmärtää sanottua. Aaltomuodossa signaalista esitetään äänenvoimakkuus ajan funktiona. Myös spektrogrammissa x-akselilla on aika, mutta y-akselilla on taajuus. Voimakkuus esiintyy spektrogrammissa värivoimakkuutena (*engl. heat map*), eli kuvassa 5b kirkkaankeltainen väri tarkoittaa vahvaa voimakkuutta tietyllä taajuudella ja ajanhetkellä.

Aaltomuodosta huomataan, että on melko helppo havaita tavujen eri ajankohdat. Spektrogrammin kuvasta taas nähdään, että jo tavujenkin sijoittaminen ajallisesti oikeaan kohtaan olisi haastavaa, sanan sisällöstä puhumattakaan.

Kuitenkin, osaava henkilö pystyy lukemaan spektrogrammista foneettisesti sen sisällön. Erilaiset fonemeenit vaikuttavat eri taajuusalueilla, jotka tuntemalla kokenut foneetikko, eli kielitieteilijä, voi lukea eri äänteet ja tavut spektrogrammista.

Nykyään spektrogrammien avulla pyritään lukemaan myös esimerkiksi henkilöiden aksentteja tai mielialoja. [13] [14] [15]

Fonemeenit ovat kielellisiä yksiköitä, joiden avulla pystytään erottamaan eri sanoja. Esimerkiksi foneemit /m/ ja /k/ erottavat sanat *matto* ja *katto* toisistaan. Useimmat kielet sisältävät noin 40 - 50 foneemia. Kansainvälinen foneettinen aakkosto (*engl. International Phonetic Alphabet*, IPA) jakaa foneemit kahteen luokkaan: vokaaleihin ja konsonantteihin. [16]

Kielitypologisesti suurimpana erona vokaalien ja konsonanttien välillä on vokaalien korkea energia, joka johtuu siitä, että kurkunpää ei ole ahtautunut niiden lausumisen aikana. Energiaerojen ansiosta spektrogrammista tunnistetut foneemit pystytään taas yhdistämään eteenpäin sanoiksi. [16]

On olemassa erilaisia spektrogrammeja, joita voidaan muodostaa eri tavoilla. Yhteistä kaikkien tapojen välillä on, että ne perustuvat Fourier -muunnoksiin (*engl. Fourier Transform*, FT), jotka matemaattisesti jakavat äänet osataajuuksiin (*engl. frequency component*). Esimerkiksi lyhytaikainen Fourier-muunnos (*engl. Short-Time Fourier Transform*, STFT) ja nopea Fourier-muunnos ovat paljon käytettyjä spektrogrammeja muodostettaessa. [16] [17]

STFT saadaan signaalille $x[n]$ ikkunoinnin avulla. Signaali jaetaan $\omega[n]$ ikkunoihin, joiden pituus on L. Ikkunoidulle osuudelle suoritetaan tämän jälkeen diskreettiaika Fourier muunnos (*engl. Discrete-time Fourier transform, DTFT*). Tämän jälkeen spektrogrammi voidaan määrittää:

$$\text{spectrogram}_k(e^{j\omega}) = |X[k, e^{j\omega}]|^2,$$

jossa $X[k, e^{j\omega}]$ on aika-riippuvainen Fourier muunnos. [16]

Tekoälyyn liittyvät asiat ovat yleisestikin kehittyneet juuri haluttuihin tarpeisiin liittyen. Usein mallia eri tekoälyn ominaisuuksiin on otettu ihmisen kehon toiminnasta, joten evoluutio on hyvä vertauskuva myös tekoälyn kehitykselle. Esimerkiksi spektrogrammeja muunnetaan puheentunnistuksessa nykyään Mel-asteikkoon (*engl. Mel-Scale*, Mel = Melody). Mel-asteikko on havainnollinen taajuusasteikko, jossa eri taajuudet ovat kuulijan mielestä yhtä kaukana toisistaan ihmisen korvalla kuunneltuna. [17] [18]

Mel-taajuusasteikon (*engl. Mel-Frequence Scale*) tarkoituksena onkin simuloida sitä, miten ihmiset kuulevat eri taajuudet. Lopputuloksena saadaan realistisempia spektrogrammeja verrattaessa lineaarisesti tuotettuihin, eli ”normaaleilla” -muunnoksilla laskettuihin, spektrogrammeihin. [18]

Mel-taajuusasteikkoon liittyen monet järjestelmät käyttävät MFC -spektreihin (*engl. Mel-Frequency Cepstral*) pohjautuvia spektrogrammeja, joka kuvaa signaalin taajuuskomponenttien voimakkuutta juurikin Mel-asteikolla. MFC:tä käytetään muodostamaan MFCC (*engl. Mel-Frequency Cepstral Coefficient*). [17] [18]

MFCC taas on audiosignaalin spektrin esitysmuoto ja se pohjautuu samaan menetelmään kuin ihmisten korvien kuuloluiden reaktiot eri taajuuksiin juurikin Mel-asteikon ansiosta. [18] [19]

Diskreetit kosinimuunnokset (*engl. Discrete Cosine Transform*, DCT) ovat usein käytössä MFCC’issä, joka toimii vähän kuin DFT, eli muuttaa signaalin aikaulottuvuudesta - taajuusulottuvuuteen, mutta ainoastaan reaalinumeroilla. Lisäksi DCT vähentää puhetaajuuksien kannalta epäolennaista informaatiota, koska reaalinumerojen myötä lopputulokset ovat tiivistetympiä. [18]

Mel-taajuusasteikkoa hyödyntävät spektrogrammit eivät ole kovin herkkiä kohinalle tai säröilylle signaaleissa. Ne ovatkin käytössä myös tämän lopputyön puheentunnistusjärjestelmässä.

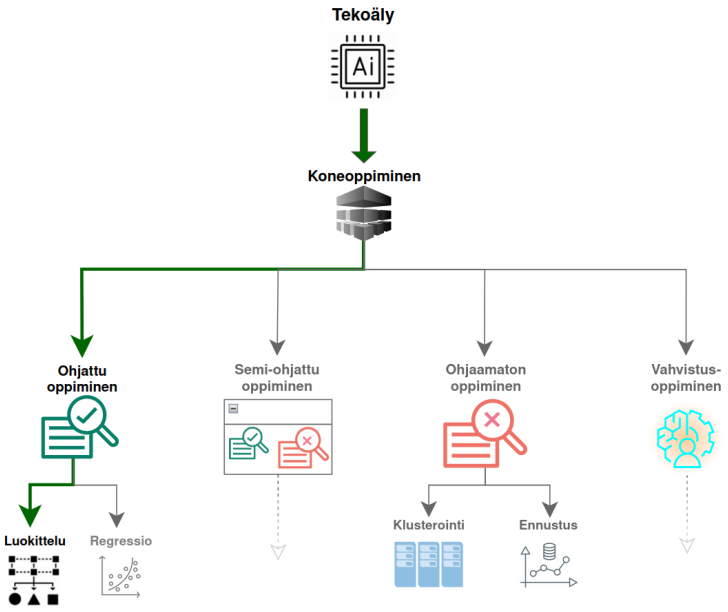
1.1.3 Tekoäly & koneoppiminen

Termille ”tekoäly” (*engl. Artificial Intelligence*, AI) ei käytännössä ole tarkkaa määritelmää. Tekoälyä on kuvattu esimerkiksi eräänlaisena kattoterminä, joka sisältää erilaisia analyysityökaluja sekä mm. kuvantunnistuksen ja puheentunnistuksen. Toinen yleinen määritelmä tekoälylle on, että tietokoneita opetetaan toimimaan ihmisen tavoin. Tekoälyä käytetään nykyään monissa eri arkipäiväisissäkin asioissa, kuten ajoneuvoissa, kohdistetussa mainonnassa sekä hakukoneissa. [16] [20]

Tarkkaa määritelmää tekoälylle on vaikea sanoa termin laajan käytön vuoksi. Nykyään tekoälyksi mielletään esimerkiksi monet menetelmät, jotka 2000-luvun alussa olisivat olleet tilastollisia malleja ja matemaattisia optimointeja. Toisaalta nykyäänkin voidaan ajatella, että kaikki tekoälymallit perustuvat ainoastaan matemaattisille periaatteille. [20]

Keskeistä tekoälylle kuitenkin on, että jotenkin sitä opetetaan toimimaan halutulla tavalla; yleensä esimerkkidatalla. Kattotermin tekoäly alapuolelle voidaan mieltää myös koneoppiminen (*engl. Machine Learning*, ML), joka perustuu tilastollisten mallien sekä algoritmien avulla datan käsittelyyn. [21]

Myös koneoppimisessa tietokoneet oppivat käsittelemään dataa älykkäästi, ja ihmisten tavoin ne pystyvät mukautumaan myös uuteen dataan. Yksinkertainen ja paljon käytetty esimerkki on roskapostiksi luokittelu uuden sähköpostin saapuessa. Koneoppiminen voidaan jakaa tekoälyn tavoin alakategorioihin, ja mainitun sähköpostin tapauksessa kyseessä olisi ohjatun oppimisen luokittelu -kategoria. [21]

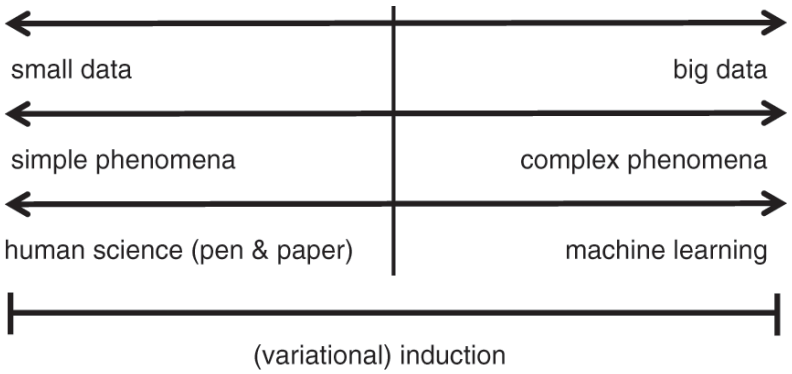


Kuva 6: Havainnekuva tekoälystä.

Kuvassa 6 esiteltynä kaaviokuva koneoppimisen rajautumisesta erilaisiin kategorioihin. Tummennetulla esitettynä tässä opinnäytetyössä keskeinen, eli ohjatun oppimisen (*engl. supervised learning*) luokittelu -kategoria (*engl. classification*). Ohjatulla oppimisella tarkoitetaan sitä, että mallia opetetaan opetusdatan avulla toimimaan tilanteesta riippuen tietyillä tavoilla. Joissakin tapauksissa tämä vaatii kuitenkin huomattavan paljon dataa. Esimerkiksi pätevälle puheentunnistusjärjestelmälle on yleisesti pidetty rajana noin 1000 tuntia monipuolista opetusdataa.

Kuten jo mainittua, teknologian kehityksen ansiosta datan säilöminen ja saatavuus sekä tiedonkäsittely ja laskentateho ovat tulleet halvemmiksi ja paremmiksi. [2]

Datan helpottuneen säilömisen ja saatavuuden myötä ollaan alettu puhua suuren datan (*engl. big data*) -aikakaudesta. Rajaa suuren ja "normaalin" datan välillä ollaan määritetty useilla eri tavoilla, kuten asettamalla suoraan kynnysrajoja sen määrälle. Wikipedian mukaan suurella datalla tarkoitetaan datamäärää, jota perinteisillä tietojenkäsittelyohjelmilla ei enää pysty käsittelemään. [22]



Kuva 7: Suuren ja pienen datamäärän ero. [22]

Pietsch et al. mukaan datamäärää voidaan pitää suurena, jos sen avulla pystytään tekemään luotettavia ennustuksia monimutkaisista asioista ja datamäärät olisivat liian suuria yksittäiselle ihmiselle käsiteltäväksi. Tämän määritelmän mukaan esimerkiksi tässä puheentunnistusjärjestelmässä käytettävää datamäärää voidaan pitää suurena datana kuvan 7 mukaisesti. [22]

Kuvassa 7 esiintyvällä vaihtelevalla induktiolla (*engl. variational induction*) tarkoitetaan havaintojoukosta luotettavien ennustuksien, toisin sanottuna yleistyksien, muodostamista. Pienellä datamäärällä yleistyksiä on vaikea luoda kompleksisista ilmiöistä. Koneoppiminen perustuu laajalti induktiiviseen päättelyyn, eli opetusdatan avulla opetetaan mallia muodostamaan yleistyksiä, joita sovelletaan uuteen dataan, esimerkiksi datan luokittelua varten. [22]

Koneoppiminen ei kuitenkaan ole mahdollistunut suuren datan aikakauden myötä, vaan sitä on jo pitkään käytetty vaihtelevissa tilanteissa monissa erilaissa tehtävissä. Suuren datan normalisoituminen ja prosessorien paraneminen ovat olleet osana mahdollistamassa koneoppimisen nopeaa kehittymistä ja laajentumista useammille, myös kompleksisten ongelmien, alueille. Lisäksi mahdollistajina ovat olleet neuroverkkorakenteiden, optimointialgoritmien sekä augmentointimenetelmien kehittyminen. [21]

Taulukko I: Koneoppia hyödyntäviä sovelluksia. [21]

Sovellus
Puheentunnistus
Hakukoneet
Käsikirjoituksen lukeminen
Esineen tunnistus kuvasta
Roskasähköpostien tunnistus
Osakemarkkinoiden analyysi
Virtuaaliavustajat (Siri, Google)
Sosiaalinen media (mm. Algoritmit)

Taulukossa I on esitelty esimerkkitehtäviä, joissa koneoppimista hyödynnetään tällä hetkellä. Useimmissa taulukon esimerkkitehtävissä koneoppimista on hyödynnetty jo pitkään. Yleisesti ottaen koneoppimista käytetään sellaisissa tilanteissa, joissa käsiteltävää dataa on todella paljon ja datan ominaisuudet muuttuvat laajasti. [21]

Puheentunnistuksen tapauksessa opetusdatalla pystytään kertomaan tekoälylle tietylle sisäänsyötölle (spektrogrammi) tietty ulossyöttö (foneemit/äänteet). Opetusdatan avulla neuroverkot optimoidaan jatkossakin tunnistamaan uudesta spektrogrammista esiintyvät foneemit, jotka koneoppimismalli myös pystyy yhdistämään sanoiksi. Tämän vuoksi opetusdatan suuruudella on merkittävä vaikutus mallin toimivuuteen; laajalla opetusdatalla saadaan koulutettua laajempi foneemi- ja sanavarasto. [21]

Opetusdatasta kerrotaan vielä tarkemmin luvussa 2, mutta seuraavassa luvussa käsitellään neuroverkkoja sekä sitä, miten ne oppivat tunnistamaan asioita, kuten spektrogrammeja.

1.1.4 Neuroverkot

Kuvassa 6 esiintyneessä ohjatussa oppimisessa koneoppimismallin neuroverkot tulee kouluttaa, jotta järjestelmä oppii toimimaan halutulla tavalla. Neuroverkot ovat siis olennainen osa puheentunnistusjärjestelmää ja ne perustuvatkin samankaltaiseen rakenteeseen kuin ihmisen aivot. Aivoissa hermosolut koostuvat useammasta peräkkäisestä neuronista, jotka kommunikoivat sähkökemiallisesti: yksittäinen neuroni prosessoi sisääntulevasta signaalista ulosmenevän signaalin, joka menee seuraavalle neuronille viejähaarakkeiden, eli aksonien, kautta. [22]

Myös koneoppimismallissa neuronit (*engl. node*) kuljettavat signaaleja x_i toistensa välillä. Painokertoimilla w_i (*engl. weights*) määritetään eri signaalien vahvuus neuronille. Signaalien tulot painokertoimien kanssa sekä vakiotermit b (*engl. bias*) summataan keskenään, jolloin saadaan aktivointiarvo Σ (*engl. activation value*). Aktivointiarvolla taas voidaan laskea aktivointifunktio f. [23]

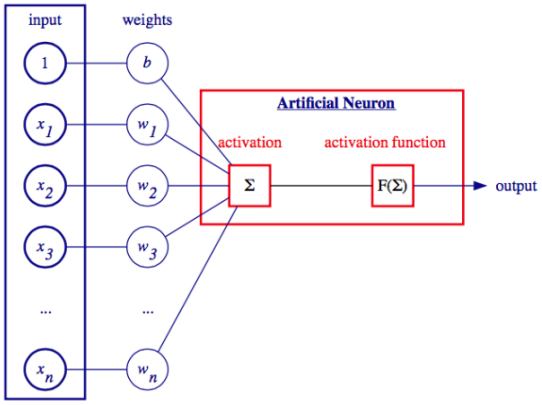
Tällöin yksittäisen neuronin yhtälöksi saadaan

$$y = f(\Sigma x_i w_i + b_i),$$

jossa y on ulossyötön arvo. Yleensä aktivointiarvo lasketaan vektorien $\vec{x}$ ja $\vec{w}$ pistetulona $y = f(\vec{x} \cdot \vec{w} + b)$.

Jotta koko neuroverkon lopputuloksena ei saataisi ainoastaan lineaarista summaa, tulee aktivointifunktioiden olla epälineaarisia. Epälineaarisia aktivointifunktiota on monia erilaisia, kuten esimerkiksi ReLU- tai Sigmoid-funktiot. Lisäksi aktivointifunktioiden tulee olla derivoituvia tai ainakin niille tulee olla määriteltyinä derivaatta eri pisteissä. Esimerkiksi ReLU -funktiolle on erikseen määritelty $f'(x = 0) = 0$. [23]

Yksittäisen neuronin toimintamenetelmää esiteltynä kuvassa 8.



Kuva 8: Yksittäisen neuronin toiminta. [23]

ReLu- ja Sigmoid-funktioden lisäksi yksi yleinen aktivointifunktio f on Softmax, jota käytetään varsinkin luokittelutehtävissä neuroverkon lopuksi.

