OULUN YLIOPISTO
UNIVERSITY of OULU

TIETOTEKNIIKAN OSASTO

Olli Korhonen, Lari-Matias Orjala, Eero Paavola

ÄÄNEN PIIRTEIDEN VERTAILU IHMISÄÄNEN LUOKITTELUSSA

Kandidaatintyö
Tietotekniikan koulutusohjelma
Toukokuu 2015

Korhonen O, Orjala L-M, Paavola E. (2015) Äänen piirteiden vertailu ihmisäänen luokittelussa. Oulun yliopisto, tietotekniikan osasto. Kandidaatintyö, 26 s.

TIIVISTELMÄ

Puheentunnistusta hyödyntävät sovellukset ovat viime vuosina yleistyneet ihmisten arkielämässä. Tavallisesti puheentunnistus perustuu äänenpiirteiden vertailuun. Piirteitä ovat muun muassa äänen taajuus- ja energiasisältö. Käytettävien piirteiden valinnalla on merkittävä vaikutus puheentunnistuksen laadussa, koska eri piirteet kuvaavat äänen eri ominaisuuksia. Tässä työssä keskitytään eri piirteiden käyttökelpoisuuden vertailuun ihmisen äänen tunnistuksessa. Ääni tulee tunnistaa ihmisen puheeksi, ennen kuin kannattaa käyttää algoritmeja, jotka etsivät äänestä esimerkiksi sanoja tai tunnetiloja. Tämän idean pohjalta toteutettiin binäärinen luokittelija, joka arvioi, onko ääni ihmisen puhetta vai ei. Luokittelija toteutettiin käyttäen yleisimpiä äänen analyysimenetelmiä, kuten piirrevektoreita ja k-NN luokittelualgoritmia. Lisäksi suoritettiin testit, joilla tutkittiin luokittelun tarkkuutta. Testien perusteella MFCC oli testatuista piirteistä paras ihmisäänen luokittelussa. Lisäksi huomattiin, että piirrevektorin sisältö vaikuttaa luokittelun tarkkuuteen enemmän kuin sen pituus.

Avainsanat: puheentunnistus, piirreirrotus, luokittelu, k-NN, MFCC

Korhonen O, Orjala L-M, Paavola E. (2015) Comparison of sound features for human sound classification. University of Oulu, Department of Computer Science and Engineering. Bachelor's Thesis, 26 p.

ABSTRACT

Applications taking advantage of automatic speech recognition (ASR) have become increasingly common in people's everyday lives. Usually speech recognition is achieved by comparing the sound's features. Some example features include the spectral and energy content of the signal. The choice of features impacts greatly the performance of a speech detection system, because different sound features describe different sound properties. The goal of this study is to compare different features and their suitability in detecting human voice. Before proceeding with algorithms that try to find words or other meanings from the sound, it should be confirmed that the sound is human voice. Following this principle, a binary classifier is implemented that can evaluate whether or not a sound is human voice. The classifier is implemented using some common methods such as feature extraction and k-NN classification. In addition tests are carried out to measure the accuracy of the classification. The tests showed that MFCC performed the best in our feature set. Furthermore, it was observed that the content of the feature vector matters more than its length.

Key words: speech recognition, feature extraction, classification, k-NN, MFCC

SISÄLLYSLUETTELO

ALKULAUSE

Haluamme kiittää kandidaatintyömme ohjaajia, prof. Juha Röningiä ja Teemu Tokolaa, perheitämme ja kurssitovereitamme arvokkaista neuvoista ja kannustuksesta.

Oulu, toukokuu 21. 2015

Olli Korhonen
Lari-Matias Orjala
Eero Paavola

LYHENTEIDEN JA MERKKIEN SELITYKSET

ANN	Artificial neural network (keinoneuroverkko)
BER	Band energy ratio
DCT	Discrete Cosine transform (diskreetti kosinimuunnos)
DFT	Discrete Fourier transform (diskreetti Fourier-muunnos)
FFT	Fast Fourier transform (nopea Fourier-muunnos)
k-NN	k-Nearest Neighbour algoritmi
MFCC	Mel-frequency cepstral coefficient
NB	Naive Bayes algoritmi
SC	Spectral centroid
SF	Spectral flux
SRO	Spectral roll off
STE	Short term energy
SV	Spectral variance
SVM	Support vector machine (tukivektorikone)
WAV	Waveform Audio File Format aka. WAVE
ZCR	Zero crossing rate

1. JOHDANTO

Tietotekniikka on helpottanut ihmisten arkea jo monta vuosikymmentä eikä kehitykselle näy loppua. Nykyajan ihmiset ovat jo tottuneet käyttämään erilaisia laitteita ja sovelluksia, toinen toistaan kehittyneempiä ja monimutkaisempia. Teknologiasta on tullut niin merkittävä osa ihmisten elämää, että uusia mullistavia sovelluksia otetaan avoimesti vastaan. Tietotekniikan lisääntyessä ihmisten arjessa kasvaa luonnollisesti myös vuorovaikutus eri laitteiden ja sovellusten kanssa. Ihminen hallitsee hyvin tietokoneen käytön, mutta miten saada tietokone ymmärtämään ihmistä? Nykyään tutkimuksen ja kaupallisen tuotekehityksen keskipisteessä ovat puheen- ja kuvantunnistukseen, sekä kosketukseen ja käden liikkeisiin perustuvat sovellukset.

Erityisesti jatkuvan puheen tunnistusta toteuttavat sovellukset ovat nousseet suosioon 2010-luvun vaihteessa esimerkiksi Google Now, Apple Siri ja Microsoft Cortana -mobiilipalveluiden myötä, joita voidaan käyttää apuna esimerkiksi paikannuksessa, sään tarkistuksessa ja muissa älykkäissä hakusovelluksissa[1-3]. Sovellukset pystyvät lukuisiin tehtäviin, mutta niiden käyttökohteita pyritään kehittämään jatkuvasti lisää ja laatua pyritään nostamaan entisestään. Useisiin puheentunnistukseen liittyviin ongelmiin ei ole olemassa yhtä oikeaa ratkaisua, joten puheentunnistuksen tutkiminen ja kehittäminen on jatkuva prosessi.

Tavallisesti puheentunnistusjärjestelmän tavoitteet ovat tunnistaa käyttäjän puhe, ymmärtää puheen sisällöstä käyttäjän valitsema tehtävä sekä toteuttaa se. Kuten ihminen kuunnellessaan toisen ihmisen puhetta, niin myös tietokoneen täytyy pystyä erottamaan ihmisen ääni ympäristön muista äänistä. Tietokoneen analyysin kohteena ovat ihmisen puheen olennaiset piirteet eli ominaisuudet, jotka erottavat ihmisen äänen muista äänistä kuten eläimistä, laitteista sekä yleisestä taustakohinasta.

Äänen tunnistus tapahtuu luokittelemalla havaittu ääni oikeaan kategoriaan äänen piirteiden avulla. Tästä syystä kattavan vertailutietokannan ylläpitäminen on tärkeää, jotta ääni luokiteltaisiin oikein. Esimerkiksi lintujen äänien erottelussa oikean lintulajin tunnistuksen takaamiseksi täytyy tietokannassa olla kyseisen lintulajin ääninäytteitä, mutta lisäksi myös muiden vertailukelpoisten lintulajien ääninäytteitä, kuten Seppo Fagerlund väitöskirjassaan esittää[4]. Työssä tutustutaan ihmisen äänentunnistukseen sekä tunnettuihin äänen piirteisiin ja niiden käytettävyyteen ihmisen äänen tunnistusjärjestelmissä.