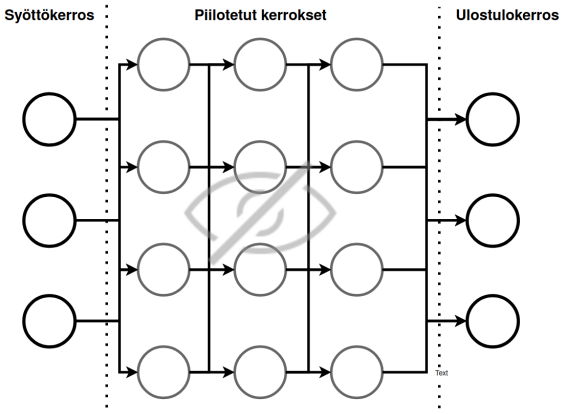
$$F(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}},$$

jossa n on mahdollisten luokkien lukumäärä ja $\Sigma y_i = 1$. Funktiosta saadaan käytännössä todennäköisyys syötteen x_i kuulumisesta luokkaan $F(x_i)$. Softmax -funktiota käytetään myös tässä lopputyössä. [23]

Neuroverkon opetuksen tarkoituksena on optimoida edellä mainittuja painokertoimia w_i. Opetukseen ja painokertoimien optimointiin tutustutaan tarkemmin luvussa 1.1.5. [20] [21]

Kuvassa 9 on havainnollistettu yksinkertaisen eteenpäinkytketyn neuroverkon rakennetta. Yksittäiset neuronit muodostavat neuroverkon, joka koostuu kolmesta erityyppisestä kerroksesta: sisääntulokerroksesta (*engl. input layer*), piilokerroksista (*engl. hidden layers*) sekä ulostulokerroksesta (*engl. output layer*). [21]

Piilokerroksien määrä vaihtelee, mutta yksinkertaisissa neuroverkoissa niitä on noin kolme kappaletta. Kuitenkin esimerkiksi syväoppivissa verkoissa piilokerroksia voi olla laskentatavasta riippuen jopa satoja. [20]



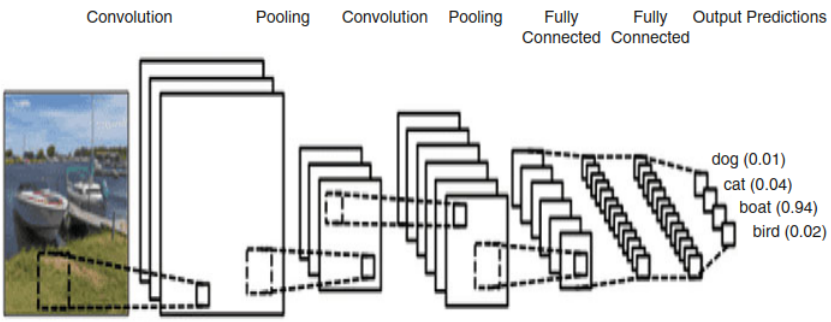
Kuva 9: Havainnekuva yksinkertaisesta eteenpäinkytketystä neuroverkosta.

Ulostulokerroksien määrä taas riippuu neuroverkon tehtävästä, kuten esimerkiksi ohjatun oppimisen luokittelu -tyyppisessä neuroverkossa lopputuloksien mahdollisuudesta; jos neuroverkon olisi tarkoitus tunnistaa sille syötetyt numerot 0-9 kuvista olisi ulostulokerroksia silloin 10 kappaletta. Tällöin kyse on ns. lokeroidusta aineistosta. [23]

Neuroverkkojen kerrokset ovat kytkettyinä toisiinsa ja esimerkiksi eteenpäinkytketyissä (myös nimellä myötäkytketty) neuroverkoissa (*engl. FeedForward Neural Network*, FFN tai FFNN) tieto liikkuu kerroksittain eteenpäin neuronilta-neuronille. [21]

Jotta verkkorakenne olisi yksinkertaisemmin esitettynä, ei yksittäisille neuroneille ole piirretty kuvassa 9 vakiotermejä b tai painokertoimia ω, mutta jokaiselle neuronille saapuvaan signaaliin nämä silti vaikuttavat.

Eteenpäinkytketty neuroverkko oli vain yksi esimerkki. Neuroverkoilla on eri arkkitehtuureja, jotka eroavat toisistaan mm. kerroksien kytkentätavoissa. Konvoluutioneuroverkot (*engl. Convolutional Neural Network*, CNN) ovat laajalti käytettyjä syväoppivissa neuroverkoissa. CNN on käytössä tyypillisesti kuvien analysoinnissa ja käytännössä puheentunnistuksen spektrogrammikin on kaksiulotteinen kuva. Konvoluutioverkon rakenne esiteltynä kuvassa 10. [20]



Kuva 10: Konvoluutioverkkojen rakenne. [21]

Myös konvoluutioverkoissa toimintatapa perustuu samankaltaisiin asioihin, kuin ihmisillä. Ihmisen näössä yksittäiset neuronit keskittyvät pieneen osaan näkökenttää. Neuronien havainnot yhdistetään, jolloin muodostuu käsitys nähdystä kuvasta. [21]

Konvoluutioverkot jakavat kuvan laatikoihin, jotka se käy läpi yksitellen. Jokaisesta laatikosta saadut piirteet syötetään konvoluutiokerrokselle. Konvoluutiokerroksissa kernelit suorittavat kuville konvoluutioita, eli lineaarisia operaatioita. Yksinkertaistettuna kernelit vähentävät kuvien analysointiin tarvittavien laskujen määrää kuitenkaan menettämättä niistä hyödyllistä informaatiota. [21]

Yksi pooling -kerroksien tarkoituksista onkin vähentää neuroverkoilla kulkevan datan määrää. Kuvan 10 oikeassa laidassa voidaan havaita myös lopputuloksena ulostulon ennusteet, eli tässä tapauksessa todennäköisyydet asialle, joka sisäänsyötetyssä kuvassa esiintyy. [21]

Konvoluutioverkkojen etuna on esikäsittelyn vähäinen tarve ja datan helpommin prosessoitavaan muotoon laittaminen, joka on käytännöllistä varsinkin korkearesoluutioisissa kuvissa tarvittavan suuren laskentakapasiteetin vähentämiseksi.

Tarkemmat koneoppimismallin käyttötarkoitukset vaikuttavat siihen, millaista arkkitehtuuria on järkevää käyttää. Esimerkiksi Kumar et al. käyttivät ns. BiRNN -arkkitehtuuria (*Bidirectional Recurrent Neural Network*) tunnistamaan tekoälyn luomia syväväärennös (*engl. deepfake*) videoita. [1]

Myös puheentunnistusjärjestelmiä itsessään on jatkokehitetty moniin eri, todella yksityiskohtaisiinkin, käyttötarkoituksiin, kuten esimerkiksi tunnistamaan NATO -aakkosia lentoliikenteessä. [24]

Konvoluutioverkkojen lisäksi puheentunnistuksessa laajasti käytettyjä arkkitehtuureja ovat olleet esimerkiksi residuaaliverkot (*engl. Residual Network, ResNet*) sekä ns. transformerit. [21] [25] [26]

Residuaaliverkot kehitettiin ratkaisemaan monikerroksisten syväoppivien verkkojen katoavan gradientin ongelmaa. Katoavan gradientin välttämiseksi residuaaliverkot hyödyntävät ns. oikotie-yhteyksiä (*engl. shortcut-connections*), jotka yhdistävät piilokerroksia, jotka eivät ole päällekkäin, ohittamalla yhden tai useamman kerroksen. Tämä realisoi aktivointifunktiolle meneviä syötteitä. Residuaaliverkot ovat edelleen laajassa käytössä monissa menetelmissä, kuten kuvantunnistuksessa. [26]

Transformer -arkkitehtuureissa, johon esimerkiksi suosittu Chat-GPT (*engl. Generative Pre-training Transformer*) perustuu, olennaisia asioita ovat koodaaja (*engl. encoder*) sekä purkaja (*engl. decoder*), jotka toimivat yhdessä ns. attention-mekanismin (*engl. attention mechanism*) avulla. Koodaaja kartoittaa syötteitä ja purkaja ulostuloja, joita mekanismi vertailee ja yhdistelee keskenään. [27]

Puheentunnistusjärjestelmissä käytettävistä arkkitehtuureista suosiotaan ovat kasvattaneet myös conformerit (*engl. conformer*). Käytännössä conformerit koostuvat konvoluutio- sekä transformerverkkojen ominaisuuksien yhdistelmästä, josta nimikin tulee. Conformereilla on saavutettu jopa suhteessa 15% parempia lopputuloksia edeltäviin parhaisiin transformer -arkkitehtuureihin nähden. [25]

Suosiostaan johtuen conformereita on tutkittu yleisesti ASR -järjestelmissä sekä tarkemmin määritellyissä käyttökohteissa, kuten lennonjohtotorneissa. Tämän lopputyön järjestelmässä käytetään conformer-arkkitehtuuria. [28] [29] [30]

1.1.5 Neuroverkkojen opettaminen

Neuroverkot opetetaan optimoimalla niiden painokertoimia opetusdatan avulla. Kun opetusdatan syötteet ovat menneet kaikkien piilokerroksien aktivointifunktioiden läpi saadaan tuloksena jokin ulossyöttö. Tätä tulosta verrataan annotoidussa opetusdatassa haluttuun oikeaan tulokseen ja sen perusteella lasketaan virhefunktio (myös nimellä hukkafunktio) (*engl. error function* tai *loss function*), joka pyritään minimoimaan erilaisilla menetelmillä.

Virhefunktio voidaan määrittää eri tavoilla, kuten MSE:n (*engl. Mean Squared Error*), MAE:n (*Mean Absolute Error*), MPE:n (*Mean Percentage Error*) tai CTC häviön (*engl. Connectionist Temporal Classification Loss*) avulla. [31]

Tämän työn koneoppimismallin virhefunktiona käytetään CTC:tä, joka laskee esimerkiksi sigmoid-aktivointifunktioiden antamien todennäköisyyksien avulla yhteistodennäköisyydet eri kirjainyhdistelmille, joiden kautta taas voidaan laskea arvo virhefunktiolle $\mathrm{CTC}(\mathbf{l}, \mathbf{x}) = -\ln p(\mathbf{l}|\mathbf{x})$. [31] [32]

Saadusta virhefunktiosta tulee seuraavaksi löytää minimi. Yleisesti matematiikassa käytetään useamman muuttujan funktion ääriarvojen löytämiseen gradienttia, joka laskettaisiin kolmiulotteiselle tapaukselle

$$\nabla f(x,y,z) = \frac{\partial}{\partial x} f(x,y,z)\vec{i} + \frac{\partial}{\partial y} f(x,y,z)\vec{j} + \frac{\partial}{\partial z} f(x,y,z)\vec{k}.$$

Gradientti kertoo funktion nopeimman kasvun suunnan osittaisderivaattojen avulla, joten virhefunktion nopeimman vähenemisen suunta saadaan yksinkertaisesti gradientin vastakkaisesta suunnasta. [23]

Virhefunktion minimointiin käytetään pääsääntöisesti stokastista gradienttia (*engl. Stochastic Gradient Descent*, SGD), joka eroaa normaalista gradientista sillä, että se lasketaan ainoastaan satunnaisille arvoille datasta, josta termi ”stokastinen” tuleekin. SGD:n laskeminen on huomattavasti normaalia gradienttia nopeampaa, koska

laskemisen määrä vähenee. [23]

Seuraavaksi virhefunktion avulla voidaan optimoida ω ja b -parametreja. Myös parametrien optimointi voidaan suorittaa useilla eri tavoilla. Suosittu tapa gradienttien laskemiseksi optimoimiseen on vastavirta-algoritmi (*engl. backpropagation*). Vastavirta -termi tulee siitä, että painokertoimia ω ja vakiotermejä b lähdetään purkamaan ulostulokerroksen kautta, kerros-kerrokselta, "väärään" suuntaan. Virhefunktion laskemista ja optimointia voidaan tehdä, kun tietty määrä opetusdataa on käyty läpi, satunnaisesti tai jokaisen opetusyksikön jälkeen. [23]

Jos neuroverkossa on useita piilokerroksia, tulee välttää ns. katoavan gradientin (*engl. vanishing gradient*) -ongelmaa. Tällöin gradientti heikkenee niin, että ensimmäisien piilokerroksien painokertoimet eivät muutu lainkaan, joka taas heikentää neuroverkon oppimista. Katoavan gradientin ongelmaan on kehitetty monia erilaisia ratkaisuja, kuten esimerkiksi LSTM (*engl. Long Short-Term Memory*) tai jo edellisessä luvussa mainittu residuaaliverkko. [23]

Edellä mainittu painokertoimen laskenta vaatii laitteistolta merkittävää kapasiteettia, jotta malli voidaan opettaa kohtuullisessa ajassa. Laskentatehoa vaativa asia on neuroverkkojen opettamisessa käytetty numeerinen optimointi. [21]

Prosesssoreille voidaan määrittää kuinka monta opetusesimerkkiä ne pystyvät keskiarvolta opettamaan mallille per sekunti. Vertailun vuoksi alapuolella on esitettynä kaksi esimerkkiä mallin opettamisesta, joiden avulla havainnollistetaan graafisen laskentakyvyn tärkeyttä. On tärkeää kuitenkin muistaa, että prosessorien keskinäinen opetusnopeuden vertailu ei kerro koko totuutta. Laskuesimerkeissä käytetään fiktiivistä 10 000 kappaleen suuruista opetusdataa. [20]

CPU (*engl. Central Processing Unit*) prosessorit sopivat laskennallisesti pienemmille koneoppimismalleille. Esimerkkinä Intelin i7-7500U, jolle Gupta et al. kertovat opetusnopeuden olevan keskiarvoltaan 115 esimerkkiä / sekunti [21]. Näin ollen 10 000 kappaleen opetusdataan kuluisi aikaa:

$$\frac{10\ 000\,\text{kpl}}{115\,\text{kpl/s}} \approx 86,95\,\text{s} > 1\,\text{min},$$

GPU (*engl. Graphics Processing Unit*) taas on laskentatehoa vaativille tehtäville nopeuden suhteen merkittävä tekijä. Gupta et al. ilmoittavat Nvidian GTX 1080 grafiikkasuorittimille koulutusnopeuden keskiarvoksi 14 000 esimerkkiä/sekunti:

$$\frac{10\ 000\,\text{kpl}}{14000\,\text{kpl/s}} \approx 0,71\,\text{s}.\ [21]$$

Tässä esimerkissä GPU:n ollessa yli 120 kertaa nopeampi voidaan todeta prosessorin valinnan olevan merkittävä jo 10 000 kappaleen opetusdatalla. On kuitenkin myös huomattava, että opetusnopeus riippuu paljon esimerkiksi käytetyn neuroverkon arkkitehtuurista tai opetusdatan muodosta; kuva- ja äänidatalle neuroverkot tekevät paljon matriisilaskentaa, joiden rinnakkaisen käsittelyn GPU -prosessorit mahdollistavat. [21].

Epookki (*engl. epoch*) on koneoppimisessa keskeinen termi. Yleisesti epookilla tarkoitetaan jotain ajankohtaa, josta uusi ajanjakso alkaa. Koneoppimisessa yhdeksi epookiksi kutsutaan yleensä sitä, kun koko opetusdata on kertaalleen mennyt neuroverkkojen läpi. Tilanteesta riippuen mallille voidaan opettaa useita satojakin kertoja koko opetusdata läpi, eli satoja epookkeja. [23]

Tässä lopputyössä epookki on kuitenkin vakioitu opetusaskelien pituuteen, josta kerrotaan vielä tarkemmin luvussa 5.

1.2 Puheentunnistusjärjestelmän optimointi

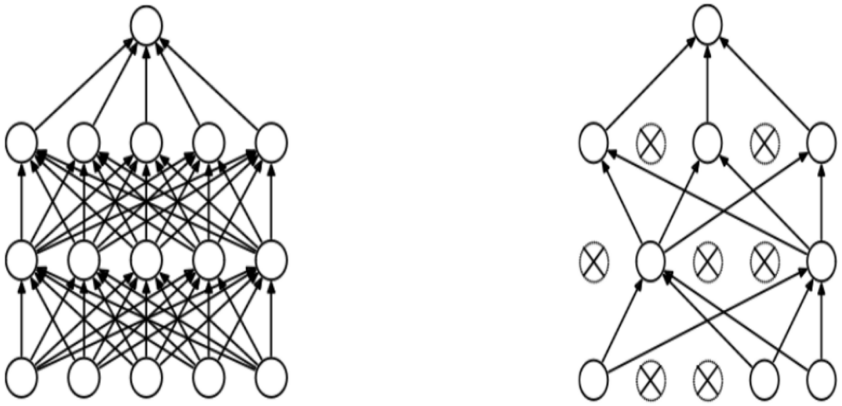
Opetusdatan avulla koneoppimismalli siis oppii lukemaan foneemien piirteitä uusista spektrogrammeista. Järjestelmän robustisuutta pyritään kuitenkin vielä parantamaan erilaisilla optimointialgoritmeilla.

Jokaista suomen kielen sanaa ei ole tärkeää saada ääninäytteeksi opetusdataan, vaan tärkeämpää on saada mahdollisimman kattavasti eri foneemeja ja kirjainyhdistelmiä. Koneoppimismalli oppii luomaan eri sanoja ja jopa havaitsemaan tiettyjä kielioppityylejä opetusdatan avulla.

Jotta ääninäytteissä puhuttu saadaan mahdollisimman tarkasti juuri sellaisenaan tekstiksi, ei tässä opinnäytetyössä ole erikseen opetettu suomen kielen sanastoa neuroverkoille. Tämä mahdollistaa myös sen, että mahdolliset uudet slangi- ja murresanat saadaan myös ulossyötteenä juuri siten, kun ne on sanottu.

Myös todennäköisyyksiä esimerkiksi mahdollisista sanoista voidaan arvioida erilaisilla menetelmillä robustisuuden parantamiseksi, kuten kielimallien avulla: jos lopputuloksen todennäköisyydeksi on saatu kaksi sanaa: *tuoli* (50%) ja *huoli* (50%), voidaan tutkia kumpi sanoista on todennäköisempi muihin tunnistettuihin sanoihin liittyvässä asiayhteydessä. Jos muutkin edellä tunnistetut sanat liittyvät huonekaluihin, on *tuoli* todennäköisempi vaihtoehto. [21]

Opetuksessa yleinen ongelma on neuroverkkojen ylisovitus opetusdataan. Myös tätä pyritään estämään useilla erilaisilla menetelmillä, kuten esimerkiksi neuronien satunnaisella poistamisella verkosta, eli dropout:illa, (*engl. dropout*), jolloin poistettujen neuronien kautta menevät signaalit häviävät opetuksen ajaksi. Myös dropout on stokastinen menetelmä satunnaisuutensa vuoksi. [33]



(a) Normaali neuroverkko. (b) Neuroverkko dropout:in kanssa.

Kuva 11: Havainnekuva dropout:ista. [33]

Kuvassa 11 on havainnollistettuna neuroverkkojen dropout:ia, jota käytetään myös tämän opinnäytetyön puheentunnistusjärjestelmässä. Tällä pyritään estämään mahdollista ylisovittamista, joka onkin yksi keskeisistä ongelmista neuroverkkojen opetuksessa. [33]

Kuvassa 11a on kuvattuna eteenpäinkytketty neuroverkko, jossa edellisen kerroksen kaikki neuronit yhdistyvät seuraavan kerroksen jokaiseen neuroniin. Kuvassa 11b on arkkitehtuuriltaan samantyyppinen verkko, jossa osa yksittäisistä neuroneista on pudotettu pois. Pudotettujen neuronien osalta kaikki niihin kohdistuvat laskennat jätetään siis suorittamatta, jolloin kokonaiskytkentöjen määrä vähenee. [33]

Laskennan määrän vähenemisen ansiosta eksponentiaalisen monta eri neuroverkkoa pystytään yhdistämään toisiinsa. Tätä kutsutaan yhdistelmätoiminnolliseksi oppimisalgoritmiksi (*engl. ensemble learning*). Dropout:in on myös havaittu pienentävän häviöfunktion arvoa testijoukossa. [33]

Neuroverkkojen kehittyneistä arkkitehtuureista ja optimointimenetelmistä huolimatta puheentunnistusjärjestelmän robustisuutta tulee testata sen toimivuuden varmistamiseksi, josta kerrotaan seuraavassa luvussa 1.3.