2. ÄÄNI JA SEN ANALYSOINTI

Ääniaalloiksi kutsutaan ilmassa tai muussa väliaineessa tapahtuvaa pitkittäistä mekaanista aaltoliikettä, jonka ihminen voi kuulla. Väliaineen molekyylit värähtelevät tasapainoasemansa ympärillä tietyllä taajuudella, joka mitataan hertseinä (Hz). Taajuuden lisäksi ääniaallon ominaisuuksia kuvaa amplitudi, joka kertoo ääniaallon aiheuttaman poikkeaman suhteessa tasapainoasemaan. Koska ääni on aaltoliikettä, sillä on aaltoliikkeen yleiset ominaisuudet, joita ovat muun muassa heijastuminen, diffraktio, interferenssi ja sironta. Äänen amplitudi ilmenee ilmakehässä ilmanpaineen muutoksina. Yleensä mielekkäämpi mittari ääniaallon voimakkuudelle onkin amplitudin sijaan äänenpaine, jota ihmisen korvat voivat havainnoida. Äänenpainetta kuvaava yksikkö on kaavassa 1 esitetty logaritminen desibeli [dB], jota käytetään ilmaisemaan äänenvoimakkuutta. Referenssi-ilmanpaine p_{ref} on normaalisti 20µPa.

$$dB = 20 \log \frac{p}{p_{ref}} \tag{1}$$

Ihminen havainnoi ääniaaltoja pääasiassa kuuloaistin avulla, mutta tarpeeksi voimakkaat tai matalat ääniaallot voi havaita värähtelynä myös tuntoaistin avulla. Yleisesti mielletty ihmisen kuuloalue on 20–20000 Hz[5], mutta vanhenemisen myötä kuuloalueen yläraja yleensä laskee. Äänen voimakkuutta ihminen aistii korviensa avulla, jotka havaitsevat ilmanpaineen muutoksia eli äänenpainetta.

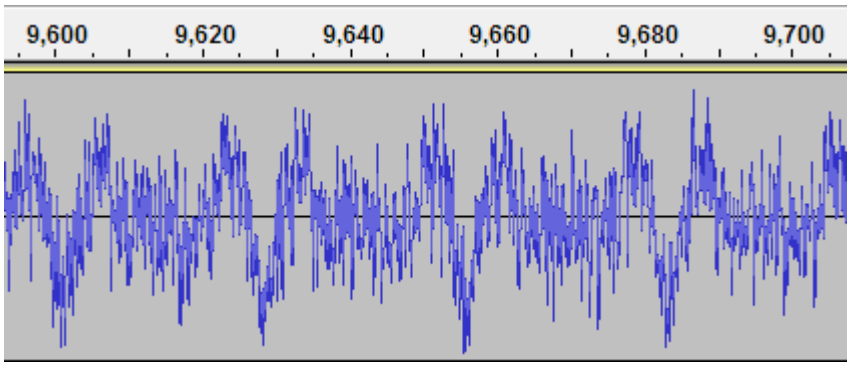
Ihmisen kuuloaisti toimii logaritmisesti, minkä takia desibelejä käytetään yksikkönä ääniä mitattaessa. Kuuloaistin logaritmisuus tarkoittaa, että äänenpaineen kymmenkertaistuminen kuulostaa ihmiselle äänenvoimakkuuden tuplaantumiselta. Ihmisen kuulokynnykseksi on valittu 0 dB. Korva ei havaitse ääniä yhtä herkästi kaikilla taajuuksilla. Herkimmillään kuuloaisti on taajuusalueella 2–5 kHz[6]. Tämä tarkoittaa sitä, että äänenpaineen ollessa vakio ihminen kuulee ääniä herkemmin tällä alueella kuin muilla taajuuksilla. Ihmisen tuottamat äänet ovat taajuusalueella 80–10 000 Hz. Suurin osa puheen energiasta ja informaatiosta keskittyy alueen matalampaan päätyyn, sadoista hertseistä muutamaan kilohertsiin. Vokaalit sisältävät yleisesti enemmän matalia taajuuksia, konsonantit puolestaan korkeampia. Taulukossa 1 on esitetty yleisesti tunnettujen äänten intensiteettitasoja ja keskimääräisiä taajuuksia.

Taulukko 1. Yleisesti tunnettuja äänten intensiteettitasoja ja taajuuksia.

Äänitapahtuma	**Intensiteettitaso (dB)**	**Keskimääräinen taajuus (Hz)**
normaali keskustelu	40–60	300
koiran haukunta	80	500
moottorisaha	100	1000
suihkumoottori	120	4000

Mikrofonien avulla äänisignaali voidaan muuttaa sähköiseksi signaaliksi. Yksinkertaisen mikrofonin toiminta perustuu samaan periaatteeseen kuin korvan: ilmanpaineen muutokset liikuttavat ohutta kalvoa, jonka liike välittyy jännitteen muutoksiksi. Tästä saadaan tietoon ääniaallon amplitudi ajan funktiona, mistä voidaan puolestaan johtaa uusia suureita, kuten taajuus. Tietokone tallentaa mikrofonin tuottamasta äänidatasta näytepisteitä vain tietyin välein eli tietyllä taajuudella. Tätä taajuutta kutsutaan näytteistystaajuudeksi. Esimerkiksi CD-levyillä yleisesti käytössä

oleva näytteistystaajuus on 44,1 kHz. Äänen analysoinnissa riittävä taajuus on huomattavasti pienempi, esimerkiksi 22 050 Hz tai 16 000 Hz.



Kuva 1. 100 millisekunnin aaltomuoto musiikkikappaleesta

Kuvassa 1 näkyvä aaltomuoto on äänisignaalin esitys aikatasossa. Tästä esitystavasta voidaan analysoida tiettyjä äänisignaalin ominaisuuksia, kuten missä kohtaa signaalia on hiljaista ja millainen on signaalin verhokäyrä. Aikatason esitystavasta on kuitenkin vaikeaa, saada hyödyllistä tietoa äänestä, kuten mitä taajuuksia se sisältää. Tähän ongelmaan ratkaisuna on siirtyminen aikatason esityksestä taajuustason esitykseen. Siinä missä aikatason esitys kuvaa signaalin amplitudin ajan funktiona, taajuustason esitys kertoo signaalin amplitudin taajuuden funktiona. Äänisignaalin tapauksessa siirtyminen aikatasosta taajuustasoon tehdään Fourier'n muunnoksella. Digitaalisessa signaalinkäsittelyssä käytetään tavallisesti diskreettiä Fourier'n muunnosta (DFT), jonka avulla mikä tahansa jaksollinen signaali voidaan esittää sinimuotoisten signaalien sarjakehitelmänä. Kaava 2 kuvaa näytepisteessä n laskettua DFT:a. Tietokoneet laskevat DFT:n tehokkaasti käyttäen hyväksi FFT-algoritmia (engl. Fast Fourier transform).

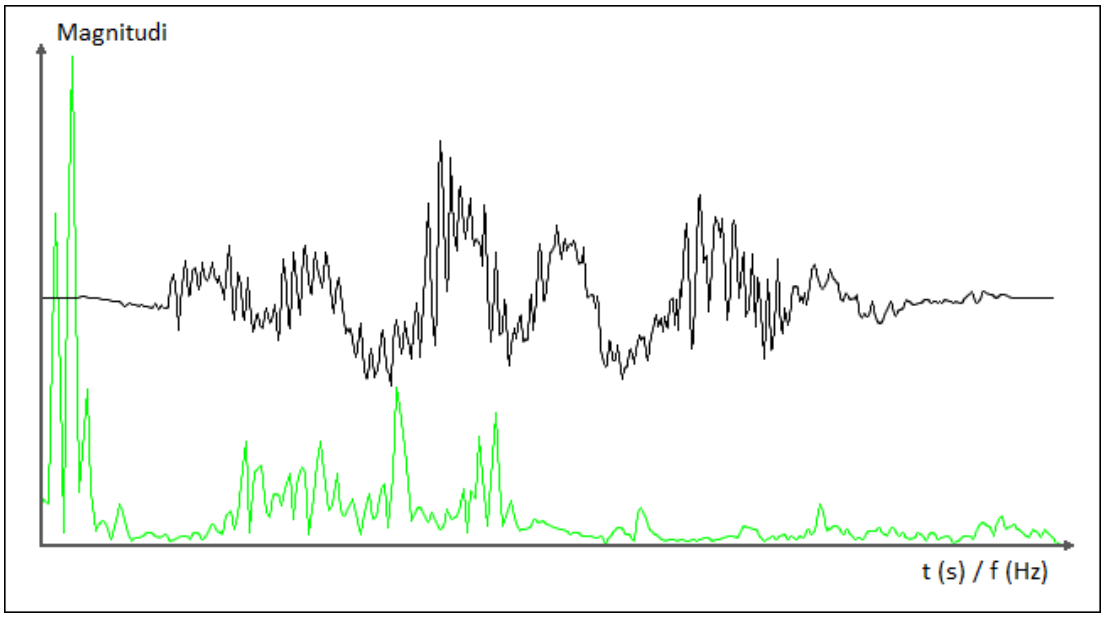
DFT:n tulokseen vaikuttavat merkittävästi signaalin näytteistystaajuus ja näyteikkunan koko. DFT voi laskea taajuussisällön vain Nyquistin taajuuteen asti, mikä on puolet näytteistystaajuudesta. DFT:n tulosten taajuuskorien resoluutioon puolestaan vaikuttaa näytteiden määrä ikkunassa.