1.3 Opetetun puheentunnistusjärjestelmän testaaminen

Opetetun puheentunnistusjärjestelmän robustisuutta tulee myös evaluoida, eli testata, jotta tiedetään kuinka hyvin käytetyt augmentointi- tai optimointimenetelmät toimivat. Robustisuudella tarkoitetaan tehokasta, oikeita lopputuloksia antavaa ja luotettavasti toimivaa järjestelmää. Järjestelmää voidaan pitää robustisena, vaikka saadut lopputulokset poikkeaisivatkin vähän syötteestä.

Myös evaluointiin tarvitaan ennalta annotoitua dataa, jota ei voida käyttää opetuksessa. Alkuperäisestä opetusdatasta valitaan satunnainen osa, jota käytetään pelkästään testidatana (evaluointidata). Luotettavan tuloksen saamiseksi testidatan määrä pitää olla tarpeeksi suuri, joka kasvattaa entisestään tarvittavan annotoidun datan määrää.

Puheentunnistusjärjestelmien evaluoimiseen *de facto* käytetty menetelmä on WER (*engl. Word Error Rate*), joka saadaan yhteenlaskettujen virheiden normalisoidulla summalla. Normalisointi tapahtuu jakamalla virheet sanojen referenssimäärällä N_r

$$\mathrm{WER} = \frac{S + I + D}{N_r},$$

jossa S on korvattavien (substitutions), I lisättävien (insertions) ja D poistettavien (deletions) sanojen lukumäärä. Tyypilliset puheentunnistusjärjestelmät pääsevät noin 5% WER -tulokseen. [34]

WER:issä pienet, yhden kirjaimen, virheet vaikuttavat yhtä paljon WER -arvoon kuin isommat monen kirjaimen virheet saman sanan sisällä. Esimerkiksi, jos sana ”kuka” on tunnistettu sanaksi ”kukka” on se WER:in mukaan yhtä paha virhe kuin tunnistaa sanaksi ”kuljettaja”.

Lisäksi yleensä asiakonteksteissa toiset sanat ovat tärkeämpiä kuin toiset, joten tietyissä sanoissa virheiden vaikutus järjestelmän luotettavuusarvioon tulisi korostua. Edellä mainituista syistä johtuen evaluoimiseen on kehitetty myös muita me-

netelmiä, kuten CER (*engl. Charatecter Error Rate*), TER (*engl. Term Error Rate*) tai Somnath et al. käyttämä ns. semanttinen-WER (SWER).

SWER painottaa WER:in muuttujia S, I ja D niille ominaisilla painokertoimilla W_{S}, W_{I} sekä W_{D}. Lisäksi menetelmässä huomioidaan sanojen tärkeys painokertoimella IW (*engl. Importance Weight*). [34]

SWER saadaan laskettua yhtälöstä

$$\mathrm{SWER} = \mathrm{score}_a + IW \cdot DW \qquad || \; DW = \frac{\mathrm{accuracy}}{N_r - \#E_{(NE \cup SENT)}}$$

$$\mathrm{SWER} = \mathrm{score}_a + IW \cdot \frac{\mathrm{accuracy}}{N_r - \#E_{(NE \cup SENT)}} \qquad || \; \mathrm{accuracy} = 1 - \mathrm{score}_a$$

$$\mathrm{SWER} = \mathrm{score}_a + IW \cdot \frac{1 - \mathrm{score}_a}{N_r - \#E_{(NE \cup SENT)}} \qquad || \; \mathrm{score}_a = \sum W_{\mathrm{S}}S + W_{\mathrm{I}}I + W_{\mathrm{D}}D$$

$$\Rightarrow \; \mathrm{SWER} = \sum (\, W_{\mathrm{S}}S + W_{\mathrm{I}}I + W_{\mathrm{D}}D) + IW \cdot \frac{1 - \mathrm{score}_a}{N_r - \#E_{(NE \cup SENT)}},$$

jossa $\#E_{(NE \cup SENT)}$ on väärien sanojen lukumäärä. [34]

Somnathin esimerkin mukaan lauseesta ”ram loves sita” saatu ”ram love sita” antaisi WER tuloksen 0.33, mutta SWER:in tulos olisi 0.00, joka olisi tässä tapauksessa realistisempi kuvaamaan järjestelmän robustisuutta. [34]

Puheentunnistuksessa on yleistä käyttää WER:iä evaluointiin, mutta foneettisesti suoraviivaisessa tunnistuksessä edellä mainitut menetelmät eivät suoraan kuvasta järjestelmän robustisuutta. Lisäksi taivutuksellisesti rikkaassa suomen kielessä puheentunnistusjärjestelmät antavat herkästi vääriä sanamuotoja.

Tässä lopputyössä mallien evaluointiin käytetään Levenshteinin editointietäisyyttä, joka on melko lähellä aiemmin mainittua CER:iä. Levenshteinin etäisyydessä lasketaan kahden lauseen ero yksinkertaisesti tarvittavien merkkimuutosten avulla. Edellä mainitulle esimerkille ”ram love sita” Levenshtein antaisi siis arvon 1. Opetuksen aikaisen virhefunktion laskemiseen käytetään kuitenkin CTC:tä. Levenshteinin etäisyys on aina 0 tai positiivinen kokonaisluku $\mathbb{Z}_+$. [35]

2 Opetusdata

Vaikka valmiiksi annotoitua dataa käytetään sekä mallin opettamiseen että testaamiseen, esitetään se tässä opinnäytetyössä yleisesti aina opetusdatana. Kuten aiemmin mainittiin, opetusmateriaalia, eli opetusdataa, muodostetaan kahdella tavalla; uutta dataa lisäämällä tai olemassa olevaa augmentoimalla.

Tässä luvussa käydään läpi, mitä uuden datan lisäämisellä, luku 2.1, sekä olemassa olevan datan augmentoinnilla, luku 2.2, tarkoitetaan. Luvussa 2.3 vertaillaan, miten opetusdatan määrä vaikuttaa järjestelmän robustisuuteen.

2.1 Uuden datan lisääminen

Yleisesti ottaen paras tapa kehittää ohjatun oppimisen tarkkuutta ja opetettavien neuroverkkojen malleja on uuden hyvälaatuisen opetusdatan lisääminen. Uusi opetusdata täytyy keräämisen lisäksi kuitenkin vielä annotoida, eli äänidatan tapauksessa litteroida. [2]

Esimerkiksi kuvantunnistuksen tapauksessa uutta dataa voitaisiin siis käytännössä lisätä ottamalla valokuvia, jonka jälkeen ne annotoitaisiin sen perusteella, mitä kuvassa esiintyy. Lopuksi annotoidut kuvat lisättäisiin opetusdataan. Datan kerääminen ja annotointi ovat kuitenkin hitaita sekä kalliita prosesseja.

Ilmaista, avoimista lähteistä löytyvää, opetusdataa on toki saatavilla tiettyjä koneoppimismenetelmiä varten, kuten esimerkiksi eläinten tunnistamiseen kuvista. Yritysten olisi kuitenkin vaikea myydä tuotteena koneoppimismallia, joka perustuu ainoastaan avoimista lähteistä saatuun opetusdataan, jonka vuoksi kaikkea dataa ei ole avoimesti saatavana. Esimerkiksi suomen kielen puheentunnistuksen tapauksessa saatavilla olevaa opetusdataa olisi niukasti saatavilla toimivan järjestelmän luomiseksi.

Koneoppimisen yleistymisen, ja tätä kautta juuri oikeanlaisen opetusdatan tarvitsemisen, myötä on tullut myös lukuisia yrityksiä, jotka myyvät palveluitaan datan annotointia varten. Tämän avulla suuriakin määriä dataa voidaan toki annotoida nopeasti, mutta kustannukset kasvaisivat silti, eikä annotoidun datan laadusta voisi olla varma. [36]

2.1.1 Äänidatan kerääminen ja annotointi

Kun kyse on puheentunnistusjärjestelmän opetusdatasta, olisi tärkeää saada järjestelmälle opetettua mahdollisimman laaja ja kattava kokoelma ääninäytteitä. Erilaiset ääninäytteet antavat eri määrän informaatiota. Opetusdataa voidaan luoda alusta asti myös itse, mutta sanojen erikseen lausuminen, nauhoittaminen, leikkaaminen ja annotoiminen on työlästä. Tätä työmäärää saadaan vähennettyä käyttämällä jo valmiiksi äänitettyä dataa.

Jos valmiiksi äänitetty data on sattumanvaraisesti sanottua, ei voida olla varmoja, että sanakirjan jokainen sana saadaan opetusdatalle. Tässä kohtaa onkin käytännöllistä, että puheentunnistusjärjestelmä ei tunnista sille opetettuja sanoja, vaan niiden foneemeja sekä äänteitä, jotka se voi yhdistää uudeksi sanaksi. Lisäksi tarpeeksi suurella sattumanvaraisesti valitulla datalla saadaan eniten yleisimpiä foneemeja, joita todennäköisesti myös tunnistettavassa datassa esiintyy eniten.

Tässä puheentunnistusjärjestelmässä käytetään muutamien eri podcastien äänitteitä, joiden käyttämiseen on pyydetty tekijöiden suostumus. Näin saatiin noin tuhannen tunnin edestä sattumanvaraista äänidataa valmiiksi nauhoitettuna. Käytettyjä ovat mm. jotkin podcastit, kuten Futucast, Rahapodi sekä Puheenaihe. Tämäkin äänidata tulee toki annotoida, ja kuten todettua, datan annotoiminen on hidasta.

Kesällä 2022 kahdeksan kesätyöntekijää tekivät datan annotointia puheentunnistusjärjestelmää varten. Annotoitavat datat oltiin valmiiksi leikattu podcasteista ja annotointia varten oli tehty sovellus, jossa työntekijän tuli kirjoittaa sanatarkasti se, mitä hän ääninäytteessä kuuli. Lisäksi ohjelmassa merkittiin suoritetun annotoinnin varmuus asteikolla 1-5.

Kesän aikana työntekijät saivat annotoitua varmalla asteikolla (varmuus ≥ 3) noin 20 000 kpl ääninäytteitä. Työtunteja tähän kului heiltä yhteensä noin 810. Keskimäärin yksi työntekijä pystyi siis annotoimaan tunnissa ääninäytteitä noin:

$$\frac{20\ 000\,\text{kpl}}{810\,\text{h}} \approx 24,69\,\frac{\text{kpl}}{\text{h}}.$$

Laskettu tulos (kappaletta tunnissa) riippuu suuresti ääninäytteiden keskimääräisestä pituudesta. Lisäksi annotointia tehtiin nyt laatu edellä; annotoinnissa ei pidetty kiirettä ja epävarmaksi arvioitua dataa hylättiin. Näiden tekijöiden vuoksi lukua ei ole järkevää suoraan vertailla toisiin annotointeihin.

Lasketun tuloksen perusteella voidaan todeta annotoinnin olevan melko hidasta. Lisäksi annotoiminen työtehtävänä on melko yksitoikkoista, joten annotointinopeus luultavasti hidastuisi entisestään pitkän ajan kuluessa. Data tuli myös vielä jälkikäsitellä, jossa ensimmäiseksi hylättiin huonolla varmuudella annotoitu data. Varmuudella 3 tai yli annotoitu data syötettiin valmiiseen puheentunnistusjärjestelmään ja ulossyöttöä, eli sitä mitä äänessä sanottiin järjestelmän mukaan, verrattiin annotoinnissa määritettyyn tekstiin. Erilaiset tulokset saanut data käytiin manuaalisesti läpi, jolloin tarkistettiin onko virhettä esiintynyt järjestelmän vai annotoijan tulkinnassa.

Annotoinnissa on yleisestikin hyvä käyttää puheentunnistusjärjestelmän malliksi ehdottamaa litterointia syötteelle. Tämä nopeuttaa annotointia huomattavasti.

2.2 Augmentointi

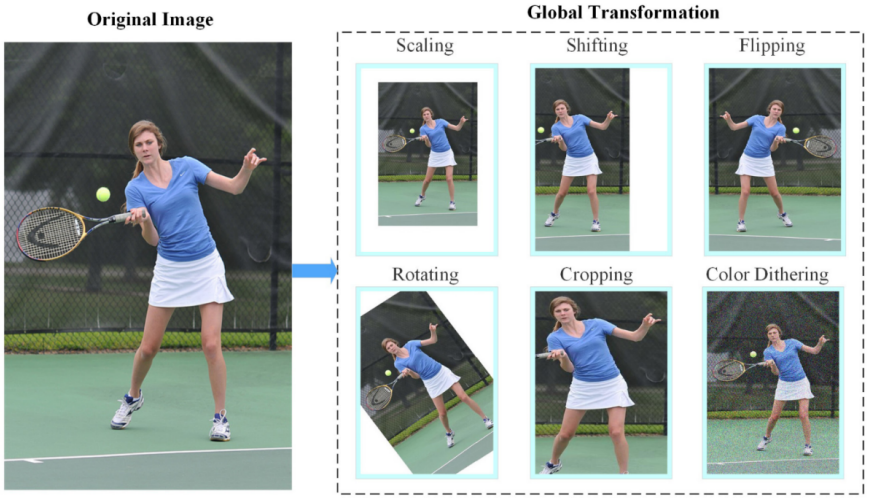
Opetusdataa lisäämällä saadaan paranneltua optimoitavien neuroverkkojen mallia, mutta ihmisen tekemänä uuden datan annotoiminen on kohtuuttoman hidasta ja kallista. Dataa voidaan kuitenkin lisätä myös toisella tavalla; muokkaamalla jo valmiiksi annotoitua dataa. Tällöin jokaisesta ihmisen annotoimasta opetusesimerkistä pyritään saamaan maksimaalinen hyöty irti tekemällä siitä muunnoksia, jotka eivät kuitenkaan muuta lopputulosta. Tätä prosessia kutsutaan augmentoimiseksi tai data-augmentoinniksi. [2] [37] [38] [39]

Augmentoinnilla on kustannuksia vähentävä vaikutus ja sen avulla opetettujen mallien on todettu olevan myös merkittävästi robustimpia. Augmentointia apuna käyttäen opetetut mallit ovat vähemmän herkkiä syötteen pienille muutoksille, jonka ansiosta myös ylisovitus vähenee. [2]

Shorten & Khoshgoftaar vertaavat datan augmentointia yöllä nähtäviin uniin: Ihmiset kuvittelevat jonkinlaisia muunnelmia tapahtumista, jotka perustuvat todellisiin kokemuksiin. Samalla tavoin voidaan ajatella, että augmentoitu data on jonkinnäköinen muunnelma alkuperäisestä datasta. [2] [39]

Opetusdatan lisäämisen ohella augmentointi on myös tärkeä työkalu, kun koneoppimismallin toimivuutta halutaan parantaa suoraan tietynlaisissa ympäristöissä. Esimerkiksi taustahälyisissä tiloissa mallia voidaan opettaa suoraan tietynlaisille esimerkeille. Augmentoinnista saatavan hyödyn maksimoimiseksi onkin tärkeää, että augmentoituakin dataa voitaisiin ainakin olettaa esiintyvän lopullisessa käyttökohteessa.

Itse augmentointimenetelmät riippuvat datatyypistä, jota augmentoidaan. Augmentointia on helppo havainnollistaa kuvantunnistuksen tapauksessa, josta esimerkki alapuolella kuvassa 12. [38]



Kuva 12: Kuvantunnistuksessa käytettäviä augmentointitapoja. [38]

Kuvantunnistuksessa augmentointimenetelmät voivat pitää sisällään esimerkiksi alkuperäisen kuvan sumentamista, terävöittämistä, zoomaamista, värien vääristämistä, aspektisuhteen muutoksia, kääntämistä peilikuvaksi tai pieniä kiertoja. Kuvan 12 tapauksessa Wang et al. augmentoivat alkuperäistä kuvaa, jotta tekoäly oppisi tunnistamaan ihmisten asentoja kuvista. [38] [39]

Toinen esimerkki augmentaatiotapojen moninaisuudesta voisi olla luonnollisen kielen tapauksessa, jossa menetelmiä ovat mm. kirjoitusvirheiden lisääminen, sanojen vaihtaminen synonyymeiksi sekä kirjakielen muokkaaminen slangiksi tai murteelliseksi.

Augmentointi on siis asiantuntijatyötä; myös augmentoidun datan tulee olla järkevää ja vastata alkuperäistä dataa. Käytettävät augmentointiomenetelmät taas riippuvat täysin varioitavasta datatyypistä.

Esimerkiksi tekstintunnistamisen tapauksessa peilikuvan ottaminen opetusdatasta tai puheentunnistusjärjestelmän tapauksessa spektrogrammien liiallinen muokkaaminen ei enää antaisi oikeanlaista opetusdataa neuroverkoille. Seuraavassa luvussa käydäänkin läpi käytettyjä augmentointimenetelmiä puheentunnistusjärjestelmälle, eli äänidatalle.

2.2.1 Äänidatan augmentointi

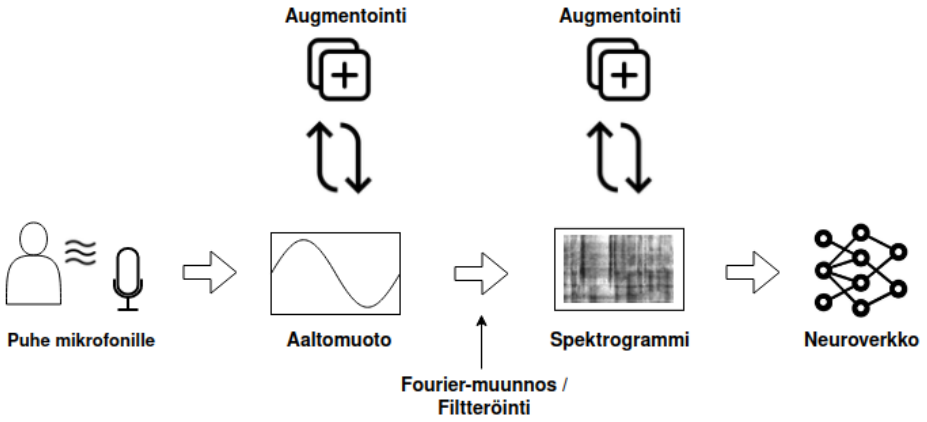
Äänidatan augmentointimenetelmät voidaan jakaa kolmeen eri luokkaan niissä mallinnetun aiheuttajan perusteella. Äänen variaatioissa aiheuttajia ovat ympäristö, puhuja sekä laitteisto. Tässä opinnäytetyössä keskitytään ympäristön ja laitteiston mukaisiin augmentointitapoihin: kaikuihin, taustameluun, säröilyyn sekä taajuusvasteiden variointeihin.

Alapuolella taulukossa VIII esiteltynä paljon yleisesti käytettyjä äänidatan augmentointitapoja sekä niiden pääsääntöinen aiheuttaja. Alleviivattuina ovat tässä opinnäytetyössä tarkasteltavat augmentointitavat. [37]

Taulukko II: Äänisignaalin augmentointitapoja [37]

Augmentointitapa	**Pääsääntöinen aiheuttaja**
Kaikujen lisääminen	Ympäristö
Taustamelun lisääminen	Ympäristö
Volyymin muuttaminen	Puhuja
Äänenkorkeuden/nopeuden muokkaaminen	Puhuja
Säröilyjen lisääminen	Laitteisto
Taajuusvasteiden variointi	Laitteisto
Ajan/Taajuuden peittäminen	Laitteisto
Ajan/Taajuuden venyttäminen	Laitteisto

Augmentointi tehdään menetelmästä riippuen aalto- tai spektrogrammimuodossa. Lisäksi joitakin menetelmiä voidaan tehdä molemmissa muodoissa. Kuvassa 13 esitettynä jo edellä näytetty kuva 2 augmentointikohtien kanssa.