$$X_k = \sum_{n=0}^{N-1} x(n) e^{-2\pi i n k / N} \ , k \in \mathbb{Z} \tag{2}$$

Koska DFT toimii vain jaksollisille ja jatkuville signaaleille, ei sitä voi sellaisenaan käyttää äänisignaaliin. Avuksi otetaan ikkunoinniksi kutsuttu menetelmä, jonka avulla signaalista käsitellään pieni osa kerrallaan. Tätä osaa kutsutaan kehykseksi. Ikkunafunktio puolestaan on funktio, joka on nolla muualla paitsi halutulla alueella. Kun kehyssignaali kerrotaan ikkunafunktiolla, menee signaalin arvo kehyksen alussa ja lopussa nollaan. Tämä poistaa epäjatkuvuuskohdat, ja peräkkäin laitetut ikkunoidut kehykset muodostavat jaksollisen signaalin. Koska ikkunointi heikentää signaalia kehyksen reunoilla, peräkkäiset kehykset valitaan yleensä siten, että ne menevät hieman päällekkäin. Kuvassa 2 on esitetty ikkunoitu kehys äänisignaalista ja sen pohjalta laskettu amplitudispektri.

Äänen analysoinnissa kehyksen koko riippuu tehtävästä, esimerkiksi puheentunnistuksessa sopiva kehyskoko on n. 20–30 ms[7]. Kehyksen pituus tulee valita tarkasti; liian pitkä kehys laskee ajallista resoluutiota, kun taas liian lyhyt kehys

huonontaa taajuusresoluutiota. Kehyksen pituuden ja näytteistystaajuuden valinta tulee tehdä siten, että sekä aika- että taajuusresoluutio ovat tyydyttävällä tasolla.



Kuva 2. 23 ms pituinen ikkunoitu kehys puhetta sisältävästä tiedostosta (musta viiva), sekä FFT:n tuottama amplitudispektri (vihreä).

2.1. Piirreirrotus

Suurin osa ääntä analysoivista systeemeistä toimii samalla periaatteella: äänisignaalista saatua ikkunoitua dataa käsitellään erinäisillä algoritmeilla aika- tai taajuustasossa. Yksittäisen piirreirrotusalgoritmin tehtävänä on laskea ikkunan datan pohjalta yksi tai useampi luku. Laskettuja ominaislukuja tai -vektoreita kutsutaan piirteiksi (engl. feature). Piirteet pyrkivät kuvaamaan mahdollisimman hyvin tiettyä yksittäistä äänen ominaisuutta, kuten energiasisältöä tai taajuusjakaumaa. Äänentunnistuksessa on vakiintunut käyttöön useita piirteitä, jotka on havaittu hyödyllisiksi ihmisäänen kuvaamisessa (taulukko 2).

Piirreirrotus on hahmontunnistuksen vaihe, jossa data kuvataan piirreavaruuteen. Siinä alkuperäisestä datamäärästä saadaan hyödyllistä informaatiota sisältävä pieni sarja piirteitä, jotka yhdessä muodostavat piirrevektorin (engl. feature vector). Oikein valittu piirrevektori kuvastaa datan tärkeimpiä ominaisuuksia. Piirrevektoreiden vertailu on yhdistävä tekijä lähes kaikille luokittelumetodeille.

Taulukko 2. Lähteissä esiintyneitä piirteitä ja niiden kuvaukset.[8, 9]

Piirre	Taso	Selitys
MFCC	taajuus	Kuvaa ihmiskuulolle tärkeiden taajuuskaistojen energiasisältöä.
spektrin keskipiste (engl. spectral centroid)	taajuus	Kehyksen keskiarvoinen taajuus.
spektrin hajonta (engl. spectral variance)	taajuus	Kehyksessä esiintyvien taajuuksien varianssi.
spektrin vinous (engl. spectral roll off)	taajuus	Ilmaisee taajuuden, jonka alapuolella 95 % kehyksen energiasta sijaitsee.
spektraalimuutos (engl. spectral flux)	taajuus	Mittaa peräkkäisten kehysten amplitudispektrien eroa.
kaistaenergiasuhde (engl. band energy ratio)	taajuus	Valitun (esimerkiksi ihmisäänen) taajuusalueen energia verrattuna kehyksen kokonaisenergiaan.
lyhyen aikavälin energia (engl. short term energy)	aika	Kehyksen kokonaisenergia.
nollanylitystaajuus (engl. zero-crossing rate)	aika	Määrittää kuinka monta kertaa signaalin etumerkki vaihtuu kehyksessä.

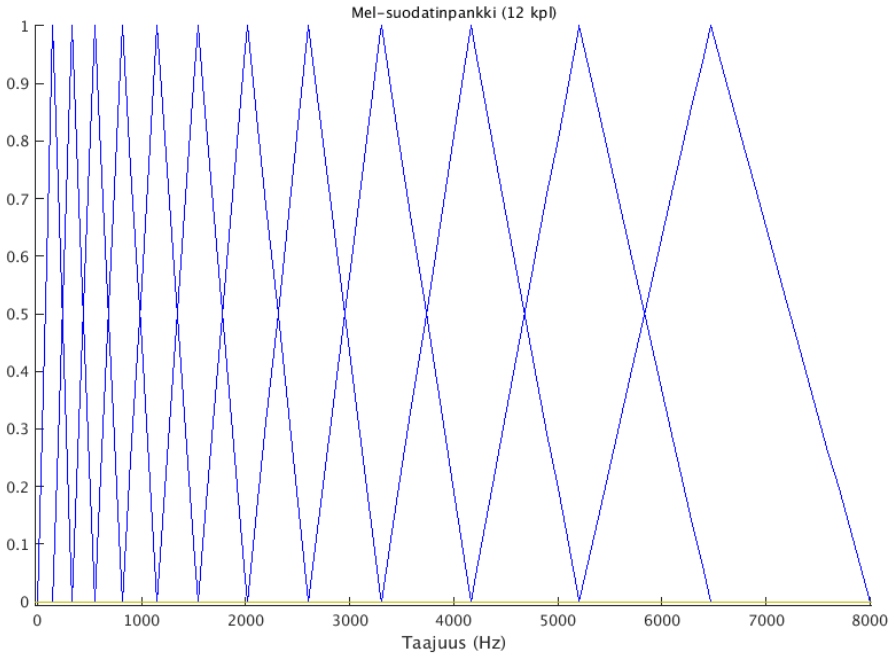
2.1.1. MFCC

Nykyään yksi käytetyimmistä piirteistä äänten analysoinnissa on nimeltään MFC-kertoimet (engl. MFCC, mel frequency cepstral coefficients), joita käytetään hyväksi niin puheen kuin ympäristöääntenkin tunnistuksessa[10]. MFC-kertoimet kehitettiin alun perin puheentunnistusta varten, mutta ne ovat osoittautuneet hyödyllisiksi myös muissa äänentunnistussovelluksissa.

Koska ihminen ei kuule sävelkorkeuksia lineaarisesti, on kehitetty erilaisia asteikkoja kuvaamaan ihmisen kuulemaa sävelkorkeutta suhteessa taajuuteen. Puheentunnistuksessa hyödylliseksi on osoittautunut Mel-asteikko[11]. Mel-asteikon referenssipiste hertseihin on 1000 meliä, joka vastaa 1000 Hz:n taajuista ääntä. 2000 meliä kuvaa ääntä, jonka äänenkorkeus on ihmisen mielestä kaksi kertaa niin korkea kuin 1000 meliä. Hertseissä 2000 meliä on noin 3000 Hz. MFC-kertoimet käyttävät hyväksi mel-skaalaa. Muunnoskaavat Hz-asteikon ja mel-asteikon välillä on esitetty kaavoissa 3 ja 4.

$$m = 1127 \ln 1 + \frac{f}{700} \tag{3}$$

$$f = 700(e^{m/1127} - 1) \tag{4}$$



Kuva 3. Mel-asteikolla tasaisesti olevat suodattimet. Hz-asteikolla suodattimet ovat painottuneet matalammille taajuuksille.

MFC-kerrointen laskemiseksi käytettävä algoritmi on pelkistettynä seuraavanlainen:

1. Jaetaan signaali ikkunoituihin kehyksiin
2. Lasketaan kehyksen tehospektri
3. Luodaan suodatinpankki Mel-asteikolla tasaisesti olevilla, kolmiomaisilla kaistanpäästösuodattimilla (kuva 3). Yleensä suodattimia on 20–35.
4. Lasketaan kunkin suodattimen alle jäävä energia
5. Lasketaan logaritmi jokaisesta summavektorin termistä
6. Lasketaan summavektorille tyypin 2 kosinimuunnos

Algoritmin tuloksena saadaan yhtä monta MFC-kerrointa, kuin suodatinpankissa on suodattimia. Nollas termi tavallisesti jätetään huomiotta, koska sen erottelukyky on huono. Loput termit muodostavat MFC-kertoimet. Tavallisesti käytetään 10–15 ensimmäistä kerrointa, eli vain noin puolta kaikista kertoimista. Kaikkia kertoimia ei käytetä, koska ylemmät kertoimet eli korkeammat taajuudet sisältävät niukasti informaatiota. Saadut kertoimet voidaan ottaa osaksi laajempaa piirrevektoria tai olla koko piirrevektori.

2.2. Luokittelu

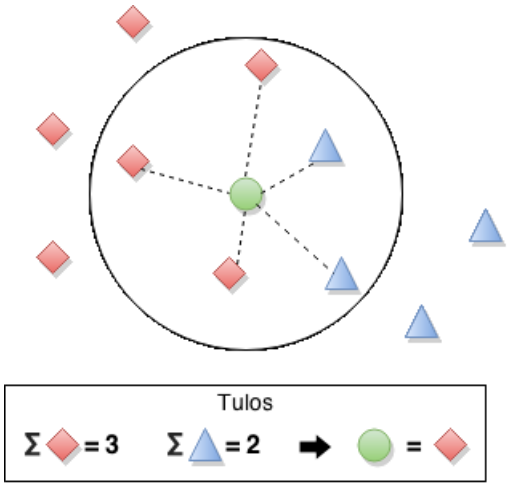
Luokittelun tehtävä on jakaa tuntemattomat näytteet oikeisiin luokkiin. Tämä tehdään yleensä hankkimalla iso määrä näytteitä määritetyistä luokista. Näistä näytteistä lasketaan tarkkaan valitut piirteet, jotka kuvastavat luokkien välistä eroa. Tuntemattomasta näytteestä lasketaan nämä samat piirteet ja tehdään päätös jo valmiiksi laskettujen piirteiden perusteella. Tunnettuja ja usein käytettyjä algoritmeja luokitteluun ovat:

- k:n lähimmän naapurin -menetelmä (k-Nearest Neighbour, k-NN)
- Naiivi Bayes (Naive Bayes, NB)
- Keinotekoinen neuroverkko (Artificial Neural Network, ANN)
- Tukivektorikone (Support Vector Machine, SVM)

Algoritmien saavuttamat tarkkuudet puhtailla näytteillä eivät eroa huomattavasti toisistaan, mutta SVM:llä on saavutettu marginaalisesti parhaita tuloksia äänien luokittelussa[12]. Tarkkuus riippuu myös olennaisesti mitattavista piirteistä.

k:n lähimmän naapurin -menetelmä on eräs yksinkertaisimmista luokittelijoista. Algoritmi on laajasti käytössä nykyäänkin varsinkin tilanteissa joissa näytteiden jakautumista ei tarkkaan tiedetä. Kuvasta 4 voidaan nähdä esimerkki pelkistetystä luokittelu tilanteesta. Toiminta perustuu karkeasti seuraaviin vaiheisiin:

- Tallenna opetusnäytteistä saadut piirrevektorit muistiin
- Laske opetusnäytteiden etäisyydet luokiteltavasta piirrevektorista
- Ota k lähintä näytettä ja palauta se luokka mikä esiintyy useimmin.



Kuva 4. k-NN algoritmin kuvaus

Parametri k kuvastaa kuinka monta lähintä näytettä valitaan luokittelua varten. Valinnalla on selvä vaikutus lajittelun tulokseen. Kasvattamalla k:n arvoa voidaan vähentää kohinan vaikutusta, mutta samalla luokkien välinen ero sumentuu. Ison k:n tuottamia ongelmia voidaan vähentää painottamalla naapurit etäisyyksien mukaan. K:n valintaan on olemassa myös laskennallisia menetelmiä, mutta optimaalinen valinnan tekeminen on yleensä laskennallisesti epäkäytännöllistä. Useita heuristisia menetelmiä on kehitetty k:n arviointiin.

Merkittävimmät heikkoudet[13]:

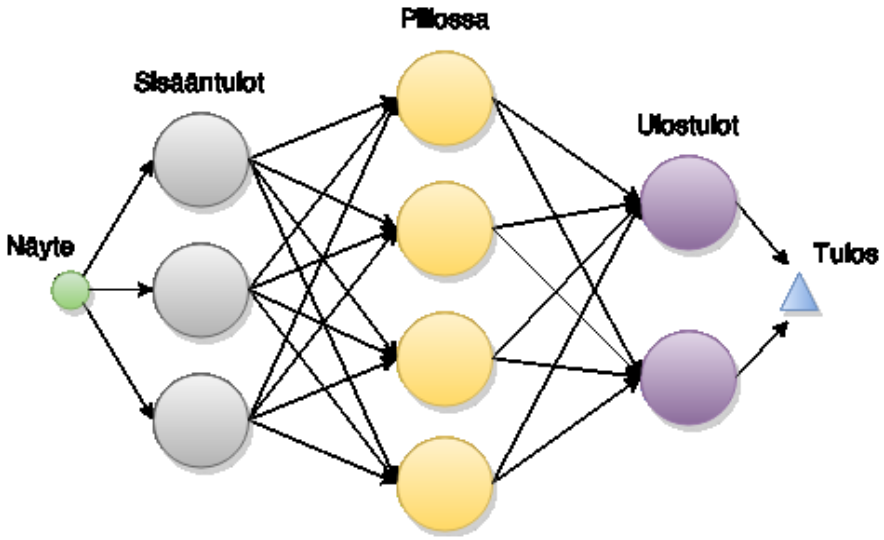
- Muistillisesti raskas suorittaa, koska harjoitustapaukset on kaikki pidettävä muistissa luokittelua varten
- Heikko kohinansieto.
- Ei luontevaa keinoa työskennellä puuttuvien argumenttien tai nimellisarvojen kanssa

Naiivi Bayes perustuu oletukseen siitä, että näytteen eri ominaisuuksilla ei ole vaikutusta toisiinsa. Toimii hyvin epäolennaisten ominaisuuksienkin kanssa. Luokkien koulutusnäytteiden vaihdellessa huomattavasti NB suosii sitä josta on enemmän näytteitä. Mainittakoon myös, että itsenäisyys oletus harvoin pitää täysin paikkaansa[14]. Bayes alkuperäinen teoreema kaavassa 5 ja naiiviolettamus kaavassa 6.