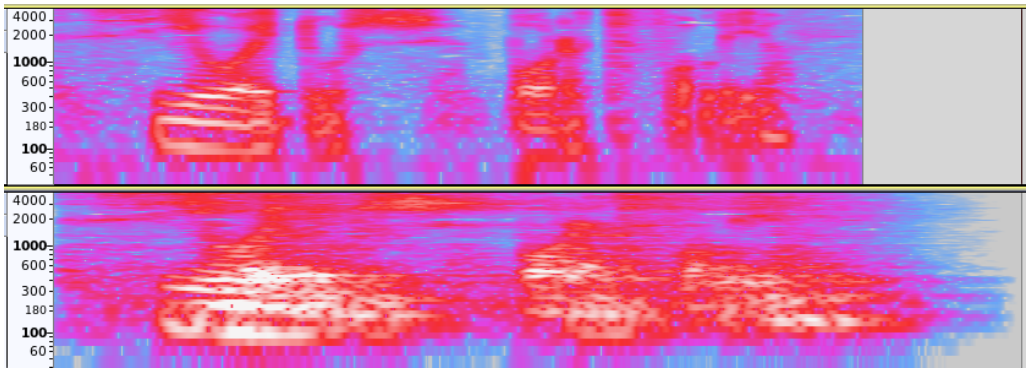


Kuva 13: Havainnekuva äänisignaalin prosessista ennen neuroverkoille syöttämistä augmentointien kanssa.

Äänidatan augmentointia voidaan siis suorittaa sekä aalto- että spektrogrammimuodoissa. Yleensä augmentointi on kuitenkin helpompaa spektrogrammille; esimerkiksi tietyn taajuuden peittäminen on pelkästään halutun taajuuden (y-akseli) poistamista datasta.

Aiemmin mainittiin, että puheentunnistusjärjestelmälle tulisi saada opetettua mahdollisimman kattava foneemivarasto. Valmiiksi olemassa olevan datan augmentointi ei kuitenkaan tuo uusia foneemeja esimerkkeihin, mutta augmentoinnin tarkoituksena onkin pelkästään ratkaista laitteiston, puhujan sekä ympäristön aiheuttamien variaatioiden virheitä. Lisäksi esimerkiksi ympäristön aiheuttamien variaatioiden osalta käytettäviä augmentointimenetelmiä voidaan keskittää kontekstiin, jossa järjestelmää tullaan käyttämään. [40]

Yksinkertaisuudessaan robustiuden parantumisessa on kyse opetusdatan suuresta lisäämisestä. Jos neuroverkolle on opetettuna sama foneemi useammalla eri opetusnäytteellä, on todennäköisempää, että jokin opetusnäytteistä vastaa testidatan olosuhteita ja tällöin myös spektrogrammin muotoa. Havainnollistetaan seuraavaksi miten kaiun lisääminen muuttaa spektrogrammia konkreettisen esimerkin avulla.



Kuva 14: Kaiun lisäämisen vaikutus spektrogrammiin ”sijoitusportfolio” -sanalle. Alemmassa ääninäytteessä lisätty kaiku tekee ääninäytteestä ajallisesti pidemmän. Kuvat `Audacity` -sovelluksesta.

Oletetaan, että neuroverkoille on opetettu yhdellä ääninäytteellä, miltä spektrogrammi näyttää jollekin sanalle studio-olosuhteissa. Toisaalta, jos sama sana sanotaan kaikuisassa tilassa niin spektrogrammi näyttää erilaiselta. Tässä tapauksessa lisäämme nauhoitukseen kaikua, jotta neuroverkoille voidaan opettaa myös, miltä näyttää kaiullinen esimerkki kyseiselle sanalle.

Kuvassa 14 spektrogrammeista ylempi on studio-olosuhteissa nauhoitettu ”sijoitusportfolio” sana ja alempi sama nauhoite lisätyn kaiun kanssa. Y -akseli on logaritminen välillä 0 - 4000 Hz. Molempia nauhoituksia kuunneltaessa sana on selvästi tunnistettavissa, eli kaikua ei ole lisätty liiaksi. Kaiullisen spektrogrammin äänidata on kestoltaan pidempi, mutta ääninäytettä pidentänyt kaiku oltaisiin voitu myös leikata pois, jolloin näytteet olisivat yhtä pitkiä.

Kuten kuvasta 14 huomataan, spektrogrammit näyttävät melko erilaisilta. Pelkästään tämän avulla voidaan perustella, että neuroverkkojen optimoiminen erilaisiin ympäristöihin on tärkeää. Augmentointimenetelmien fysikaalisista perusteluista kerrotaan myöhemmin luvussa 3.

Augmentoinnissa on kuitenkin varottava yli-augmentointia, jolloin tiettyä esimerkkiä opetettaisiin liiassa määrin neuroverkolle. Esimerkiksi säröilyn lisääminen tapauksessa spektrogrammi olisi niin sekaista, ettei siitä olisi enää erotettavissakaan alkuperäistä sanaa.

Yli-augmentointia vältetään erilaisilla menetelmillä. Eri augmentoinnissa varioinnin voimakkuuteen vaikuttavat muuttujat ovat satunnaisia, mutta rajoitettuja. Tällöin esimerkiksi kaikujen voimakkuudet ja viiveet tulevat ennalta testaamalla tarkistettujen arvojen väliltä. Yli-augmentointia pyritään välttämään myös esimerkiksi lisäämällä puheettomia ääninäytteitä opetusdataan.

Augmentointi on tietokonepohjaista, joten uuden datan luominen on huomattavasti nopeampaa kuin uuden kaiullisen ääninäytteen nauhoittaminen ja annotoiminen. Augmentointinopeus riippuu ääninäytteiden pituudesta ja laitteistosta, mutta myös käytetystä augmentointitavasta sekä sen koodillisesta toteutettamisesta. Tämän opinnäytetyön augmentoinneissa pyritään siihen, että ne eivät hidasta neuroverkkojen opettamista.

Tietokone pystyy augmentoimaan keskimäärin 1000 kappaletta ääninäytteitä sekunnissa. Luvussa 2.1.1 laskettiin, että yksi kesätyöntekijä sai annotoitua noin 25 kpl ääninäytteitä tunnissa. Kesätyöntekijöiden litteroiman 20 000 ääninäytteen lisääminen kestäisi augmentoimalla noin 20 sekuntia.

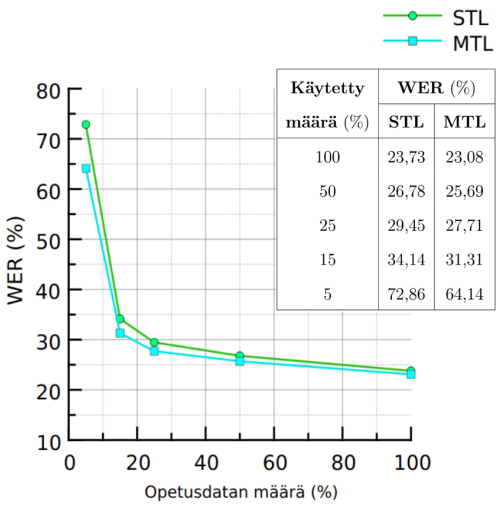
On kuitenkin muistettava, että myös alkuperäistä dataa tulee olla tarpeeksi, jotta augmentointi on ylipäätään kannattavaa; yhtä ääninäytettä loputtomasti augmentoimalla ei saada toimivaa puheentunnistusjärjestelmää aikaiseksi.

Pääsääntönä augmentoinnissa on, että sen jälkeenkin ääninäytteiden tulee kuulostaa realistisilta. Fysikaalisesti perusteltujen augmentointitapojen, joihin tämä opinnäytetyö painottuu, tulisi teoriassa olla tällaisia.

2.3 Opetusdatan määrän vaikutus robustisuuteen

Lähtökohtaisesti lisäämällä opetusdataa saadaan parempi koneoppimismalli. Tarvittavan datan määrä riippuu käyttökohteesta. Puheentunnistusjärjestelmän tapauksessa luotettavana määränä on pidetty noin 1000 tuntia monipuolista opetusdataa.

Tarkkaa verrannollisuutta opetusdatan määrän ja järjestelmän robustiuden välillä on vaikea määrittää useammastakin syystä: esimerkiksi, kun kahdella eri järjestelmällä on eriävät opetusdatat, voi toisella järjestelmällä olla tärkeät opetusesimerkit juuri testidatan kannalta. [41]



Käytetty määrä (%)	WER (%) STL	MTL
100	23,73	23,08
50	26,78	25,69
25	29,45	27,71
15	34,14	31,31
5	72,86	64,14

Kuva 15: Opetusdatan määrän vaikutus WER:iin. [41]

Vaikka korrelaatiota on vaikeaa mitata on sitä siitä huolimatta tutkittu esimerkiksi eri arkkitehtuurien välillä. Pironkov, Dupont ja Dutoit tutkivat opetusdatan määrän vaikutusta MTL (*engl. Multi-Task Learning*) ja STL (*Single-Task Learning*) välillä. Tulokset esiteltynä kuvassa 15. [41]

Pironkovin esimerkissä opetusesimerkkejä oli 7138 kappaletta (100%), mutta suoranaisen kappalemäärän lisäksi olennaisia tekijöitä ovat esimerkkien pituus sekä myös laatu. Kuvan 15 kuvaajasta olennaisin huomio lieneekin käyrän jyrkkä muutos kun opetusdatan määrä laskee alle 20%:iin. Opetusdatan määrän ja järjestelmän robustiuden välinen suhde ei oletetustikaan ole lineaarinen.

3 Fysikaaliset perustelut augmentointitavoille

Kuten mainittua, olisi olennaista, että augmentointimenetelmät muokkaisivat dataa sellaiseksi kuin se esiintyy käytettävän järjestelmän ympäristössä, jonka lisäksi datan tulisi edelleen olla realistista. Tämän lopputyön augmentointimenetelmien realistisuus perustellaan fysiikalla ja augmentointien voimakkuus kuuntelemalla ääninäytteitä.

Yleisesti puheen ymmärrettävyyttä eri tiloissa voidaan mitata ja ilmoittaa tulos puheensiirtoindeksinä (*Speech Transmission Index*, STI) välillä 0-1. Arvolla 0 puheesta ei pysty erottamaan yhtään tavua ja arvolla 1 jokainen tavu on selkeästi erotettavissa. Hyväksi STI -arvoksi on määritelty yli 0.75. [42] [43]

STI:n arvo riippuu monista eri tekijöistä, kuten jälkikaiunta-ajasta, taustamelusta, säröilystä, puhujan äänenvoimakkuudesta, taajuuksista sekä etäisyyksistä. Tämän luvun kaikki augmentointimenetelmät liittyvätkin STI -arvoon. Käytännössä muilla menetelmillä huononnetaan STI -arvoa, mutta taajuusvasteiden augmentoinnilla, luvussa 3.3, pyritään parantamaan sitä. [42] [43]

3.1 Kaiut

Aikaisemmin esimerkkinä mainittu kaiku on yksi augmentoitavista muuttujista, jonka tärkeys voidaan perustella luvun 2.2.1 kuvan 14 esimerkillä, jossa spektrogrammi muuttui olennaisesti kaikua lisäämällä.

Kuten kappaleessa 1.1.1 mainittiin, ääniaallot ovat pitkittäistä aaltoliikettä, joka etenee jossain väliaineessa. Väliaineen lisäksi ääniaallot voivat kohdata pinnan, jonka vuoksi osa ääniaallosta heijastuu takaisin ja osa jatkaa etenemistään toisella puolella

pintaa. Takaisin referenssipisteeseen kulkenutta aaltoa kutsutaan kaiuksi. [5]

Kaiku voidaan jakaa aikaiseen sekä myöhäiseen kaikuun (*engl. early & late reverberation*). Myöhäisen kaiun on todettu heikentävän puheen ymmärrettävyyttä. Useimmissa tilanteissa heijastunut ääni etenee kuitenkin niin nopeasti takaisin referenssipisteeseen, että ihmisen korvaan se ainoastaan vahvistaa alkuperäistä ääniaaltoa. Tällöin kyse on aikaisesta kaiusta, jonka on itseasiassa todettu helpottavan puheen ymmärrettävyyttä. [4] [5]

Spektrogrammeissa aikaiset kaiut eivät kuitenkaan suoraan helpota puheen ymmärrettävyyttä. Suurin osa opetusdatasta on äänitetty studio-olosuhteissa, joissa kaiun syntymistä pyritään välttämään esimerkiksi erilaisilla huoneakustiikan äänenvaimennusmateriaaleilla. Tämän vuoksi kaikujen augmentointi onkin äärimmäisen tärkeää tässä lopputyössä käytössä olevan puheentunnistusjärjestelmän opetusdatalle.

Kaiun voimakkuus ja kesto riippuvat monista tekijöistä, kuten äänilähteen voimakkuudesta, heijastavan pinnan ominaisuuksista, etäisyydestä äänilähteeseen ja kaikujen välillä kuluvasta ajasta. [44]

Kaikuun liittyvistä huoneen akustisista ominaisuuksista puhuttaessa käytetään usein mittarina jälkikaiunta-aikaa (*engl. Reverberation time*, RT), joka on tietyn äänenvoimakkuuden vaimentumiseen kuluva aika. Jälkikaiunta-aikaan vaikuttavat esimerkiksi huoneen tilavuus ja materiaalien sijoittelut, mutta myös huoneen materiaalien vaimennuskertoimet α_s. [44]

Yleensä jälkikaiunnassa käytetään 60 dB vaimentumiseen kuluvaa aikaa T_{60} tai RT_{60}. Jälkikaiunnan tarkka määrittäminen on laskennallisesti monimutkaista, jonka vuoksi sille on määritelty arviointimenetelmiä. [44]

Yksi menetelmä jälkikaiunta-ajan estimoimiseen on Sabinen yhtälö:

$$T_{60} = 0,161 \, \frac{\mathrm{s}}{\mathrm{m}} \cdot \frac{V}{\sum S_i \alpha_i} \, ,$$

jossa S_i on pinnan pinta-ala, α_i on materiaalin absorptiokerroin ja V on huoneen tilavuus. Lisäksi 0,161 s/m on kokeellisesti määritetty vakio, josta käytetään yleisesti arvoja väliltä 0,160 - 0,164, välillä yksikön kanssa ja välillä ilman. Sabinen yhtälössä ei huomioida ympäröivän ilman aiheuttamaa vaimentumista, joka on merkittävää varsinkin suurissa huoneissa. [44]

Toinen useasti käytetty yhtälö jälkikaiunta-ajan estimoimiseen on Eyringin yhtälö, jossa huomioidaan myös ympäröivän huoneilman aiheuttama vaimentuminen vaiheittaisen energian vaimenemisen avulla. Sabinen yhtälöä voidaankin pitää Eyringin yhtälön yksinkertaistettuna mallina. [45]

Eyringin mukaan Sabinen yhtälö ei anna tarkkaa kuvaa huoneen oikeasta jälkikaiunnasta varsinkaan silloin, jos huone itsessään absorboi paljon ääntä. Lisäksi kaikumisaika riippuu myös huoneen muodosta, jota ei Sabinen yhtälössä huomioida. [45]

Esimerkiksi seuraavassa luvussa esiteltävä `Pyroomacoustics` -kirjasto kuitenkin käyttää vakio arvoltaan Sabinen yhtälöä arvioimaan jälkikaiunta-aikaa. Tämä johtuu luultavasti siitä, että Sabinen yhtälöllä saadaan tarpeeksi hyvä arvio jälkikaiunnalle, kun puhutaan normaalien kokoisista tiloista.

3.1.1 Kaikujen augmentointi

Kaikujen varsinaista augmentointia varten kokeiltiin muutamia erilaisia menetelmiä, joita testattiin ja arvioitiin kuuntelemalla niiden aitoutta sekä neuroverkkoa opettamalla.

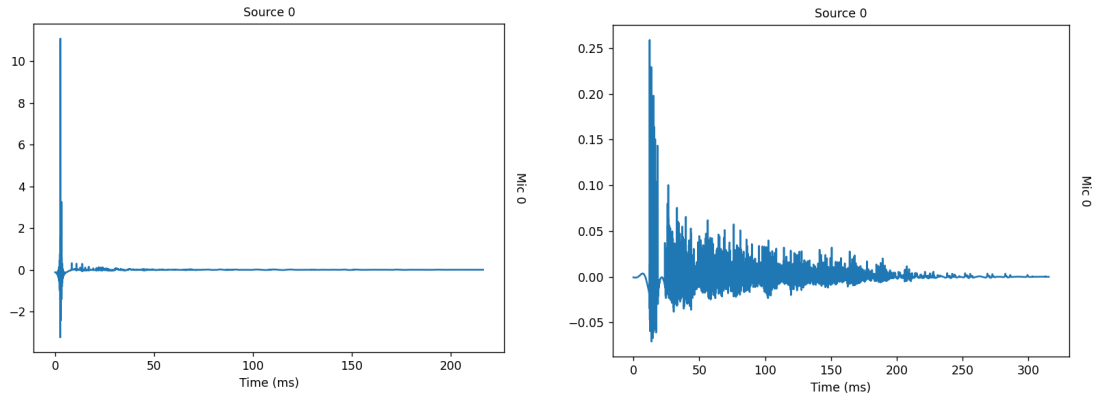
Aluksi äänisignaalia lisättiin vaimennettuna viiveellä itseensä, joka noudatti käytännössä samaa tekniikkaa kuin kampasuodattimet (*engl. comb filter*). Tuloksena saatiin kuitenkin epäluonnolliselta kuulostavia ääninäytteitä. Lisättyyn signaaliin päätettiin vielä soveltaa Oren-Nayar reflektiomallia (*engl. Oren-Nayar reflectance model*), joka simuloi heijastuksien fysikaalisia ominaisuuksia. [46]

Vaikka reflektiomalliin sovelletuilla kaiuilla saatiin hyviä tuloksia, päätettiin vielä kokeilla valmista `Pyroomacoustics` nimistä python -kirjastoa. Kirjaston avulla saatiin aidoimmilta kuulostavia ääninäytteitä laskennallisesti kevyimmällä tavalla. Lisäksi niiden satunnaistaminen oli triviaalia. [47]

`Pyroomacoustics` -kirjaston avulla voidaan simuloida satunnainen huone, jonne sijoitetaan myös äänilähde (kaiutin) sekä vastaanotin (mikrofoni). Näiden avulla pystytään mittaamaan impulssivaste (*impulse response*, IR) huoneelle.