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} \tag{5}$$

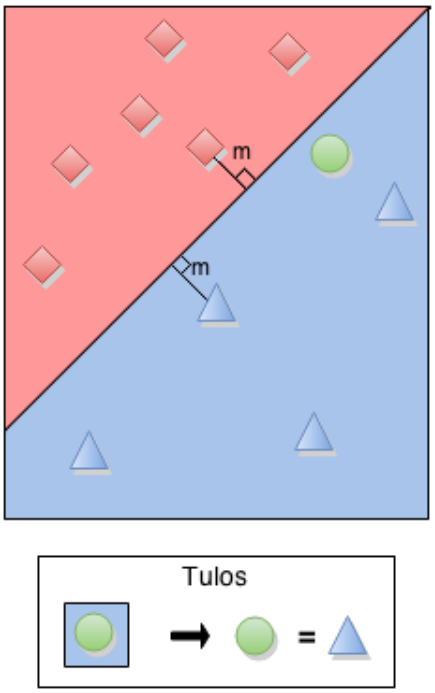
$$P(H|E) = \frac{P(e_1|H)\ldots P(e_n|H)P(H)}{P(E)} \tag{6}$$

Keinotekoinen neuroverkko simuloi ihmisen aivojen neuronien toimintaa luokittelun ongelmaan. Toimii hyvin isojen ominaisuus määrien kanssa ja kun ne ovat vahvasti toisiinsa yhteydessä. Isojen opetus joukkojen kanssa kohina aiheuttaa vääristymiä (overfitting)[15]. Kuvassa 5 nähtävissä yksinkertaistetun ANN luokittelijan toiminnan kuvaus.



Kuva 5. ANN luokittelija.

Tukivektorikone laskee optimaalisen hypertason kahden joukon välille tasossa maksimoiden niiden välisen marginaalin. Voi tuottaa hyvin tarkkoja luokittimia, mutta nämä ovat raskaita laskea[16]. SVM kestää hyvin kohinan vaikutusta. Se on myös luonteeltaan binääri luokittelija, eli luokittelu tehdään aina kahdesta vaihtoehdosta. Jos luokkia on useampia, ne on jaettava pareihin joka voi olla haastavaa[16]. Kuvassa 6 on SVM algoritmin kaksiulotteinen esimerkki.



Kuva 6. SVM luokittelija.

3. LUOKITTELIJAN TOTEUTUS

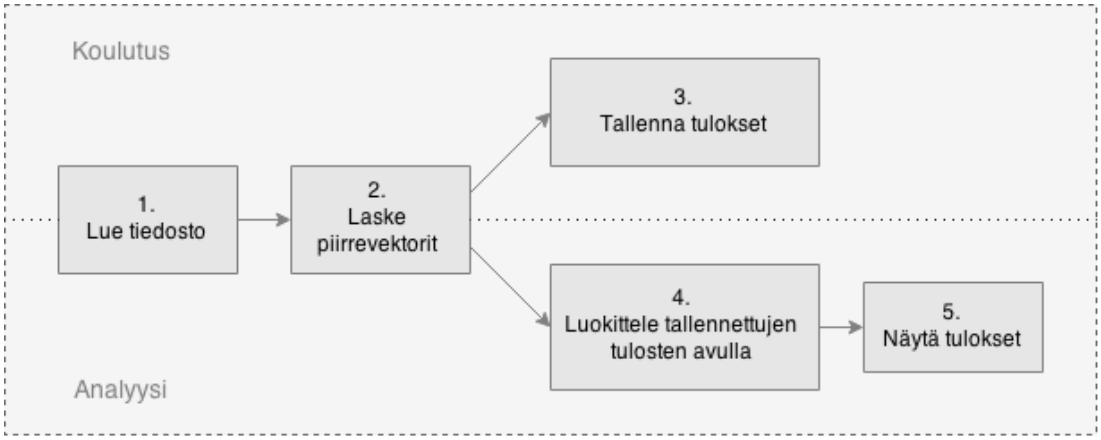
Työssä keskityttiin ihmisäänen tunnistamiseen sekalaisesta äänimaailmasta, sillä se on ensimmäinen vaihe puheentunnistusjärjestelmässä. Käytännössä toteutettiin luokittelija, joka kertoo kehys kerrallaan, onko näytteessä ihmisen ääntä vai ei. Järjestelmä luokittelee ääninäytteen kehysten enemmistön perusteella joko ihmisääneksi tai muuksi ääneksi.

Yleisesti kyseinen luokittelu tehdään aluksi, jonka jälkeen voidaan keskittyä syvällisempään puheen analysointiin ja sanojen tunnistukseen. Matalan tason tunnistus on tarpeen, sillä ensiluokittelun perusteella voidaan päätellä, tarvitseeko tarkempia puheentunnistusalgoritmeja ajaa.

3.1. Toiminnallisuus

Käytännössä järjestelmä toteutettiin Raspberry Pi -pienoistietokoneelle käyttäen Python 2.7.x ohjelmointikieltä sekä SciPy[17] ja Numpy-kirjastoja[18]. Lähdeäänitiedostojen tiedostomuoto oli yksikanavainen WAV 44,1 kHz:n näytteistystaajuudella. Kun näyte ladattiin luokittelijaan, se desimoitiin 22 050 Hz:n taajuuteen, ja ikkunoitiin 23,2 ms:n (512 näytettä) Hanning ikkunalla.

Ohjelman toiminnallisuus jakautuu kahteen osaan: opetusmoduuliin ja analyysimoduuliin. Moduulit on esitetty kuvassa 7. Molemmille toteutettiin yksinkertaisen graafinen käyttöliittymä, mikä tekisi ohjelman testauksesta huomattavasti selkeämpää ja nopeampaa verrattuna komentorivillä käytettävään ratkaisuun. Käyttöliittymä mahdollistaa myös tulosten graafisen esityksen.



Kuva 7. Moduulien toiminta

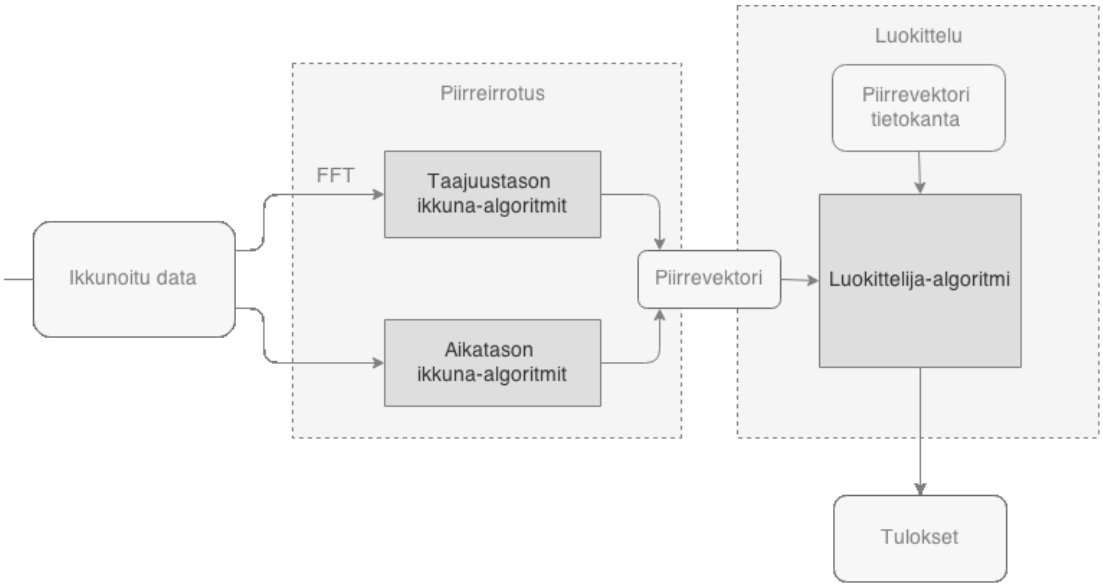
Opetusmoduulia käytetään nimensä mukaisesti aluksi luokittelijan opettamiseen. Ennen kuin luokittelija voi liittää ääniä kategorioihin, pitää sille luoda pohjatiedot. Opetusäänitiedostoista lasketaan äänitiedostoille kehyskohtaiset piirrevektorit. Saadut piirrevektorit yhdistetään lähdekansionsa perusteella oikeaan kategoriaan kuuluviksi. Opetusnäytteet täyttävät piirreavaruuden, johon testattavia ääninäytteitä verrataan.

Pääosan toiminnoista suorittaa analyysimoduuli, jolla voidaan analysoida äänitiedostojen sisältöä. Äänitiedostosta saatuja piirrevektoreita verrataan tietokannassa oleviin vektoreihin k-NN luokittelualgoritmilla. Toteutettu k-NN luokittelija etsii luokiteltavalle piirrevektorille viisi lähintä vektoria ja pisteyttää ne etäisyyden mukaan. Luokittelija vertailee jokaista näytteen kehystä ja palauttaa arvion

siitä, onko kehyksen sisältö ihmisen ääntä vai ei. Lopuksi luokittelija palauttaa ihmiseksi ja muuksi ääneksi arvioitujen kehysten lukumäärät. Tuloksia voidaan tarkastella graafisesti ja kehyskohtaisesti analyysin valmistuttua, mikä helpottaa ongelmakohtien löytämisessä.

Laskettaviksi äänen piirteiksi valittiin taulukossa 2 olevat kahdeksan piirreirrotusalgoritmia, joista jokainen ottaa syötteenä joko raakadatakehyksen tai kehyksestä lasketun FFT-datan (kuva 8). Piirteiden valintaperusteena olivat piirteen yleisyys alan julkaisuissa sekä niiden laskennallinen yksinkertaisuus, joka mahdollisti suhteellisen nopeasti piirreirrotuksen Raspberry Pi:lla. Kaikkien algoritmien toteutuksen pohjana oli piirteen matemaattinen määritelmä. Joissain piirreirrotusalgoritmeissa tehtiin pieniä yksinkertaistuksia tai muutoksia, joiden katsottiin parantavan tuloksia tai nopeuttavan niiden laskentaa. MFCC toteutettiin 25 suodattimella, joista kolmeatoista ensimmäistä käytettiin.

Luokittelussa eri piirteiden saamat arvot vaihtelivat suuruusluokaltaan huomattavan paljon, joten arvot normalisoitiin. Ilman normalisointia piirrevektorien välisiä euklidisia etäisyyksiä ei voida laskea suosimatta jotain piirteitä.



Kuva 8. Datan piirreirrotus ja luokittelu

4. TULOKSET

Testien lähtökohtana oli kysymys: mitkä piirteet ovat hyviä ihmisen äänen erotteluun? Ideaalinen erottelukyvyn resoluutio olisi sama kuin ihmisellä eli tavoitteena ihmisen kaltainen erottelukyky. Testattavat äänen piirteet valittiin taulukon 2 mukaan ja testiaineisto luokiteltiin käyttäen yhtä tai useampaa algoritmia samanaikaisesti.

Testiaineisto koostui 117 ääninäytteestä, joista jokainen oli äänitetty 44,1 kHz näytteistystaajuudella. Testiaineistossa oli kattavasti naisten ja miesten puhetta sekä suomeksi että muilla kielillä, lintujen ja eläinten ääniä, musiikkia, räjähdyksiä, koneita sekä useita muita eri kuuloisia äänilähteitä.

Testit suoritettiin valitsemalla kolme testisarjaa ääninäytteistä. Jokaiseen testisarjaan kuului 20 muutamien sekuntien mittaista ääninäytettä, joista 10 oli ihmisääntä ja 10 muita ääniä. Luokittelija opetettiin jäljelle jääneillä 97 näytteellä. Verrattain isolla opetustietokannalla pyrittiin minimoimaan luokittelualgoritmin vaikutusta, ja keskittymään piirteisiin.

Aluksi testattiin MFCC erikseen, jonka jälkeen muut piirteet testattiin yksin, pareittain sekä kolmen piirteen ryhmissä. Lopuksi suoritettiin testi käyttäen kaikkia piirteitä yhdessä MFCC:tä lukuun ottamatta. MFCC tuottaa 12 termiä piirrevektoriin, joten se erotettiin muista piirteistä. Alustavien testien perusteella MFCC:n tarkkuus oli luokittelussa niin hyvä, että muiden piirteiden vaikutus luokitteluun jäisi minimaaliseksi. Tulokset testeistä on kerätty taulukoihin 4, 5, 6 ja 7. Taulukoissa käytettävät lyhenteet on lueteltu alla taulukossa 3:

Taulukko 3. Piirteiden lyhenteet.

Lyhenne:	Piirre:
SC	Spectral centroid
SF	Spectral flux
SRO	Spectral roll off
SV	Spectral variance
BER	Band energy ratio
ZCR	Zero crossing rate
STE	Short term energy

4.1. Yksittäiset piirteet

Kuten taulukosta 4 näkee, kontrollipiirre MFCC onnistuu erottelemaan äänet parhaiten. Taulukosta voi myös huomata, että muilla piirteillä ei saavuteta käytettävää tasoa. Erityisen heikosti pärjäsivät SRO ja ZCR, jotka eivät onnistuneet erottamaan luokkia ollenkaan. Parhaiten MFCC:n jälkeen suoriutuivat SF ja STE, vaikka yksittäisten ikkunoiden luokittelussa olikin huomattavaa heilahtelua. Yleisesti voidaan myös nähdä oikein lajiteltujen ikkunoiden osuudesta, että MFCC:tä lukuun ottamatta näytteistä lasketuissa piirteissä esiintyy vahvaa limittymistä.

Taulukko 4. Testin tulokset (piirteet erikseen).

Piirre	Näytteitä oikein			Kehyksiä oikein
	Ihmisiä	Muita	Yhteensä	
MFCC	30/30	27/30	57/60	81,90 %
SF	27/30	17/30	44/60	54,07 %
STE	28/30	15/30	43/60	53,53 %
BER	26/30	15/30	41/60	48,30 %
SV	24/30	15/30	39/60	54,28 %
SC	23/30	12/30	35/60	48,93 %
ZCR	30/30	0/30	30/60	38,10 %
SRO	30/30	0/30	30/60	37,99 %

4.2. Piirreparit

Kun piirteet yhdistettiin pareiksi, saavutettiin kautta linjan hieman paremmat luokittelutulokset. Kehystasolla luokittelu parani keskimäärin 8,43 prosenttiyksikköä, ja näytetasolla paras piirrepari luokitteli oikein 50/60 näytettä. Tämä parannus johtui enimmäkseen parantuneesta kyvystä suodattaa pois muut kuin ihmisen äänet. Tuloksista huomataan myös, että ikkunakohtaisen luokittelutarkkuuden parantaminen ei välttämättä näy suoraan näytekohtaisen tarkkuuden lisääntymisenä.

Taulukko 5. Testin tulokset (kaksi piirrettä).