Kirjastossa pystytään satunnaistamaan huoneen ominaisuuksia, kuten muotoja ja kokoja sekä pintojen materiaaleja. Satunnaisuuksien avulla saadaan haluttua ominaisuutta augmentointiin, eli aina toisistaan eroavia tiloja, joista saadaan erilaisia kaikuja. [47]

Impulssivasteet ovat malleja tilojen kaikuvasteista ja ne voidaan mitata myös aidolle huoneelle. Tällöin mittaamiseen yleensä käytetään siniaalto pyyhkäisyä (*sine sweep*), mutta siihen voidaan käyttää myös esimerkiksi jotain lyhyttä kovaa pamausta. [44]



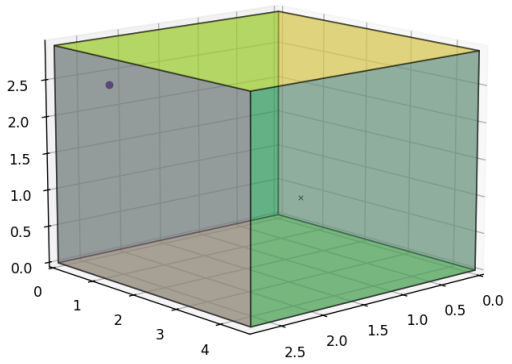
(a) (b)

Kuva 16: Kaksi esimerkkiä pyroomacoustics -kirjaston avulla luoduista impulssivasteista. Huomattavaa on, että akselien suhteet eroavat toisistaan.

Kuvassa 16 esiteltynä kaksi satunnaista impulssivastetta, jotka luotiin kirjaston avulla käyttäen lyhyttä pamausta. Kuvaajien akselien eivät ole samassa suhteessa keskenään.

Kuvien 16a ja 16b kuvaajia vertaamalla voidaan havaita, että impulssivasteet eroavat paljon toisistaan, eli satunnaistaminen toimii halutusti. `Pyroomacoustics` -kirjastolla voidaankin mm. valita satunnaisesti monista eri materiaaleista huoneen pinnat, joilla on erilaiset ominaisuudet äänen heijastumisen kannalta. Materiaaleja ovat esimerkiksi tiili, betoni, puu, kaiuton pinta, kovapinta, puuvilla sekä metalli, mutta myös monia muita materiaaleja niitä ollessa kaikkiaan yli 50.

Kaikuefektillä (*engl. reverb effect*) voidaan simuloida kaiku uusiin ääninäytteisiin hyödyntäen esimerkiksi kaikukammioita tai tietokoneita. Ns. konvoluutiokaiun (*engl. convolution reverb*) avulla voidaan käyttää impulssivasteita syntetisoimaan kaiku uuteen signaaliin. [44] [47]



Kuva 17: `Pyroomacoustics` -kirjaston avulla simuloitu huone. Kuvassa piste ja rasti vastaavat lähteen ja vastaanottimen paikkoja. Kuvassa olevan huoneen impulssivaste esiteltynä erikseen kuvassa 16b.

Kirjaston avulla luotiin 10 000 impulssivastetta, joita augmentoinnissa käytettiin satunnaisesti. Huoneet luotiin siten, että lähde ja vastaanotin ovat aina täysin satunnaisessa sijainnissa, mutta 50% simulaatioista lähellä toisiaan ja 50% eri paikoissa keskenään. Huoneen materiaalit satunnaistettiin erikseen jokaiselle pinnalle.

Huoneista satunnaistettiin 20% pienikokoisiksi (alle 1-2 m sivut ja 1-2m korkeus), 20% todella suuriksi (5-50 m sivut ja 4-10 m korkeus) ja loput 60% normaalien huoneiden kokoisiksi (2-6 m sivut ja 2-4 m korkeus).

Kuvan 16a impulssivasteen huone on todella suuri (koko 33,81m x 11,63m x 6,04m) ja vastaanotin samassa paikassa kuin äänilähde. Kuvan 16b huone on normaali (koko 2,75m x 4,54m x 2,95m) ja vastaanotin eri paikassa kuin äänilähde. Impulssivasteen 16b huone esitelty kuvassa 17.

Simuloituja impulssivasteita hyödyntäen opetusdatan voidaan lisätä kaikua siten kuin se olisi tallennettu kyseisessä huoneessa konvoluution avulla. Tällöin opetusdata kerrotaan impulssivasteen kanssa:

$$y(n) = \sum_{k=-\infty}^{\infty} (f(k) \cdot h(n-k))$$

jossa $h(n-k)$ on impulssivaste siirrettynä aikaerolla.

Simuloiduissa impulssivasteissa jälkikaiunta-aika laskettiin `Pyroomacoustics` -kirjastossa Sabinen yhtälöllä. Kirjaston metodissa olisi kuitenkin mahdollista määritellä funktioksi myös Eyringin -yhtälö. Tällöin oltaisiin esimerkiksi voitu määritellä Eyringin yhtälöä käytettävän ainakin isoihin tiloihin, joita luotiin 20% impulssivasteista.

Konvoluutiossa käytetään ns. diskreettien funktioiden sarjakehitelmien summaa, joka lasketaan `scipy.signal` -kirjaston convolve-metodin avulla. Konvoluution tuloksena saadaan uusi augmentoitu opetusdata, joka sisältää impulssivasteen avulla simuloidun huoneen kaikujen vaikutukset. Impulssivasteiden hyödyntäminen on laajassa käytössä lukuissa erilaisissa signaalinkäsittelyn tehtävissä.

3.2 Taustamelut ja säröilyt

Vastaavasti kuin kaiun kanssa, ei studio-olosuhteissa tallennetussa opetusdatassa ole myöskään ääniä esimerkiksi ohi ajavista ajoneuvoista tai taustalla olevien henkilöiden puheesta.

Taustaäänien augmentoinnissa tulee ottaa huomioon esimerkiksi mikrofonityypistä riippuva muiden äänien päätyminen signaaliin. Säröilyjä taas voi syntyä esimerkiksi laitteistosta tai signaalin käsittelystä johtuvista syistä. Käytetyt mikrofonit ja mahdolliset puheentunnistusjärjestelmän käyttökohteet vaikuttavat myös paljon mahdolliseen säröilyyn tunnistettavassa äänisignaalissa. [4]

Ideaalitilanteessa tallennetuksi tulisi ainoastaan haluttu signaali. Kuitenkin todellisuudessa analogiset sekä digitaaliset signaalit aina muokkautuvat ja vääristyvät, minkä lisäksi usein tallentuu myös ylimääräistä ulkoista ääntä. [4]

Olennainen tekijä taustaäänien päätymisessä tallennettavaan signaaliin on mikrofoni, kuten luvussa 1.1.1 esiteltiin. Mikrofonin ominaisuuksista yksi tärkeimmistä on taajuusvaste, jota käsitellään vielä tarkemmin taajuusvasteiden luvussa 3.3. [4]

Tallennetun signaalin säröilyyn vaikuttaa olennaisesti myös käytetty laitteisto, kuten esimerkiksi erilaisten esivahvistimien käyttö. Vahvistimien vaste on usein lineaarinen matalilla taajuuksilla, mutta epälineaarinen korkeilla taajuuksilla. [4]

Muita säröilyjä signaaliin aiheutuu esimerkiksi kompressoinnista, salauksesta tai konversiosta analogisesta digitaaliseksi. Analogisen signaalin muokkaamisessa digitaaliseksi pätee jälleen Nyquist-Shannon teoreema, eli muuntamisen jälkeen näytteenottotaajuus tulisi olla vähintään kaksinkertainen alkuperäisen signaalin korkeimpaan taajuuteen verratessa. [4]

Taustahälyjä ja säröilyjä pyritään välttämään monilla tavoin, kuten valitsemalla käytettävä laitteisto tarpeiden mukaan. Esimerkiksi monien viranomaisten toimintaympäristöissä normaalin mikrofonin käyttö aiheuttaisi huomattavaa häiriötä signaaliin. Tällaisia viranomaisia ovat esimerkiksi moottoripyöräpoliisit, sotilaat ja palomiehet. Taustaäänien lisäksi kyseiset henkilöstöryhmät tarvitsevat yleensä tehtävissään kätensä vapaaksi, joka rajoittaa tavallisten tangettien käyttöä.

Edellä mainituilla viranomaisilla laajassa käytössä ovat olleet erilaiset ns. kontaktimikrofonit, eli esimerkiksi kurkku- (laryngofoni) sekä kallomikrofonit.

Kontaktimikrofonit tallentavat puhetta pintojen värähtelyn avulla. Normaalit akustiset mikrofonit mittaavat ilmassa tapahtuvaa paineenvaihtelua, mutta kontaktimikrofonit tallentavat puhetta eräänlaisilla iholla olevilla kiihtyvyysantureilla. Esimerkiksi kurkkumikrofonit mittaavat kurkunpään värähtelyjä. [48]

Ihon värähtelyn mittaamisen ansiosta kontaktimikrofonit ovat nykyään erinomaisia, jos ympäröivä melutaso on korkea, mutta haittapuolena on signaalin heikko taso verrattuna normaaliin akustiseen mikrofoniin. Kontaktimikrofonien käyttöä on kuitenkin tutkittu paljon niiden kehittämiseksi, kuten esimerkiksi miten niitä voitaisiin käyttää yhdessä akustisten mikrofonien kanssa. [48] [49]

Nykyään ovat yleistymässä myös ns. korvakäytävämikrofonit, jotka mittaavat värähtelyn suoraan korvakäytävästä saman laitteen toimien myös kaiuttimena saapuvalle radiosignaalille. [50]

Yhdysvaltojen armeijan maavoimien ilmailulääketieteellinen tutkimuslaboratorio (*US Army Aeromedical Research Laboratory*) tutki normaalien akustisten melua vaimentavien (*engl. noise-cancelling*) ja erikoisjoukkojen käyttämien kurkkumikrofonien välistä eroa puheen tunnistettavuudessa helikoptereiden kanssa toimiessa. [49]

Tutkimuksissa todettiin kurkkumikrofoneille parempi signaali-kohinasuhde. Kurk-

kumikrofonin korkeat taajuudet olivat kuitenkin niin heikkoja, että konsonantteja oli huomattavasti vaikeampi havaita, jonka vuoksi niiden puheen ymmärrettävyys oli heikompi verrattaessa akustiseen melua vaimentavaan mikrofoniin. [49]

20 vuotta sitten tehdyssä tutkimuksessa jo kuitenkin arvioitiin, että teknologian kehityksen avulla kurkkumikrofonien suorituskykyä pystytään parantamaan, jolloin ne ohittaisivat myös melua vaimentavat mikrofonit STI -arvoja vertailtaessa. [49]

3.2.1 Taustamelun ja säröilyn augmentointi

Taustamelua voidaan augmentoida opetusdataan esimerkiksi lisäämällä suhteessa matalemmalla äänenvoimakkuudella ohi ajavan auton ääntä tai satunnaista puhetta.

Tämän lopputyön puheentunnistusjärjestelmän opettamiseen taustamelua on kerätty useista eri lähteistä. Yhteensä taustameluja on yli 62 000 kappaletta, joista jokainen on 2 sekunnin mittainen. Taustamelu yksinkertaisesti yhdistetään opetusdataan varioiden ja suhteuttaen sen amplitudia (-20 dB $<$ x $<$ -7 dB).

Säröilyn lisäystyyli arvottiin kahdesta eri menetelmästä 50/50 -suhteella. Ensimmäisessä, kovemman säröilyn, versiossa opetusdatan ääniaallot rajoitettiin satunnaisesti arvoituille amplitudeille. Toisessa versiossa säröilyä lisättiin suhteessa alkuperäiseen voimakkuuteen hyperbolisen tangentin -avulla. Jälkimmäisen version luomaa säröilyä voidaan pitää huomattavasti pehmeämpänä.

Molemmissa tapauksissa raja-arvot amplitudien rajoituksille testattiin kuuntelemalla augmentoituja ääninäytteitä ja uusi säröilyä sisältävä opetusdata skaalattiin vielä suhteessa alkuperäiseen amplitudiin.

3.3 Taajuusvasteet

Luvussa 1.1.1 kerrottiin, että on olemassa erilaisia mikrofoneja, joiden toimintamenetelmät eroavat toisistaan, jonka vuoksi myös äänet tallentuvat eri tavoin. Yksi tällainen jokaiselle mikrofonille ominainen muuttuja on taajuusvaste, jonka vaikutuksen augmentointi on tärkeää.

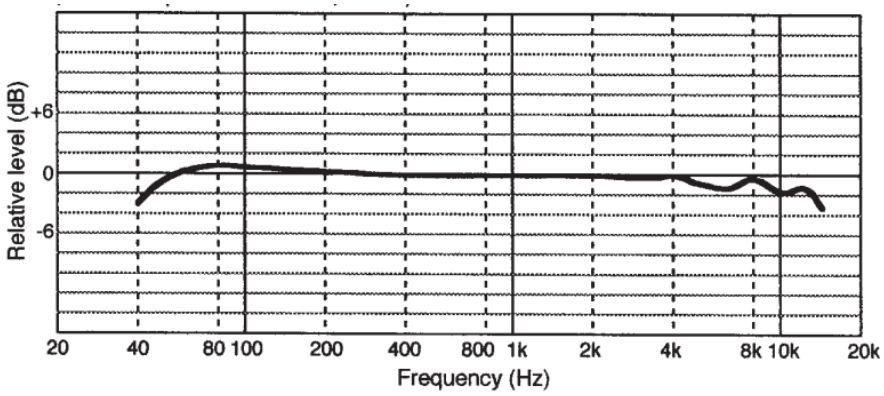
Taajuusvaste (*engl. frequency response*) on mikrofonien ominaisuus, joka määrittää taajuudesta riippuvan mikrofonin vasteen. Yleensä taajuusvaste esitetään graafisesti taajuusvastekäyrän (*engl. frequency response curve*) avulla, jolloin kyseessä on eräänlainen Bode-diagrammi. Bode-diagrammit ovat kuvaajia, joilla kuvataan eri asioita taajuuden funktiona, eli tässä tapauksessa suhteellinen äänenvoimakkuus taajuuden funktiona. [51]

Käytännössä taajuusvasteesta voidaan määrittää edellä mainittu suhteellinen äänenvoimakkuus, jolla mikrofoni vastaanottaa ääntä eri taajuusalueilla, sekä tietysti myös taajuusalue, jolla mikrofoni ylipäätään toimii. [7] [51]

Taajuusvastekäyrässä x-akselilla ovat taajuudet ja y-akselilla desibelit (dB). Taajuudet esitetään yleensä logaritmisella asteikolla ja suhteellinen dB -nollataso tarkennetaan yksikön dimensiomattomuudesta johtuen. Esimerkkikuva mikrofonin tasaisesta taajuusvastekäyrästä esiteltynä kuvassa 18. [7]

Käyttötarkoituksesta riippuen, mikrofonin taajuusvastekäyrä voi olla esimerkiksi kuvan 18 mukainen tasaisen vasteen mikrofoni *(engl. flat response microphone).* Tällöin volyymintaso ei ole taajuusriippuvaista, joten taajuudet tallentuvat suhteessa toisiinsa samalla volyymilla. [7]

Toinen yleinen taajuusvaste on muotoillun vasteen mikrofoni (*engl. shaped response microphone*), jossa mikrofoni tallentaa eri volyymilla eri taajuuksia. [7]



Kuva 18: Esimerkki taajuusvastekäyrästä tasaisen vasteen mikrofonille. [7]

Esimerkiksi musiikkia tallennettaessa on oleellista saada kaikkia taajuuksia tasaisesti tallennettua, joten tasaisen taajuusvasteen mikrofonit ovat laajasti käytettyjä musiikin parissa. Toisissa tapauksissa, kuten esimerkiksi podcasteissa, on taas tärkeämpää tallentaa puheen kannalta olennaiset matalat taajuudet kuin korkeat taustahälyt. [7]

Mikrofonien tallentamaa ääntä voidaan käsitellä esimerkiksi vahvistamalla tai heikentämällä tiettyjä taajuuksia ekvalisaattorin avulla. Tällä tavalla mikrofoni voidaan paremmin kalibroida käyttötarkoitukselleen tarkoitettuun ympäristöön.

Taajuuskorjausta voidaan tehdä myös mikrofonin tyypistä johtuen: esimerkiksi edellisessä kappaleessa mainituissa kallo- tai kurkkumikrofoneissa tiettyjä taajuuksia tulee vahvistaa voimakkaasti, jotta sanotun puheen ymmärtäminen on ylipäätään mahdollista.

Seuraaksi luvussa 3.3.1 tutkitaan kaupallisen mikrofonin ominaisuuksia vastaanottaa ääntä eri taajuusalueilla eri sijainnista äänilähteeseen nähden.

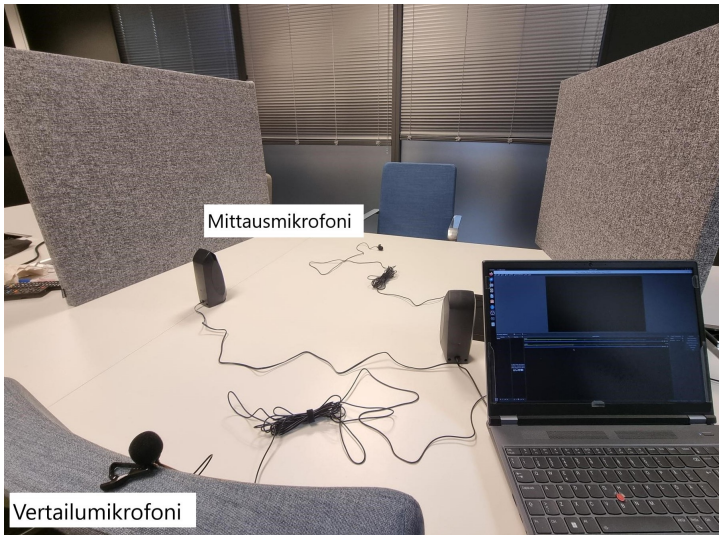
3.3.1 Taajuusvasteen mittaaminen mikrofonille

Mittauksissa käytettiin *BOYA, BY-M3* mikrofonia, josta tutkittiin suhteellista äänen vastaanottamista eri sijainneista eri taajuusalueilla. Kokeita varten hankittiin kaksi kappaletta edellä mainittuja mikrofoneja. Mittauksien tarkoituksena oli saada suhteellisia taajuusvasteita samalle mikrofonille kahdesta eri sijainnista tallennettuna.

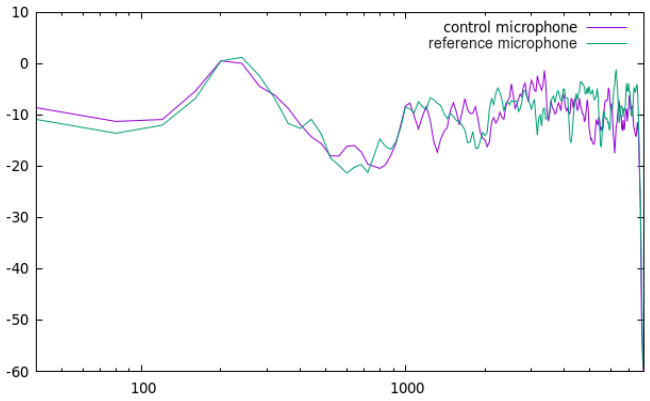
Suhteellisten taajuusvasteiden avulla pystytään augmentoimaan äänenvoimakkuutta taajuuden perusteella, kun tiedetään tallentamiseen käytetyn mikrofonin sijainti. Esimerkiksi, jos 1 metrin päässä olevassa mikrofonissa tietyt taajuudet vaimenevat nopeammin kuin vieressä olevassa mikrofonissa tulisi kyseisten taajuuksien äänenvoimakkuutta kasvattaa enemmän suhteessa muihin.