Piirteet	Näytteitä oikein			Kehyksiä oikein
	Ihmisiä	Muita	Yhteensä	
SRO, BER	30/30	20/30	50/60	60,84 %
SV, SRO	29/30	19/30	48/60	65,30 %
STE, SRO	29/30	18/30	47/60	62,86 %
ZCR, SRO	30/30	15/30	45/60	56,96 %
SRO, SF	29/30	15/30	44/60	60,53 %
SC, SRO	28/30	16/30	44/60	56,82 %
SC, STE	29/30	15/30	44/60	54,38 %
SC, SV	30/30	14/30	44/60	54,75 %
BER, SF	28/30	15/30	43/60	54,53 %
ZCR, STE	29/30	14/30	43/60	56,93 %
SV, STE	28/30	15/30	43/60	60,64 %
ZCR, BER	28/30	14/30	42/60	57,19 %
STE, SF	28/30	13/30	41/60	53,43 %
ZCR, SF	28/30	13/30	41/60	55,51 %
SC, SF	29/30	12/30	41/60	53,21 %
SC, ZCR	26/30	15/30	41/60	52,83 %
SV, BER	28/30	12/30	40/60	51,95 %
SV, ZCR	29/30	11/30	40/60	56,69 %
SC, BER	28/30	12/30	40/60	51,93 %
SV, SF	27/30	12/30	39/60	54,18 %
STE, BER	24/30	11/30	35/60	51,63 %

4.3. Kolmen piirteen ryhmät

Parhaidenkaan yhdistelmien kanssa ei saavutettu 50/60 tarkempaa lajittelua johon päästiin jo pelkästään SRO ja BER piirreparilla. Keskimäärin kehyskohtainen luokittelu parani taas hieman. Piirteiden tyypillä vaikuttaa olevan luokittelun tarkkuuteen suurempi vaikutus kuin piirteiden määrällä.

Taulukko 6. Testin tulokset (kolme piirrettä).

Piirteet	**Näytteitä oikein**			**Kehyksiä oikein**
	Ihmisiä	**Muita**	**Yhteensä**	
ZCR, SRO, BER	30/30	20/30	50/60	65,72 %
SRO, BER, SF	30/30	18/30	48/60	66,13 %
STE, SRO, SF	30/30	18/30	48/60	63,61 %
SV, SRO, SF	30/30	18/30	48/60	64,14 %
SV, STE, SRO	29/30	19/30	48/60	66,36 %
SV, SRO, BER	30/30	17/30	47/60	67,15 %
SC, SRO, BER	30/30	17/30	47/30	62,61 %
STE, SRO, BER	30/30	16/30	46/60	66,92 %
ZCR, STE, SRO	30/30	16/30	46/60	64,00 %
SV, ZCR, SRO	30/30	16/30	46/60	63,76 %
SC, ZCR, SRO	30/30	16/30	46/60	59,95 %
SV, ZCR, BER	30/30	15/30	45/60	61,51 %
SC, STE, SF	30/30	15/30	45/60	55,75 %
SC, STE, SRO	30/30	15/30	45/60	62,53 %
SC, SV, SF	30/30	15/30	45/60	55,70 %
SC, SV, SRO	30/30	15/30	45/60	62,45 %
ZCR, SRO, SF	29/30	15/30	44/60	61,86 %
ZCR, STE, BER	30/30	14/30	44/60	61,18 %
SV, STE, SF	26/30	18/30	44/60	55,17 %
SC, ZCR, STE	30/30	14/30	44/60	56,88 %
SC, SV, ZCR	30/30	14/30	44/30	57,26 %
STE, BER, SF	28/30	15/30	43/60	54,73 %
ZCR, BER, SF	29/30	14/30	43/60	59,25 %
SV, BER, SF	28/30	15/30	43/60	55,00 %
SC, ZCR, BER	28/30	15/30	43/60	56,40 %
ZCR, STE, SF	30/30	12/30	42/60	56,59 %
SV, STE, BER	26/30	16/30	42/60	53,09 %
SV, ZCR, SF	30/30	12/30	42/60	56,81 %
SV, ZCR, STE	28/30	14/30	42/60	57,30 %
SC, SRO, SF	29/30	13/30	42/60	59,79 %
SC, SV, STE	28/30	14/30	42/60	54,41 %
SC, ZCR, SF	29/30	12/30	41/60	54,93 %
SC, STE, BER	29/30	11/30	40/60	57,67 %
SC, SV, BER	29/30	11/30	40/60	57,96 %
SC, BER, SF	29/30	10/30	39/60	54,90 %

Lopuksi ajettiin testisarjat kaikilla piirteillä lukuun ottamatta MFCC:tä. Kuten taulukosta 7 nähdään, epäolennaisten piirteiden lisääminen ei välttämättä paranna luokittelun tarkkuutta.

Taulukko 7. Testin tulokset (7 piirrettä).

Piirteet	**Näytteitä oikein**			**Kehyksiä oikein**
	Ihmisiä	**Muita**	**Yhteensä**	
SC, SV, ZCR, STE, SRO, BER, SF	30/30	19/30	49/60	66,10 %

5. POHDINTA

MFCC saavutti tarkimman luokitteluasteen. Eräissä lähteissä mainitaan MFCC:n erottelukyvyn heikkenevän tietynlaisilla äänillä[19], mutta käytetyllä testausaineistolla se suoriutui yllättävän hyvin. Oikein luokitelluista ihmisnäytteissä lähes jokainen ikkuna oli oikein. Oli myös yksittäisiä tapauksia, joissa ikkunakohtainen tarkkuus vaihteli enemmän, mutta lopputulos pysyi useimmiten oikeana.

Vaikka SRO sai heikkoja tuloksia yksinään, se oli osana kaikkia parhaiten suoriutuneita kolmikoita. Tämä tukee muissakin lähteissä havaittua piirteiden ominaisuutta: yksinään epäolennaiselta vaikuttava piirre voi parantaa luokittelukykyä osana laajempaa piirrevektoria[20]. Yleensä ottaen kolmen piirteen piirrevektorit tunnistivat ihmisen ääntä sisältäneet näytteet luotettavasti.

Testeissä käytettiin enintään kolmesta piirteestä rakentuneita piirrevektoreita, jotta laskenta-aika ja datan määrä ei kasvaisi liian suureksi. Neljän, viiden tai kuuden piirteen yhdistelmillä olisi mahdollisesti voitu saada tuloksia, jotka ovat lähempänä MFCC:n suorituskykyä. Taulukon 7 tulokset eivät kuitenkaan tue tätä spekulaatiota. Yhdistelmistä saatujen tulosten perusteella huomataan, että pelkkä piirteiden määrä piirrevektorissa ei aina johda parempaan luokittelutulokseen.

Hieman yllättäen yleisin virhetyyppi piirteestä riippumatta oli ihmisen puheeksi tunnistettu muu ääni (väärä positiivinen). Tällainen virhetyyppi on hyväksyttävämpää kuin ihmisäänen tunnistamatta jättäminen (väärä negatiivinen) järjestelmässä, jonka tehtävä on tunnistaa, milloin ihmisen ääntä kuuluu.

Hyvälaatuiset ja monipuoliset opetusnäytteet osoittautuivat ensiarvoisen tärkeiksi toteutetun kaltaisessa luokittelijassa. Toteutusvaiheessa havaittiin, että sisällyttämällä opetusnäytteisiin näytteitä, joissa on kattavasti eri foneemeja, saavutetaan parempia tuloksia kuin sattumanvaraisesti valituilla puhenäytteillä.

k-NN luokittelijan toteutus vaikuttaa myös marginaalisesti tulokseen. Heuristisesti valituilla k:n arvoilla tai eri luokittelualgoritmeilla tulokset olisivat voineet olla hieman erilaiset. K:n arvoa rajoitti toteutusalustan rajallinen suorituskyky.

Työssä keskityttiin vain “puhtaiden” näytteiden luokitteluun, eli kaikki systeemiin tulevat näytteet sisältävät vain yhtä äänilähdettä jolla on korkea signaali-kohinasuhde. Täten tuloksemme eivät kuvaa tunnistuksen laatua meluisissa ympäristöissä.