Mittaushuoneena toimi suhteellisen hiljainen neuvottelutila, jossa äänen heijastumiseen oli jo valmiiksi kiinnitetty huomiota. Kaikilta taustaääniltä tai heijastuksilta ei kuitenkaan voitu välttyä. Kaikujen syntymistä pyrittiin vielä erikseen välttämään asettamalla tilaan lisää pehmeitä akustiikkalevyjä. Äänen tallentamiseen käytettiin `OBS Studio` -sovellusta, jonka avulla saatiin yhdistettyä kaksi mikrofonia samalle tietokoneelle. Kuvassa 19 esiteltynä mittausasetelma.

Aluksi suoritettiin vertailumittaus, jossa molemmat mikrofonit olivat samassa paikassa, jotta voitiin varmistua laitteiston toimivuudesta. Vertailumittauksen tarkoituksena oli varmistaa, että samassa kohdassa saadaan lähes identtiset taajuusvastekäyrät molemmista mikrofoneista. Kuvassa 20 esiteltynä vertailumittauksen tulos.



Kuva 19: Kokeen mittausasetelma.



Kuva 20: Vertailumittaus. Mikrofonit vierekkäin. X-akselilla taajuudet logaritmisella asteikolla ja y-akselilla äänenvoimakkuuden suhteelliset desibelit.

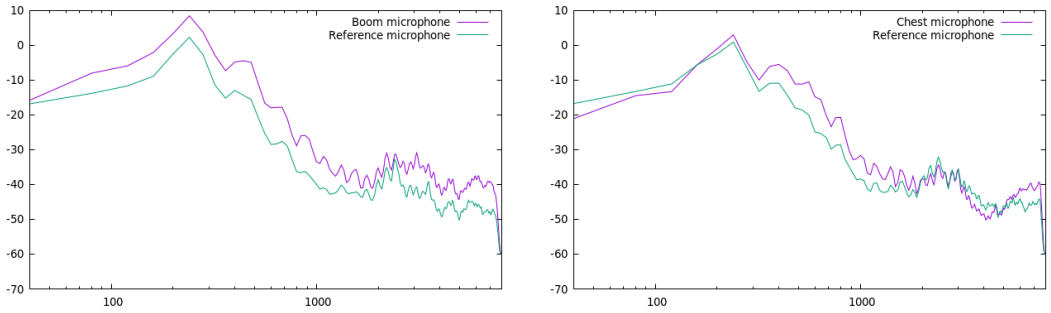
Kuvan 20 perusteella voitiin todeta taajuusvasteiden olevan lähellä toisiaan, jonka avulla voitiin varmistua mikrofonien ja mittauksien luotettavuudesta.

Varsinaiset mittaukset suoritettiin kahdesta eri mittausmikrofonin sijainnista, jotka olivat kuulokkeiden päässä puomissa (*engl. boom*) sekä puhujan rinnassa. Vertailumikrofonin sijainti oli metrin päässä suoraan puhujasta. Ensimmäiseksi suoritettiin

mittaus, jossa mittausmikrofoni oli kiinni kuulokkeiden puomissa.

Mittaukset suoritettiin siten, että kolme eri henkilöä kävivät kaikki samat vokaalit läpi. Ensimmäinen puhuja laittoi päähänsä headsetin, jonka puomissa mittausmikrofoni oli kiinni. Tietokoneesta laitettiin tallennus päälle ja puhuja alkoi lausumaan ennalta määriteltyjä vokaaleja ja lauseita erittäin selkeästi. Tärkeintä mittauksien kannalta oli, että niissä käytettiin laajaa taajuusaluetta. Tämän vuoksi kokeet suoritettiin kolmella eri puhujalla (2x mies, 1x nainen).

Seuraavaksi mikrofoni vaihdettiin puhujan rintaan kiinni ja samat vokaalit lausuttiin uudelleen. Tuloksiksi saatiin kolmesta eri puhujasta lasketut keskiarvoistetut taajuusvasteet, jotka esiteltynä kuvassa 22.



(a) Mittausmikrofoni puomissa. (b) Mittausmikrofoni rinnassa.

Kuva 21: Mittauksista saadut mikrofonin taajuusvastekäyrät.

Saaduista kuvaajista voidaan havaita, että mikrofonien taajuusvasteet muotoilevat toisiaan molempien mikrofonien ja mittauksien välillä, kuten voitiin olettaakin.

Kuitenkin, molemmissa mittauksissa noin 2-3 kHz välillä mittausmikrofonin arvot ovat lähempänä toisiaan kuin muualla mittausalueella. Mittauksien perusteella voidaan siis augmentoida taajuusvasteiden perusteella dataa siten, että taajuusalueen 2-3 kHz:n väliltä amplitudia ei kasvateta yhtä paljon kuin muualta.

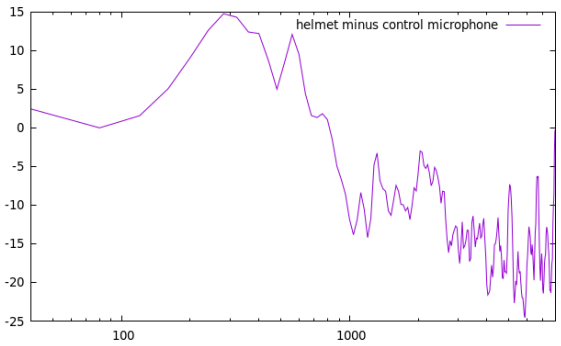
Lisäksi, mikrofonin ollessa puomissa kiinni, noin 300-800 Hz:n kohdalla on ero

referenssimikrofoniin suhteellisen suuri. Tämä tarkoittaa sitä, että kyseessä olevilla taajuusalueilla amplitudia tulisi kasvattaa suhteessa enemmän kuin muualta.

Lisäksi tutkittiin palolaitokselta lainassa olleen palomieskypärän avulla sen kykyä vaimentaa eri taajuuksia kypärän sisällä. Kypärä esiteltynä kuvassa 22a.

Mittaukset suoritettiin samoilla mikrofoneilla kuin aikaisemmin. Aluksi mittausmikrofoni asetettiin kypärän sisälle ja vertailumikrofoni sen ulkopuolelle. Kypärä asetettiin päähän ja toistettiin kaiuttimista eri ääniä laajalta taajuusalueelta. Tulokseksi saatiin mittaus- ja vertailumikrofonin amplitudien erotus eri taajuuksilla, josta kuvaaja esitelty kuvassa 22b.

(a) Kokeessa käytetty palomieskypärä.



(b) Mikrofonien vasteiden erotus.

Kuva 22: Mittauksista saadut mikrofonin taajuusvastekäyrät.

Kuvan 22b kuvaajasta huomataan, että noin 1 kHz taajuuksista ylöspäin kypärä sulkee enemmän ääntä pois kuin sitä matalimmilla taajuuksilla. Noin 20 dB:n ero amplitudissa on melko suuri ja se tullaankin tulevaisuudessa ottamaan huomioon taajuusvasteiden augmentoinnissa.

Kokeissa oltiin kiinnostuttu saatujen taajuusvasteiden suhteellisuudesta sijaintiin nähden, eikä tarkoituksena ollutkaan mitata absoluuttista taajuusvastetta mikrofonille. Virallinen mikrofonin taajusvaste mitattaisiin hiljaisessa huoneessa, jossa käytettäisiin kalibroidulla kaiuttimella toistettavaa sinipyyhkäisyä (*engl. sine sweep*), joka kävisi tietyn taajuusalueen, kuten 20 Hz - 20 kHz kokonaan läpi. [7] [50]

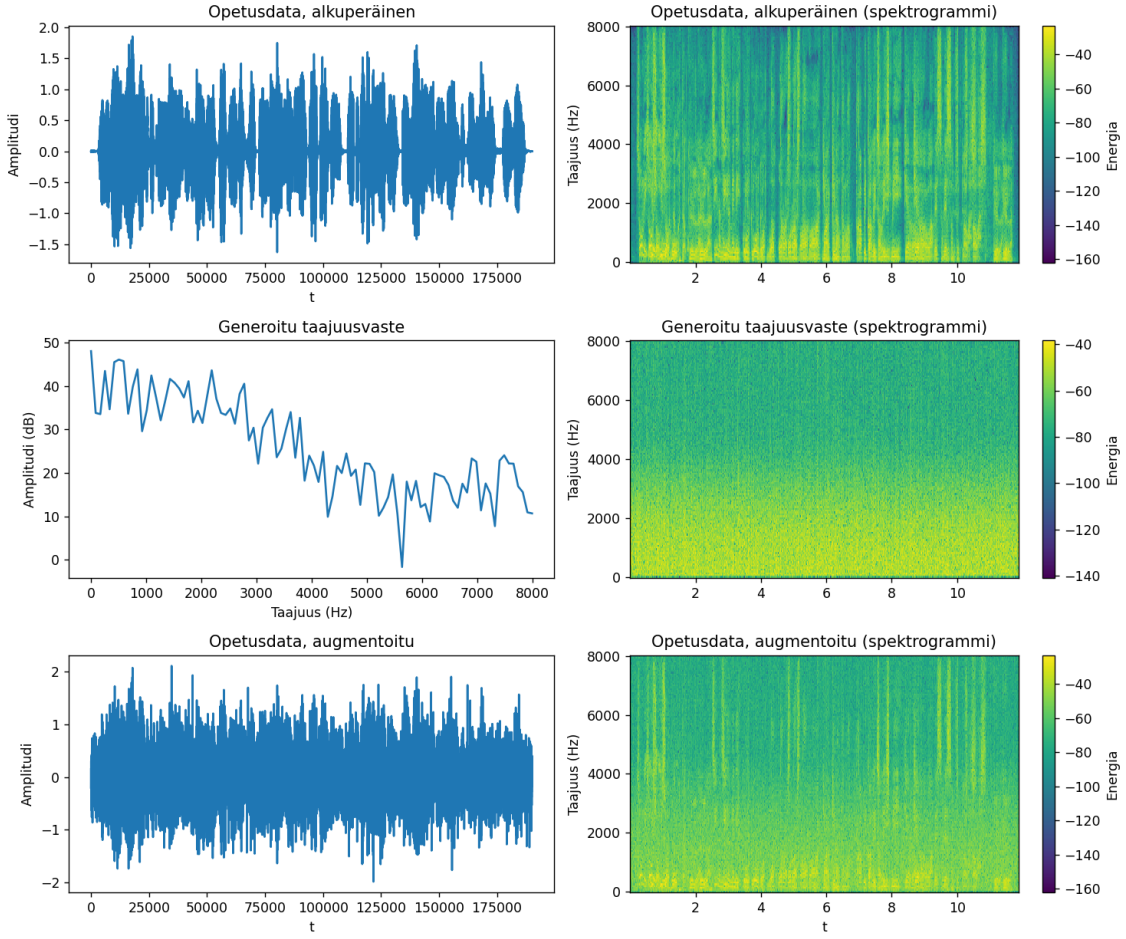
Tulokseksi saatiin taajuusvasteiden suhteita, joita tullaan käyttämään tulevaisuudessa taajuusvasteiden augmentoinnissa. Kuitenkin tämän opinnäytetyön puheentunnistusjärjestelmän taajuusvasteen augmentoinnissa käytettiin satunnaisesti luotuja taajuusvasteita, joista kerrotaan tarkemmin seuraavassa luvussa.

3.3.2 Taajuusvasteen augmentointi

Taajuusvasteiden augmentointiin ei tämän opinnäytetyön augmentaatiotesteissä vielä hyödynnetty edellä suoritettuja mittauksia, vaan augmentoinnissa käytettiin satunnaisesti generoituja taajuusvasteita.

Aluksi luotiin satunnainen lista, joka suhteutettiin jälleen satunnaistetuilla arvoilla. Suhteutetun listan arvoista muodostettiin tietokoneella generoitu taajuusvaste, jonka avulla voidaan augmentoida erilaisia vääristymiä ja muutoksia signaalin arvoihin. Lopuksi generoitu taajuusvaste yhdistetään alkuperäiseen opetusdatakappaleeseen. Augmentoitu opetusdatakappale on siten ikäänkuin tallennettu käytetyn taajuusvasteen omaavalla mikrofonilla.

Kuvassa 23 esiteltynä taajuusvasteiden augmentoinnin eri vaiheet sekä aaltomuotoina että spektrogrammeina. Ylimpänä alkuperäinen opetusdata, keskellä generoitu taajuusvaste ja alimpana taajuusvateen avulla augmentoitu opetusdata.



Kuva 23: Esimerkkikuva taajuusvasteen augmentoinnista. Kuvassa ylimpänä alkuperäinen opetusdata, keskellä generoitu taajuusvaste ja alimpana taajuusvateen avulla augmentoitu opetusdata.

Myös tämän augmentointimenetelmän luotettavuudesta varmistuttiin aluksi kuuntelemalla muokattua opetusdataa, jonka lisäksi suoritettiin kokeiluja neuroverkkojen opetukselle.

Taajuusvasteen augmentoinnin avulla saadaan simuloitua erilaisia taajuusvasteita, joiden avulla pystytään augmentoimaan haluttuja mikrofonin ominaisuuksia. Generoimalla taajuusvasteet tietokoneella säästetään aikaa taajuusvasteiden mittaamiselta.

4 Aikaisemmat tutkimukset ja testien tarkoitus

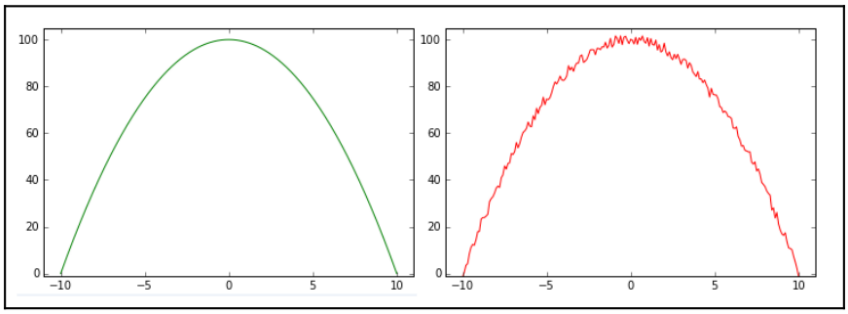
Vaikka aikaisemmissakin tutkimuksissa on saavutettu jopa alle 5% WER -tuloksiin pääseviä puheentunnistusjärjestelmiä, pyritään niitä silti saamaan entistä robustimmiksi. Luotettavan järjestelmän avulla puheentunnistusta pystyttäisiin alkaa käyttämään entistäkin vaativimmissa ja tärkeämmissä käyttökohteissa. [34]

Järjestelmien robustisuus on vieläkin ongelma varsinkin niissä tapauksissa, kun tunnistettavassa puheessa on paljon muita häiriöitä. Robustiuden parantamiseksi on tehty monia tutkimuksia, joista monet liittyvät neuroverkkoihin tai augmentointiin.

Puheentunnistusjärjestelmien robustiuden parannuksia on pyritty myös tutkimaan esimerkiksi lisäämällä audion lisäksi visuaalinen tunnistus, jolloin kyseessä olisi ns. AVSR -järjestelmästä (*engl. Audio-Visual Speech Recognition*). AVSR-järjestelmissä videokuvaa käytetään puheentunnistuksessa tukena, jolloin voidaan havaita kun puhujan huulet liikkuvat. [3] [52]

Käytännössä jo pelkästään videokuvasta on huuliltaluvun avulla mahdollista ymmärtää mitä henkilöt puhuvat ja ainoastaan videokuvan avulla onkin päästy noin 20% WER -arvoihin. AVSR:n on todistettu olevan käytännöllinen varsinkin taustameluisissa ympäristöissä, joissa taustamelu ei tietenkään vaikuta visuaaliseen sanotun tunnistamiseen. AVSR:n tuleekin välttää McGurk -efektiä. [3] [52]

Yksi olennaisimmista asioista robustiuden kannalta on opetusdatan ja todellisen käyttöympäristön keskinäinen samanlaisuus. Tällä tarkoitetaan sitä, että esimerkiksi taustahälyisissä ympäristöissä myös opetusdatassa olisi samanlaista taustahälyä. Ratkaisuksi on pyritty tekemään järjestelmän toimintaympäristölle kohdennettua augmentointia, jossa esimerkiksi lisätään juuri tietyntaajuista taustaääntä opetusdataan. [12] [34]



Kuva 24: Esimerkkikuva yksinkertaisesta ylisovitetuksesta pallon heiton lentoradalle. Vasemmalla hyvä ja oikealla huono sovitus lentorataan. [23]

Yleinen ongelma koneoppimisesssa on ylisovitus opetusdataan, jolloin neuroverkko oppii opetusdatan esimerkit niin tarkkaan, että se ei pysty tunnistamaan jatkossa enää uudesta datasta haluttuja asioita. [23]

Yksinkertainen ylisovituksen esimerkki esiteltynä kuvassa 24. Kuvassa vihreällä hyvä ja punaisella huono sovitus pallon lentoradasta. Huonossa esimerkissä koneoppimismalli on oppinut pallon lentoradan liian tarkasti esimerkiksi datassa olevien mittaushäiriöiden vuoksi. [23]

Edeltävissä kappaleissa on kerrottu erilaisia tapoja, joilla ylisovitusta pyritään välttämään tässäkin lopputyössä: esimerkiksi arkkitehtuurien valinta sekä dropout. Lisäksi tässä järjestelmässä käytetään mm. regularisointikertoimia (myös nimellä normalisointikerroin), L_1 ja L_2, early-stopping -menetelmää sekä muiden parametrien optimointia, joista kerrotaan vielä tarkemmin luvussa 5.1.

Lisäksi muita aikaisempien tutkimuksien yleisiä ongelmia ovat olleet opetusdatan riittämättömyys tietyille kielille, kuten esimerkiksi suomen kielelle.

Tässä lopputyössä tutkittavat augmentointimenetelmät ovatkin hyödyllisiä työkalu, koska ne auttavat moneen eri koneoppimisongelmaan, kuten ylisovituksen estämiseen sekä robustiuden parantamiseen.

5 Augmentointikokeet

Tässä luvussa käydään läpi tämän lopputyön koejärjestelyitä sekä itse augmentointikokeet ja niistä saadut tulokset. Tuloksista esitetään vielä erikseen kootut lopputulokset kappaleessa 6.

5.1 Koejärjestelyt

Tämän puheentunnistusjärjestelmän opettamiseen käytetään noin 170 000 kappaletta ääninäytteitä, jotka vastaavat ajallisesti noin 200 tuntia annotoitua puhetta. Yleisesti kirjallisuudessa luotettavan järjestelmän rajana pidettiin noin 1000 tuntia opetusdataa. Opetusdatan näytteenottotaajuutena on 16 000 Hz, jolloin Nyquist-Shannon -teoreeman mukaan puheen kannalta oleellisilta taajuusalueilta ei tulisi hävitä informaatiota. [11]

Yleisesti koneoppimisessa epookki on usein koko opetusdatan läpikäynti, mutta kuten kappaleessa 1.1.5 mainittiin, on yksi epookki tässä työssä vakioitu opetusaskelien määrään.

Epookkina käytetään kahdeksaa opetusaskelta johtuen siitä, että ennen tämän lopputyön testejä opetusdataa lisäiltiin aika-ajoin. Näissä tapauksissa epookkien vakiointi muuttuvaan dataan ei olisi ollut mielekästä. Kuitenkin tämän lopputyön testeissä käytetty opetusdata pidettiin vakiona.

Opetusdatasta otetaan 25 millisekunnin pituinen ikkuna 10 millisekunnin välein 400 Hz:n taajuudella käyttäen STFT:tä. Lisäksi jokaiselle STFT -muunnokselle lasketaan 128:n pituinen Mel-Frequency Cepstral -esitys (MFC), joille suoritetaan myös ns. Slaney-normalisointi. [53]

Opetuksessa neuroverkkojen arkkitehtuurina käytetään kirjallisuudessa tällä hetkellä parhaimpia tuloksia tuottavan conformer -arkkitehtuurin variaatiota. Alkuperäisestä conformer-artikkelista poiketen tässä puheentunnistusjärjestelmässä ei kuitenkaan käytetä ns. conformer-blokkeja, vaan LSTM-verkon transformer-pohjaisia dekoodaustasoja sekä attention -mekanismia. [23] [25] [27] [28]

Alkuperäisen conformer -artikkelin tavoin käytetään myös 10% dropout:ia sekä neuroverkon kernel- ja bias-normalisointia kertoimilla $L_1 = 0$ ja $L_2 = 10^{-6}$. Normalisointikertoimet estävät ylisovittamista varmistamalla, että muut parametrit ovat sopivissa rajoissa koulutuksen aikana muuttamalla häviöfunktiota. [23] [25] [33]

Aktivointifunktiona käytettiin softmax -funktiota ja optimointialgoritmina Adadeltaa. Opetusaskeleen pituus oli 0.05, joka laskettiin vielä kymmenesosa-arvoon 0.005 kun CTC -häviöfunktio ei enää laskenut neljään epookkiin, eli 32:een opetusaskeleeseen. Opetusaskeleeen pienentämisen avulla voidaan hienosäätää oppimista lokaalissa minimissä. [31] [54]

Vaikka residuaaliverkot ovatkin helpottaneet neuroverkkojen optimointia, ei kyseessä silti ole yksiselitteinen tehtävä, joka olisi uusittavissa aina samalla tavalla. Tieteellisesti määriteltynä neuroverkkojen opetus on vahvistettavissa replikoimalla, mutta opetuksen uusittavuuteen vaikuttavat laajalti myös satunnaisuudet.

Testien keskinäisen vertailukelpoisuuden kannalta on tärkeää normalisoida eri opetusmittaukset keskenään. Kuitenkin, esimerkiksi dropout sekä satunnaisparametroidut augmentaatiot tuovat oman satunnaisen komponenttinsa jokaiseen opetukseen. Esimerkiksi taustahälyt valitaan satunnaisesti yli 62 000 taustahälyesimerkin joukosta.

Itseasiassa, jo ilman augmentointimetodeja tai neuroverkkoja, pelkästään opetusdatan muodostamisestä syntyy satunnaisuutta, kun opetusdata yhdistellään ja

sijoitetaan satunnaisiin ikkunoihin.

Testien luotettavuutta arvioitiin kuitenkin vertaamalla tarkkuutta erilliseen 229 ääninäytteen vakioituun testidataan (validointidataan) ja laskemalla etäisyys luvussa 1.3 kerrotulla Levenshteinin etäisyydellä.

Edellä mainittujen satunnaisuuksien vuoksi testejä pyrittiin suorittamaan useita, jolloin tuloksien vertailua voitaisiin pitää luotettavampana. Useiden opetuksien mediaanin estimoitiin päätettiin käyttää Harrell-Davis (HD) -arviointia. [55]

Harrel-Davis -arviointi lasketaan painotetun summan avulla

$$M_{HD} = \sum_{i=1}^{n} W_{n,\,i}^{HD} X_{(i)} \ ,$$

jossa kertoimet $W_{n,i}^{HD}$ saadaan ns. epätäydellisestä beta-suhteesta $I_\alpha = (a, b)$:

$$W_{n,\,i}^{HD} = I_{i/n}\left(\frac{n+1}{2}\ ,\ \frac{n+1}{2}\right) - I_{(i-1)/n}\left(\frac{n+1}{2}\ ,\ \frac{n+1}{2}\right)\ , i = 1, \ldots, n. \text{ [55]}$$

Esimerkiksi $W_{n,\,i}^{HD}$ saadaan laskettua vertailutestien (luku 5.2.2) kolmelle tulokselle 1056, 1051 ja 1014:

$$\begin{aligned}
W_{3,\,1014}^{HD} &= I_{\frac{1014}{3}}\left(\frac{3+1}{2}, \frac{3+1}{2}\right) - I_{\frac{1014-1}{3}}\left(\frac{3+1}{2}, \frac{3+1}{2}\right) \approx 0{,}1875 \\
W_{3,\,1051}^{HD} &= I_{\frac{1051}{3}}\left(\frac{3+1}{2}, \frac{3+1}{2}\right) - I_{\frac{1051-1}{3}}\left(\frac{3+1}{2}, \frac{3+1}{2}\right) \approx 0{,}6250 \\
W_{3,\,1056}^{HD} &= I_{\frac{1056}{3}}\left(\frac{3+1}{2}, \frac{3+1}{2}\right) - I_{\frac{1056-1}{3}}\left(\frac{3+1}{2}, \frac{3+1}{2}\right) \approx 0{,}1875
\end{aligned}$$

$$\begin{aligned}
M_{HD} &\approx W_{3,\,1014}^{HD} \cdot 1014 + W_{3,\,1051}^{HD} \cdot 1051 + W_{3,\,1056}^{HD} \cdot 1056 \\
&= 0{,}1875 \cdot 1014 + 0{,}6250 \cdot 1051 + 0{,}1875 \cdot 1056 \approx 1045
\end{aligned}$$

Valitettavasti puheentunnistuksen opettaminen on tehokkaallakin GPU:lla varsin hidasta. Lisäksi yksittäisen opetuksen lopettamisen kohtaa on hankala valita, koska yleensä huomattavaa oppimista esiintyy vielä viikonkin opetuksen jälkeen.

Yksittäisen augmentointitestin kannalta viikkojen opettaminen ei kuitenkaan ole tarpeellista, koska nopeammallakin aikataululla saadaan selville augmentointimenetelmien toimivuus, joka on myös testien tarkoitus.

Testien nopeuttamiseen käytettiin kahta eri menetelmää, joista ensimmäinen oli jokaiselle testille yhteisen esiopetetun mallin käyttö, joka tarkoittaa, että testit suoritetaan jatkaen esiopetetun perusmallin opetusta. Esiopetuksen avulla voidaan myös vakioida tuloksia keskinäisen vertailun helpoittamiseksi. Vaikka esiopetus nopeuttaa jokaista testiä, kestivät ne siitä huolimatta useita vuorokausia. [56]

Toinen testeissä käytetty menetelmä opetuksen nopeuttamiseksi oli early-stopping. Tämän avulla järjestelmää ei opeteta tiettyjen epookkien verran, vaan opetus keskeytetään kun CTC-tulokset eivät enää parane määrättyjen epookkien aikana. Early-stopping arvoksi päätettiin kuusi epookkia, eli jos CTC-tulos ei parane 48:een opetusaskeleeseen, on testi tässä vaiheessa valmis ja opetus lopetetaan.

Testien nopeuttamisen lisäksi early-stopping (ns. aikainen-pysäytys) on yleisestikin laajassa käytössä varsinkin edellisessä luvussa puhutun ylisovituksen estämiseksi. Early-stopping -menetelmää voidaan käyttää eri pysäytys-kriteereillä (*engl. stopping-criterion*), mutta tämän lopputyön testeissä sitä päätettiin käyttää edellisessä kappaleessa esitellyllä tavalla. [57]

Toinen mahdollinen kyseessä olevien testien kannalta toimiva tapa olisi voinut olla pysäyttäminen tiettyyn CTC -virheeseen, jolloin käytännössä oltaisiin mitattu ainoastaan aikaa, joka opettamiseen olisi kulunut. Käytetty menetelmä nähtiin kuitenkin monipuolisemmaksi ja perustellummaksi. [57]

Nopeuttavista menetelmistä huolimatta testeihin käytettiin yhteensä useita viikkoja. Opetuksissa käytettiin NVIDIA GeForce RTX 3090 GPU'ta, jolla yhden epookin läpikäyntiin kului noin 30 minuuttia. Jokaisessa opetuksessa epookkeja käytiin läpi, hieman testistä riippuen, noin 150 kappaletta, joten yksittäiseen testiin kului kokonaisuudessaan aikaa noin 60 tuntia.

Tässä lopputyössä malleja opetettiin eri testeissä yhteensä 31 kappaletta, joten yhteensä kaikkien testien opetukseen kului aikaa noin 80 GPU-vuorokautta. Lisäksi tehtiin esiopetus, joka tosin oli huomattavasti lyhyempi varsinaisiin testeihin verrattuna.

Todellisuudessa eri malleja ja verkkoja opetettiin merkittävästi pidempään kuin 80 GPU-vuorokautta erilaisten kokeilujen ja augmentointivariaatioiden muodossa puheentunnistusjärjestelmää varten. Lisäksi järjestelmää pyrittiin parantamaan myös tämän lopputyön ulkopuolelle rajattujen asioiden kanssa.

On myös huomioitavaa, että toisinaan malleja opetettiin usealla GPU:lla samanaikaisesti. Mainittuja GPU:ita oli käytössä yhteensä neljä kappaletta.

Järjestelmän luotettavuuden parantamisen lisäksi lopputyössä pyritään nopeuttamaan opetuksessa kuluvaa aikaa. Augmentoinnin kannalta tämä tarkoittaa sitä, että vaikka augmentoinnin tulee parantaa dataa, itse datan augmentointi-prosessi ei saa hidastaa verkkojen opettamista.

Kappaleissa 6 ja 7 vertaillaan eri testien vaatimia opetusaikoja ja pohditaan myös opetuksen mahdollista nopeutumista.

5.2 Kokeiden suoritus

Kokeet koostuivat käytännössä viidestä eri vaiheesta. Aluksi suoritetaan esiopetus, jossa puheentunnistusmalli opetetaan ilman mitään augmentointimenetelmää. Tämän avulla saadaan kaikille tuleville testeille yhteinen esiopetettu malli, josta neuroverkot jatkossa saavat alkuparametrit opetukseen.

Esiopetuksen jälkeen opetetaan järjestelmä ilman augmentointimenetelmiä. Testin tuloksista saadaan vertailutulos muita tulevia testejä varten.

Seuraavaksi testataan augmentointimenetelmiä akkumulaatio-testien avulla, joissa jokaista menetelmää testataan yksitellen. Näin varmistutaan, että yksittäinen menetelmä ei heikennä järjestelmää, joka on *conditio sine qua non* augmentointimenetelmän käyttämiseksi lopullisessa järjestelmässä. Akkumulaatiotestit eivät yleisesti ole laajassa käytössä augmentointimenetelmien testauksessa, jonka lisäksi neuroverkkojen opetus kestää jostain syystä niissä huomattavasti kauemmin kuin muissa testeissä. Pitkän opetusajan ja yleisen kirjallisuuden arvostuksen puutteen vuoksi akkumulaatiotestejä suoritettiin vain muutamia.

Augmentointimenetelmien on tarkoitus toimia yhdessä, jonka vuoksi suoritetaan ablaatiotestejä. Ablaatiotestissä jätetään yksi augmentointimenetelmä pois ja opetetaan malli kaikilla muilla menetelmillä. Akkumulaatiotestiin verrattuna ablaatiotestit antavat yhdessä tehtynä saman tiedon yksittäisen augmentointimenetelmän toimivuudesta, mutta nopeuttavat oppimista huomattavasti. Lisäksi useampaa menetelmää kerralla käyttäen, ja analysoimalla tietoa eri testien välillä, saadaan enemmän informaatiota myös yksittäisten augmentointimenetelmien toimivuudesta.

Lopuksi järjestelmä opetetaan kaikkien augmentointimenetelmien kanssa, josta saadaan lopullinen Levenshtein -etäisyys puheentunnistusjärjestelmälle.

Lopputuloksien kooste esiteltynä luvussa 6 ja lopputuloksia pohditaan luvussa 7.

5.2.1 Esiopetus

Esiopetuksen avulla hyödynnetään progressiivista opetusta, jota on käytetty esimerkiksi EfficientNetV2 -verkoissa opetuksen merkittäväksi nopeuttamiseksi. Esiopetuksen avulla saadaan alustavat painokertoimet, joiden avulla jokainen tuleva opetus saa arvot parametreihin. [56]

Painokertoimien avulla vältetään myös mahdollinen tilanne, jossa testin painokertoimien satunnaisen alkuarvon vuoksi neuroverkot eivät konvergoituisikaan kohtuullisessa ajassa. Tällöin early-stopping mahdollisesti pysäyttäisi opetuksen ja Levenshteinin-etäisyydeksi saataisiin huomattavasti huonompi tulos. [56]

Muista testeistä poiketen esiopetus suoritettiin vain kerran käyttäen 1280 optimointiaskelta. Jokaisessa askeleessa oli 800 kappaletta opetusnäytteitä, jotka olivat 16 sekunnin mittaisia. Esiopetukseen käytettiin siis ajallisesti yli 180 vuorokauden edestä dataa.

Esiopetuksesta saatavaa lopputulosta ei voida suoraan vertailla seuraavaksi saataviin tuloksiin, koska vaiheen tarkoituksena on vain nopeuttaa opettamista, eikä mallin opetusta jatkettu samalla tavalla kuin tulevien testien. Testistä saatu Levenshtein etäisyys esiteltynä alapuolella taulukossa III.

Taulukko III: Esiopetuksesta saadut testitulokset.

Esiopetus		
Menetelmä	**Ajojen Levenshtein etäisyydet**	**Harrell-Davis**
Esiopetus	1353	**1353**

Esiopetusta suoritettiin hieman yli vuorokauden ajan.

5.2.2 Vertailutestit

Kaikki tulevat testit toteutettiin luvussa 5.1 esitetyllä tavalla. Muutoksena edelliseen esiopetusvaiheeseen on siis se, että opetusta jatkettiin kunnes häviöfunktion arvot eivät pienentyneet enää kahdeksaan epookkiin.

Seuraavaksi suoritettiin kolme opetusta ilman mitään augmentointimenetelmiä, jotta saatiin vertailutulos tulevia testejä varten. Tulokset esiteltynä taulukossa IV.

Taulukko IV: Ilman augmentointimenetelmiä saadut testitulokset.

Vertailutulos				
Menetelmä	**Ajojen Levenshtein etäisyydet**			**Harrell-Davis**
Ilman augmentointeja	1056	1051	1014	**1042,70**

Ilman augmentointimenetelmiä tehdyt testit kestivät jokainen noin 4-5 vuorokautta.

Saatua Harrell-Davis -arviointia ei voi suoraan vertailla esiopetettuun tulokseen, mutta vertauskuvana nyt kun opetusta jatkettiin 4 vuorokautta saatiin noin 20% vähemmän virheitä tuottava järjestelmä.

Yksittäisten testien tuloksissa ei ole suurta hajontaa, mikä lisää testien luotettavuutta. Harrell-Davis -estimoitua tulosta käytetään tulevien testien lopputuloksien vertailuun.

5.2.3 Akkumulaatiotestit

Seuraavaksi suoritettiin augmentointimenetelmille muutamia akkumulaatiotestejä. Akkumulaatiotesteissä käytetään vain yhtä augmentointimenetelmää, joten ne antavat selkeän tuloksen käytetyn menetelmän toimivuudesta.

Ainoastaan yhden menetelmän käyttö kuitenkin hidastaa neuroverkon oppimista huomattavasti varsinkin opetuksen alkuvaiheessa. Testien hitauden sekä yleisen kirjallisuuden arvostuksen puutteen vuoksi vaiheen testejä suoritettiin yhteensä ainoastaan 6 kappaletta. Testit kestivät vähän yli 4 vuorokautta kappaleelta, joten pienestä määrästä huolimatta yhteensä testeihin käytettiin noin 25 GPU-vuorokautta. Tulokset esitelty alapuolella taulukossa V.

Taulukko V: Akkumulaatiotestien tulokset.

Akkumulaatio			
Menetelmä	**Ajojen Levenshtein etäisyydet**		**Harrell-Davis**
Vain kaiut	992	960	**976,00**
Vain taustahälyt	951	949	**950,00**
Vain säröilyt	975	-	**975,00**
Vain taajuusvastemuutokset	1010	-	**1010,00**

Yksittäinen akkumulaatiotesti kesti noin 4 vuorokautta.

Tuloksia verrattaessa ilman augmentointeja suoritettujen testien tuloksiin voidaan päätellä kaikkien augmentointimenetelmien parantavan puheentunnistusjärjestelmän robustisuutta.

Tuloksien perusteella voidaan myös päätellä taustahälyjen augmentoinnin parantavan järjestelmää eniten. On kuitenkin syytä huomata, että opetuksien pienen määrän vuoksi Levenshtein -etäisyyksille saatu virhemarginaali on korkea.

5.2.4 Ablaatiotestit

Tässä lopputyössä tutkittavien augmentointimenetelmien lisäksi ablaatiotesteissä käytettiin myös äänenvoimakkuuden sekä -korkeuden ja puhenopeuden variaatioita augmentoimaan opetusdataa. Lisäksi satunnaisuutta aiheutti jo mainittu opetusdatan satunnainen yhdisteleminen sekä dropout:in käyttö.

Ablaatiotestejä suoritettiin jokaisella augmentaatiolle neljä kappaletta, joista laskettiin jälleen Harrell-Davis estimoitu mediaani. Tulokset esitelty alapuolella taulukossa VI.

Taulukko VI: Ablaatiotestien tulokset.

Ablaatiotestit					
Menetelmä	**Ajojen Levenshtein etäisyydet**				**Harrell-Davis**
Kaiut poistettuna	931	952	941	911	**934,59**
Taustahälyt poistettuna	930	898	893	925	**911,50**
Säröilyt poistettuna	962	876	911	907	**912,13**
Taajuusvastemuut. poist.	911	885	928	923	**913,72**

Akkumulaatiotesteihin verrattuna ablaatiotestit olivat huomattavasti nopeampia niiden valmistuessa jo noin 3 vuorokaudessa.

Verrattuna kahden edellisen luvun testeihin, saadut tulokset tukevat oletusta, että eri augmentoinneista on kumuloituva hyöty. Käytännössä eri augmentointimenetelmiä kannattaa siis olla mahdollisimman suuri määrä, mutta jokaisen uuden augmentoinnin merkitys on kuitenkin nopeasti vähenevä.