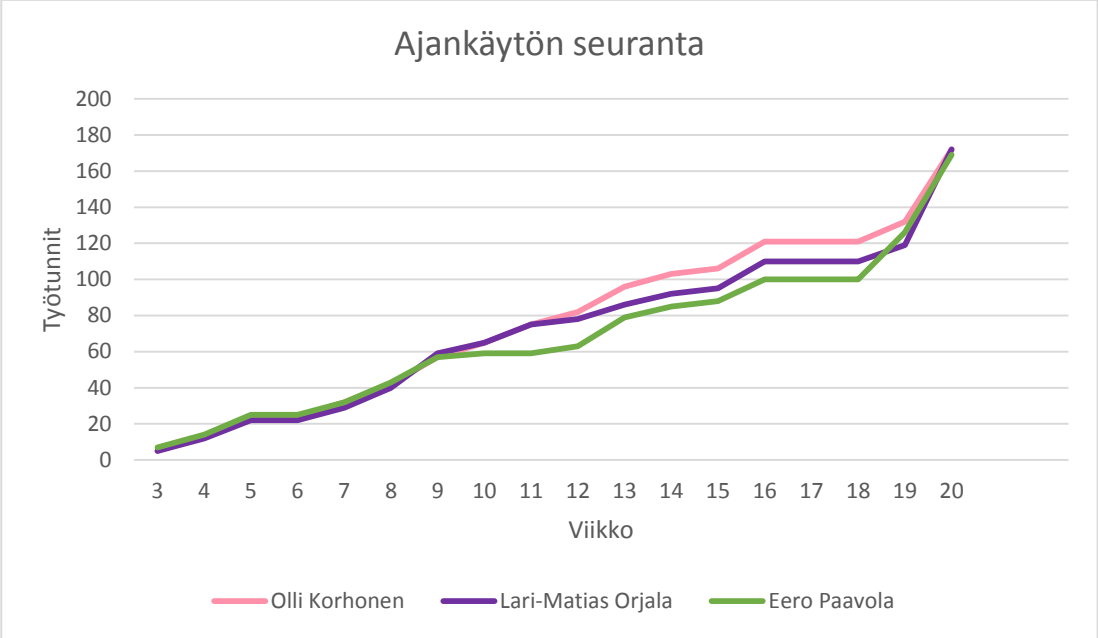
5.1. Kehitysmahdollisuudet

Puheen tunnistusta voitaisiin jatkokehittää uusilla piirteillä ja mahdollisesti yhdistämällä niitä MFCC:n kanssa. Mahdollisimman kattavan puheentunnistuksen toteuttamiseksi, tulee luokittelijan pystyä optimoimaan luokitteluaan. Itseoppiva järjestelmä pystyisi vajaalla tai jopa olemattomalla tietokannalla analysoimaan äänisignaaleja, kasvattamaan näytetietokantaa ja siten parantamaan puheen tunnistusta.

Eri luokittelualgoritmit voivat parantaa puheen tunnistamista tai luokittelun nopeutta. Tarkkuuden kehittämisen lisäksi tulisi pyrkiä järjestelmän toimintanopeuden parantamiseen, jolloin voitaisiin prosessoida näytteitä reaaliajassa.

Virheiden määrää voitaisiin vähentää myös parantamalla signaalihäiriöiden sietoa. Häiriön muotoja ovat muun muassa taustamelu, huono lähdemateriaalin laatu sekä ääntä vääristävät ilmiöt kuten kaiku.

6. PROJEKTIN KUVAUS



Projektin työmäärä jaettiin tasaisesti. Jokainen osallistui aluksi taustatiedon ja lähteiden keruuseen. Olli Korhonen toteutti k-NN luokittelijan ja testausympäristön. Lari-Matias Orjala toteutti ulkoasun, äänitiedostojen käsittelyn ja ikkunoinnin. Eero Paavola toteutti testiaineiston keräämisen ja oikoluvun. Lisäksi jokainen toteutti piirreirrotusalgoritmeja: Olli(SV, SF, BER), Lari-Matias(SRO, SC) ja Eero(ZCR, STE, MFCC).

7. YHTEENVETO

Työssä toteutettiin binäärinen ihmisen äänentunnistusjärjestelmä, jonka toimintaa testattiin useilla erityyppisillä äänillä. Tavoitteena oli verrata yleisesti käytettyjä äänen piirteitä ja tutkia, mitkä niistä ovat käyttökelpoisia ihmisäänen tunnistuksessa. Hyvin valituilla piirteillä mahdollistetaan äänen luokittelun tarkkuus ilman ylimääräistä laskentaa ja siten koko järjestelmän tehokkuus.

MFCC-algoritmi ja useita muita piirreirrotusalgoritmeja toteutettiin piirteiden laskemista varten. Suoritettiin luokittelutestejä painotetun k-NN luokittelijan avulla. Testiaineistona käytettiin 117 ääninäytettä. Valittiin 20 näytettä luokiteltavaksi ja ylitsejääneiden näytteillä opetettiin luokittelija.

Testeistä kävi ilmi, että MFCC on erityisesti ihmisen äänen erottamiseen toimiva ja tehokas äänenpiirre. MFCC hyötynä on sen ihmisen kuulolle tärkeiden taajuusalueiden energiasisällön esittäminen ja siksi se onkin yksi luotettavimmista valinnoista ihmisen äänen luokitteluun. Lisäksi huomattiin, että luokitteluun käytettyjen piirteiden määrä ei ole suoraan verrannollinen luokittelutarkkuuteen. Piirrevektorin pituutta olennaisempaa on käyttötarkoitukseen sopivat ja toisiaan täydentävät piirteet.

8. LÄHTEET

[1] Google Inc. (2015) Google Now. URI: http://www.google.com/landing/now/. Cited 1.3.2015.

[2] Apple Inc. (2015) Apple iOS8 Siri. URI: https://www.apple.com/ios/siri/. Cited 1.3.2015.

[3] Microsoft (2015) Microsoft Phone Cortana. URI: http://www.windowsphone.com/fi-fi/how-to/wp8/cortana/meet-cortana. Cited 1.3.2015.

[4] Fagerlund S (2014) Studies on Bird Vocalization Detection and Classification of Species. Aalto University.

[5] Rosen S & Howell P (2011) Signals and Systems for Speech and Hearing. In: Anonymous , Emerald: 163.

[6] Gelfand SA (2011) Essentials of Audiology. In: Anonymous , Thieme: 87.

[7] Zhu Q & Alwan A (2000) On the use of variable frame rate analysis in speech recognition. 2000 IEEE Interntional Conference on Acoustics, Speech, and Signal Processing. Piscataway, NJ, United States, IEEE 3: 1783-1786.

[8] Mitrovic D, Zeppelzauer M & Eidenberger H (2007) Analysis of the data quality of audio descriptions of environmental sounds. J Digit Inf Manage 5(2): 48-54.

[9] Mitrović D, Zeppelzauer M & Breiteneder C (2010) Chapter 3 - Features for Content-Based Audio Retrieval. Advances in Computers 78(0): 71-150.

[10] Sahidullah M & Saha G (2012) Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition. Speech Commun 54(4): 543-565.

[11] Memon S, Lech M & He L (2009) Using information theoretic vector quantization for inverted MFCC based speaker verification. 2009 2nd International Conference on Computer, Control and Communication, IC4 2009.

[12] Chen L, Gündüz S & Özsu MT (2006) Mixed type audio classification with support vector machine. 2006 IEEE International Conference on Multimedia and Expo, ICME 2006. 2006: 781-784.

[13] Aha DW, Kibler D & Albert MK (1991) Instance-based learning algorithms. Mach Learn 6(1): 37-66.

[14] Rennie JDM, Shih L, Teevan J & Karger D (2003) Tackling the Poor Assumptions of Naive Bayes Text Classifiers. Proceedings, Twentieth International Conference on Machine Learning. 2: 616-623.

[15] Tu JV (1996) Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. J Clin Epidemiol 49(11): 1225-1231.

[16] Auria L & Moro RA (2008) Support Vector Machines (SVM) as a Technique for Solvency Analysis. , Deutsches Institut f{\u}r Wirtschaftsforschung.

[17] Scipy developers (2015) Scipy. URI: http://scipy.org/. Cited 1.3.2015.

[18] Numpy developers (2015) Numpy. URI: http://www.numpy.org/. Cited 1.3.2015.

[19] Chu S, Narayanan S & Kuo C-J (2009) Environmental Sound Recognition With Time–Frequency Audio Features. Audio, Speech, and Language Processing, IEEE Transactions on 17(6): 1142-1158.

[20] Iguyon I & Elisseeff A (2003) An introduction to variable and feature selection. Journal of Machine Learning Research 3: 1157-1182.