Ablaatiotestien tuloksien perusteella yksittäisestä augmentointimenetelmästä kaikujen augmentoinnilla saadaan suurin hyöty puheentunnistusjärjestelmälle. Kolme muuta menetelmää tuovat testien perusteella suunnilleen yhtä suuren hyödyn.

5.2.5 Lopullinen järjestelmä

Akkumulaatio- ja ablaatiotestien perusteella voidaan todeta, että yksikään augmentointimenetelmä ei heikennä järjestelmän toimivuutta. Lopuksi voidaan kouluttaa järjestelmä kaikilla augmentaatiomenetelmillä.

Tässä lopputyössä tutkittujen augmentointimenetelmien lisäksi lopullisessa järjestelmässä käytetään myös äänenvoimakkuuden sekä -korkeuden ja puhenopeuden variaatioita augmentoimaan opetusdataa, eli samalla tavoin kun ablaatiotesteissä.

Saadut tulokset esiteltynä taulukossa VII.

Taulukko VII: Puheentunnistusjärjestelmän testaustulokset.

Lopullinen järjestelmä							
Menetelmä	**Ajojen Levenshtein etäisyydet**						**Harrell-Davis**
Kaikki käytössä	956	913	895	916	915	897	**912,20**

Yksi opetus kesti jälleen noin 3 vuorokautta. Järjestelmä opetettiin ja evaluoitiin samalla tavalla kuin edellissä testeissä, vaikka opetusta oltaisiin lopullisesssa järjestelmässä voitu jatkaa pidempäänkin. Opetus suoritettiin kuusi kertaa, joista laskettiin jälleen Harrell-Davis estimoitu mediaani.

Lasketun Harrell-Davis estimaation tuloksessa huomioitavaa on, että vaikka nyt kaikki augmentointimenetelmät olivat käytössä saatu tulos ei parantunut huomattavasti ablaatiotestien tuloksista.

Lisäksi saatu virhetulos on korkeampi kuin taustahälyt poistettuna saadun ablaatiotestin tulos, mutta kuitenkin akkumulaatiotesteissä paras tulos saatiin juuri ainoastaan taustahälyjä käyttämällä. Tätä tulosta ja syitä siihen arvioidaan vielä myöhemmin luvussa 7.

6 Lopputulokset

Tässä luvussa esitetään tiivistetysti käytetty järjestelmä sekä testeistä saadut olennaisimmat lopputulokset. Saatuja tuloksia ja niiden merkittävyyttä pohditaan vielä luvussa 7.

Alapuolella taulukossa VIII opetetun puheentunnistusjärjestelmän olennaisimmat parametrit.

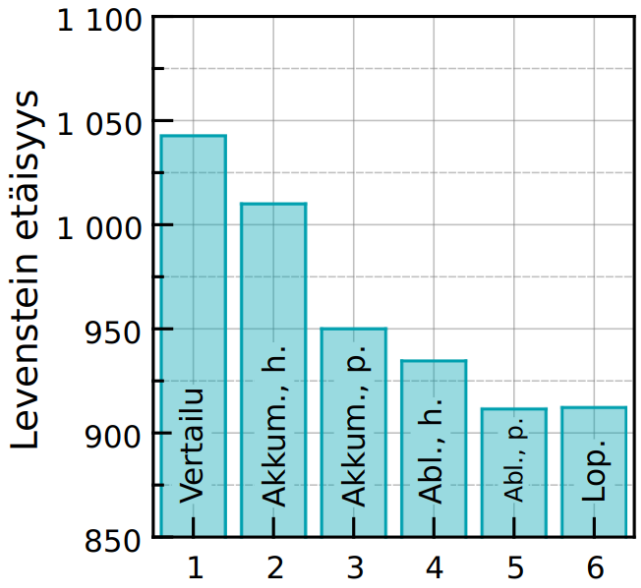
Taulukko VIII: Puheentunnistusjärjestelmän olennaisimmat hyperparametrit.

Parametri	**Tarkennus**
Arkkitehtuuri	Conformeri (Varioitu) [25]
Aktivointifunktio	Softmax
Optimointialgoritmi	Adadelta [54]
Opetusaskel	0.05 (0.005)
Dropout	10%
Virhefunktio	CTC [31]
Evalutointi	Levenshteinin etäisyys
Regularisointi	$L_1 = 0,\ L_2 = 10^{-6}$

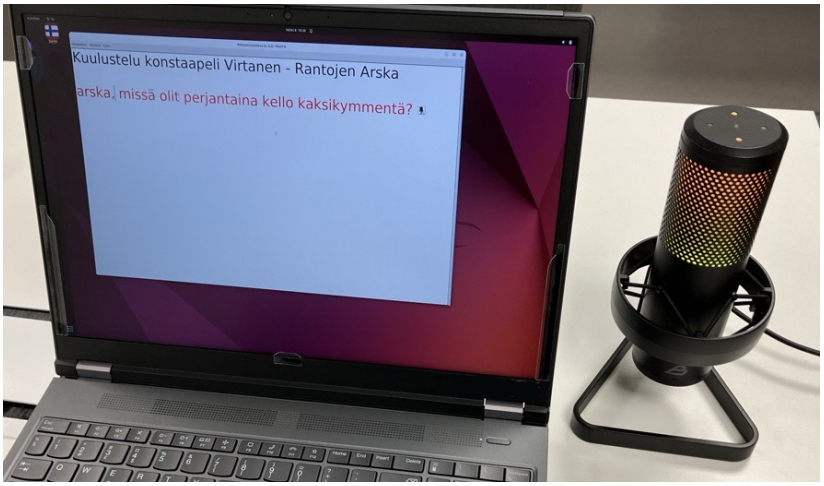
Taulukossa IX esiteltynä tiivistetyt lopputulokset ja kuvassa 25 taulukon tulokset pylväsdiagrammina.

Taulukko IX: Lopputyön testien tiivistetyt tulokset.

Testi	**Augmentointi**	**Harrell-Davis**
1. Vertailu	Ei mitään käytössä	1042,70
2. Akkumulaatio, huonoin	Vain taajuusvaste	1010,00
3. Akkumulaatio, paras	Vain taustahälyt	950,00
4. Ablaatio, huonoin	Kaiut poistettuna	934,59
5. Ablaatio, paras	Taustahälyt poistettuna	911,50
6. Lopullinen	Kaikki käytössä	912,20



Kuva 25: Pylväsdiagrammi taulukon IX lopputuloksista.



Kuva 26: Esimerkki opetetun puheentunnistusjärjestelmän käytöstä.

Kokonaisuudessaan lopputyön testien tuloksiin voidaan olla tyytyväisiä. Kuvassa 26 esimerkki puheentunnistusjärjestelmän ulossyötteestä, kun tunnistettavaksi lauseeksi sanottiin ”Arska, missä olit perjantaina kello kaksikymmentä?”.

Lopputuloksien perusteella voidaan päätellä kaikujen augmentaation olevan järjestelmää eniten parantava menetelmä. Taajuusvastemuutoksien voidaan taas päätellä olevan järjestelmää vähiten parantava menetelmä. Syitä edellä mainittuihin pohditaan vielä seuraavassa luvussa.

Vertailtaessa opetusaikoja augmentoidun ja augmentoimattoman neuroverkkomallin välillä huomattiin, että augmentointi auttoi järjestelmää oppimaan alussa nopeammin, jolloin myös parempiin tuloksiin päästiin huomattavasti nopeammassa ajassa. Tämän ansiosta myös opetukset valmistuivat nopeammin vertailtaessa akkumulaatio- ja ablaatiotestejä keskenään.

Augmentointimenetelmät eivät myöskään käyttäneet niin paljon CPU -laskentaa, että ne olisivat koodillisesti hidastaneet järjestelmän oppimista. Akkumulaatiotestit kestivät keskiarvolta yli vuorokauden pidempään verrattuna ablaatiotesteihin, joka myös tukee nopeampaa oppimista augmentointimenetelmiä käytettäessä.

7 Yhteenveto

Tämän opinnäytetyön ulkopuolella puheentunnistusjärjestelmää on opetettu samoilla menetelmillä muutaman vuorokauden sijasta useampia viikkoja, jolloin Levenshtein -etäisyydeksi on saatu 737.

Muutamankin vuorokauden opetuksen jälkeen Levenshtein etäisyys kuitenkin korreloi vahvasti järjestelmän robustiuden kanssa, joten testien kannalta opetusta ei ole syytä jatkaa viikkoja.

Testien perusteella yksittäisistä augmentointimenetelmistä eniten järjestelmää paransi akkumulaatiotestien perusteella taustahälyt ja ablaatiotestien perusteella kaikujen augmentointi. Vähiten järjestelmää paransi akkumulaatiotestien perusteella taajuusvastemuutokset ja ablaatiotestien perusteella taustahälyt.

Mielenkiintoisena tuloksena saatiin taustahälyjen ablaatiotestille parempi Harrell-Davis -estimoitu virhetulos kuin lopullisesti opetetulle puheentunnistusjärjestelmälle, mutta silti akkumulaatiotesteistä juuri taustahälyille paras tulos.

Ristiriitaisuuksia voidaan selittää ainakin ainoastaan yhdellä tehdyllä taustahälyjen akkumulaatiotestillä, jonka vuoksi sen virhe voidaan arvioida suhteellisen suureksi. Akkumulaatiotestien tarkoitus ei olekaan antaa tarkkaa kuvaa siitä kuinka hyvin menetelmä parantaa järjestelmän robustisuutta, mutta jo muutama testi kertoo, että kyseinen augmentointimenetelmä ei ainakaan huononna sitä.

Lisäksi lopullisen järjestelmän opetus päätettiin nyt samoilla parametreillä kun testeissä. Todellisuudessa järjestelmää opettettaisiin paljon pidempään, jolloin voidaan myös olettaa augmentointimenetelmien pienentävän enemmän lopullista saatavaa Levenshtein -virhetulosta. Edellä mainittuun liittyen testeissä käytetty early-stopping voi helposti keskeyttää akkumulaatiotestin liian aikaisin.

Levenshtein etäisyydeksi laskettu tulos riippuu täysin käytetystä testidatasta. Nyt käytetty testidata oli studio-olosuhteissa tallennettua puhetta ilman taustahälyjä tai kaikuja. Tämän vuoksi käytetyt augmentoinnit olisivat myös hyvin voineet heikentää järjestelmän virhetulosta. Todellisissa sovelluskohteissa voimme kuitenkin aina olettaa olevan käytettyjen augmentointimenetelmien kaltaista ääntä.

Järjestelmän robustiuden kehittämisen lisäksi havaittiin, että paremmalla opetusdatalla neuroverkko kehittyy varsinkin aluksi nopeammin, jolloin augmentointimenetelmät myös nopeuttavat neuroverkkojen oppimista.

Kuitenkin, jokaisen ajon viemä pitkä aika valitettavasti rajoittaa satojen testien ajamista sekä käyttämästä perinteisiä tutkimusmenetelmiä, kuten varianssien, keskihajontojen, tai tilastollisten merkitsevyyksien p-arvojen mittaamista.

Alkuperäisen conformer -arkkitehtuurin variointi johtui ennen tämän lopputyön testejä suoritetuista kokeiluista, joiden aikana havaittiin leveämmän ja matalamman neuroverkon toimivan paremmin kuin mitä Gulati et al. käyttivät omassa artikkelissaan. Tämä luultavasti johtuu siitä, että tämän opinnäytetyön puheentunnistusjärjestelmällä pyritään foneettisesti aitoon puheentunnistukseen sen sijaan, että puhetta korjattaisiin kielioppiin tai sanakirjoihin pohjautuen.

Kokonaisuudessaan lopputuloksiin voidaan olla tyytyväisiä. Työssä onnistuttiin kehittämään ja optimoimaan realistiselta kuulostavia augmentointimenetelmiä, jotka paransivat puheentunnistusjärjestelmän robustisuutta.

Suoritettujen kokeiden perusteella voidaan todeta augmentoinnilla olevan merkittävä rooli puheentunnistusjärjestelmien robustiuden parantamisessa. Tulevaisuudessa koneoppimismalleja, mukaanlukien puheentunnistusjärjestelmiä, tullaan varmasti entisestään kehittämään ja apuna tullaan käyttämään augmentoitua dataa.

Viitteet

[1] A. K. Singh, S. Priyanka ja K. Nathwani, "Using Deep Learning Techniques and Inferential Speech Statistics for AI Synthesised Speech Rec." (CoRR) (2021).

[2] C. Shorten ja T. M. Khoshgoftaar, Journal of Big Data **6**, 1 (2019).

[3] J.-S. Lee ja C. Hoon, *Adaptive Decision Fusion for Audio-Visual Speech Recognition* (*Speech Recognition*), InTechOpen, (2008), pp. 275–296.

[4] T. Virtanen, R. Singh ja B. Raj, *Techniques for noise robustness in automatic speech recognition*, Wiley, (2012).

[5] J. E. K. Foreman, *Sound Analysis and Noise Control*, Van Nostrand Reinhold, (1990).

[6] J. J. Eggermont, *Hearing loss: causes, prevention, and treatment*, Academic Press, (2017).

[7] J. Eargle, *The Microphone Book*, 2nd ed., Focal Press, (2004).

[8] L. L. Beranek ja T. Mellow, *Acoustics: Sound Fields and Transducers*, 1st ed., Academic Press, (2012).

[9] V. Pulkki ja M. Karjalainen, *Communication Acoustics: An Introduction to Speech, Audio and Psychoacoustics*, 3rd ed., Wiley, (2015).

[10] H. Wyatt ja T. Amyes, *Audio post production for television and film: an introduction to technology and techniques*, 3rd ed., Focal Press, (2005).

[11] C. E. Shannon, Bell System Technical Journal **27**, 379 (1948).

[12] D. Wu, B. Li ja H. Jiang, *Normalization and Transformation Techniques for Robust Speaker Recognition* (*Speech Recognition*), InTechOpen, (2008), pp. 311–330.

[13] J. H. Connolly *et al.*, International Journal of Man-Machine Studies **24**, 611 (1986).

[14] P. Berjon, A. Nag ja S. Dev, Soft Computing Letters **3**, 100018 (2021).

[15] A. K. Das ja R. Naskar, Biomedical Signal Processing and Control **90**, (2024).

[16] M. Wölfel ja J. McDonough, *Distant Speech Recognition*, Wiley, (2009).

[17] S. S. Stevens, J. Volkmann ja E. B. Newman, The Journal of the Acoustical Society of America **8**, 185 (1937).

[18] N. Singh, R. A. Khan ja R. Shree, International Journal of Computer Applications **54**, 9 (2012).

[19] D. Prabakaran ja S. Sriuppili, Journal of Physics: Conference Series **1717**, 012009 (2021).

[20] A. Yarali, *Intelligent Connectivity*, Wiley, (2021).

[21] P. Gupta ja N. K. Sehgal, *Introduction to Machine Learning in the Cloud with Python*, Springer, (2021).

[22] W. Pietsch, *Big data*, Cambridge University Press, (2021).

[23] V. Ivan *et al.*, *Python Deep Learning*, 2nd ed., Packt Publishing Limited, (2019).

[24] G. Zietsman ja R. Malekian, Journal of Internet Technology **23**, 1527 (2022).

[25] A. Gulati *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition" (arXiv) (2020).

[26] K. He *et al.*, "Deep Residual Learning for Image Recognition" (CoRR) (2015).

[27] A. Vaswani *et al.*, "Attention Is All You Need" (CoRR) (2017).

[28] Y. Yang, P. Wang ja D. Wang, "A Conformer Based Acoustic Model for Robust Automatic Speech Recognition" (arXiv) (2022).

[29] J. Zuluaga-Gomez *et al.*, Aerospace **10**, 490 (2023).

[30] V. N. Sukhadia ja S. Umesh, "Domain Adaptation of low-resource Target-Domain models using well-trained ASR Conformer Models" (IEEE SLT) 295 (2023).

[31] A. Graves *et al.*, ACM International Conference Proceeding Series **148**, 369 (2006).

[32] H. Li ja W. Wang, Pattern Recognition **105**, 107392 (2020).

[33] N. Srivastava *et al.*, Journal of Machine Learning Research **15**, 1929 (2014).

[34] R. Somnath, "Semantic-WER: A Unified Metric for the Evaluation of ASR Transcript for End Usability" (CoRR) (2021).

[35] L. Yujian ja L. Bo, IEEE Transactions on Pattern Analysis and Machine Intelligence **29**, 1091 (2007).

[36] Y. Li, X. Yu ja N. Koudas, Proceedings of the VLDB Endowment **14**, 1832 (2021).

[37] T. Iqbal *et al.*, "Enhancing Audio Augmentation Methods with Consistency Learning" (IEEE ICASSP) 646 (2021).

[38] D. Wang *et al.*, Digital signal processing **129**, 103681 (2022).

[39] C. Shorten, T. M. Khoshgoftaar ja B. Furht, Journal of big data **8**, 101 (2021).

[40] L. Garcia *et al.*, *Histogram Equalization for Robust Speech Recognition* (*Speech Recognition*), InTechOpen, (2008), pp. 23–44.

[41] G. Pironkov, S. Dupont ja T. Dutoit, "Investigating the impact of the training data volume for robust speech recognition using multi-task learning" (ISSPIT), 382 (2017).

[42] IEC (60268-16), Sound system equipment - Part 16: "Objective rating of speech intelligibility by speech transmission index", 2011.

[43] H. Liu *et al.*, Applied Acoustics **167**, 107400 (2020).

[44] R. Abdullah, S. Ismail ja N. N. Dzulkefli, Journal of Physics: Conference Series **1529**, 022031 (2020).

[45] C. F. Eyring, The Journal of the Acoustical Society of America **1**, 217 (1930).

[46] M. Oren ja S. K. Nayar, International Journal of Computer Vision **14**, 227 (1995).

[47] R. Scheibler, E. Bezzam ja I. Dokmanić, "Pyroomacoustics: A Python package for audio room simulations and array processing algorithms" (IEEE ICASSP) (2017).

[48] Z. Liu *et al.*, *Direct filtering for air- and bone-conductive microphones* (*6th Workshop on Multimedia Signal Processing*), IEEE, (2004), pp. 363–366.

[49] B. Acker-Mills, J.-M. Adrianus ja A. W. A., Aviation, space, and environmental medicine **77**, 26 (2004).

[50] J. Havakka ja I. Huhtakallio, *Keskustelu Patria Aviation Oy ja Savox Communications Oy, (22.1.2024).*

[51] S. N. Awan *et al.*, American Journal of Speech-Language Pathology **31**, 959 (2022).

[52] C. Bregler ja Y. Konig, *"Eigenlips" for robust speech recognition* (*Proceedings of ICASSP*), IEEE, (1994), pp. II/669–II/672.

[53] M. Slaney, "Normalizing Non-Linear Speech Speed for Maintaining Listener Comprehension at Increased Playback Speeds" (TDC), (2021).

[54] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method" (arXiv), (2012).

[55] B. Shulkin ja S. Sawilowsky, "Estimating A Population Median With A Small Sample" (Festschrift), 143 (2009).

[56] M. Tan ja Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training" (arXiv) (2021).

[57] L. Prechelt, *Early Stopping — But When?* (*Neural Networks: Tricks of the Trade*), Springer Berlin Heidelberg, (2012), pp. 53–67.