TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Henry Laur 164371IAPB

MUUSIKAŽANRITE TUVASTAMINE KASUTADES NÄRVIVÕRKU

Bakalaureusetöö

Juhendaja: Ago Luberg
Magistrikraad

Tallinn 2019

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Henry Laur

21.05.2019

Annotatsioon

Töö eesmärgiks on luua rakendus, mis suudab tuvastada helifaili žanri. Klassifitseerimiseks on viis erinevat žanrit. Helifaili žanri tuvastamiseks on kasutusel konvolutsiooniline närvivõrk, mida treenitakse FMA andmestiku helifailidel.

Töös analüüsitakse erinevaid meetodeid närvivõrgu täpsuse suurendamiseks. Uuritakse, kas helifailidest saadud andmete tükeldamine aitab sellele kaasa. Lisaks uuritakse erinevaid vastuse kombineerimismeetodeid.

Rakendusele tehakse kasutajaliides, mis võimaldab näha konvolutsioonilise närvivõrgu vastust sisestatud helifailile.

Täpsuseks saadi mitte tükeldatud andmetega treenitud närvivõrgul 64% ja saagis 62%. Tükeldatud andmetega treenitud närvivõrgul, mis kasutab liitmismeetodit on täpsus 73% ja saagis 71%. Tükeldatud andmetega treenitud närvivõrgul, mis kasutab hääletusmeetodit on täpsus 71% ja saagis 69%.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 39 leheküljel, 6 peatükki, 11 joonist, 7 tabelit.

Abstract

Music genre classification using neural networks

One of the main ways to classify a song is to use genres. There is no exact definition of any single genre and classifying them is subjective.

Genre classification gives a way to understand the general make up of a song without listening to it. It also helps people find similar songs that they might like. The main way to achieve this is by automating the genre classification process.

Genre classification will be done with the help of neural networks. Neural networks will be able to train on songs which genres are already known. This enables them to find features which help in classifying new songs. In this thesis five different genres will be used to classify songs. The main reason for this is more genres increase the total training time and make the network less accurate. Fewer classes however do not offer enough coverage of different music genres.

The goal of this thesis is to create a desktop application which using convolutional neural netowks can identify a song's genre. Furthermore testing if splitting the song data into smaller frames will increase the neural networks accuracy. This would enable better accuracy without additional training data. Additionally different combination methods will also be explored.

A user interface will also be created for the application. It will show the neural networks answer to the song's genre. It will also show the likelihood of each genre for the song.

Using the FMA dataset the presicion of the neural network trained on the unsplit data was 64% and recall 62%. The neural network which was trained on split data using the addition combination method achieved a precision of 73% and recall of 71%. Using the voting combination method the neural network achieved a precision of 71% and recall of 69%.

The thesis is in Estonian and contains 39 pages of text, 6 chapters, 11 figures, 7 tables.

Lühendite ja mõistete sõnastik

Aktiveerimisfunktsioon	*Activation function*
Andmestik	*Dataset*
Dropout-meetod	*Dropout*
Epohh	*Epoch*
Juhendatud õpe	*Supervised learning*
Kadu	*Loss*
Konvolutsiooniline võrk	*Convolutional network*
L2 regulariseerimine	*L2 regularization*
MFCC	*Mel-frequency cepstral coefficients*
Peidetud kiht	*Hidden layer*
ReLU	mittenegatiivne lineaarfunktsioon
Saagis	*Recall*
Samm	*Stride*
Sisendkiht	*Input layer*
Tagasilevi	*Back-propagation*
Tunnused	*Features*
Tunnuskaart	*Feature map*
Täpsus	*Precision*
Väljundkiht	*Output layer*
Ääris	*Padding*
Õpisamm	*Learning rate*
Ülesobitus	*Overfitting*

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Üks peamisi viise klassifitseerida muusikalugusid on nende žanrite järgi. Žanrite täpsed piirid on hägused ning nende määramine on subjektiivne.

Žanrite tuvastamine võimaldab saada üldise ülevaate muusikaloost, ilma seda kuulamata ning aitab inimestel leida sarnaseid lugusid, mis neile võiksid meeldida. Ainuke viis selleks oleks automatiseerimine, mis vähendaks inimese aega ning tööd, mis kuluks žanrite määramiseks.

Žanrite automaatseks klassifitseerimiseks on antud töös kasutatud närvivõrke. Närvivõrgud suudavad õppida eelnevalt teadaolevate lugude ja nende žanrite pealt piisavalt palju erinevaid tunnuseid, et leida uue loo žanr. Töös on klassifitseerimiseks viis erinevat žanrit, sest rohkemate žanrite klassifitseerimisega kasvab treenimiseaeg ning väheneb täpsus [1]. Vähemate žanritega ei oleks võimalik katta piisavalt palju erinevaid žanre.

Antud töö eesmärgiks oleks luua rakendus, mis kasutades konvolutsioonilist närvivõrku suudab tuvastada antud helifaili žanri viiest eeldefineeritud žanrist. Selleks, et tõsta närvivõrgu täpsust, katsetatakse, kas helifailist saadud andmete tükeldamine tõstab närvivõrgu täpsust. See võimaldaks ilma lisaandmete kasutamiseta suurendada närvivõrgu efektiivsust.

Töö koosneb neljast erinevast osast, esimene on töös kasutatavate muusikažanrite kirjeldus. Teises osas kirjeldatakse klassifitseerimise viisid, närvivõrgu olemus ning antakse ülevaade konvolutsioonilisest närvivõrgust. Kolmandas osas kirjeldatakse täpsemalt rakendust, kasutatavat andmestikku ja helifailidest saadavaid andmeid. Lisaks antakse kolmandas osas ülevaade andmete tükeldamisest, töös kasutatava konvolutsioonilise närvivõrgu arhitektuurist ning tehnoloogiatest. Viimases peatükis analüüsitakse saadud tulemusi, võrreldakse närvivõrgu täpsuseid tükeldatud ja mitte tükeldatud andmete peal. Lisaks analüüsitakse ka vastuse kombineerimismeetodeid. Lõpuks kirjeldatakse valminud kasutajaliidest ning selle sidumist närvivõrguga.

2 Muusikažanrid

Muusikažanr liigitab erinevad muusikalood nende traditsioonide ja tavade järgi [2]. Kuigi nende määramine on subjektiivne ja osad žanrid võivad kattuda, on enamus lugusid ikkagi piisaval määral erinevad, et neid oleks võimalik klassifitseerida.

Käesolevas töös analüüsitakse viit erinevat žanrit, mida närvivõrk hakkab tuvastama: Valitud on viis žanrit seetõttu, et saavutatud täpsus oleks võimalikult hea. Kuid samas oleks võimalik klassifitseerida piisavalt palju erinevaid žanre, et tuvastada enamus muusikalood.

- Pop
- Hiphop ja räpp
- Instrumentaalmuusika
- Rokk
- Folkmuusika

Žanrite valik on mõjutatud andmestiku valikust.

2.1 Popmuusika

Popmuusika tekkis 1950ndatel ning arenes välja *rock and roll'ist* [3]. Laulud on väikese pikkusega, tavaliselt 3 minutit, lihtsa meloodiaga ning madala struktuurse keerukusega [4].

Popmuusika on üldiselt väga lai žanr ja see muudab klassifitseerimise raskemaks. Popmuusika on kõige lähemalt seotud rokkmuusikaga ning seetõttu võib nende žanrite eristamine olla raskem.

2.2 Rokkmuusika

Rokkmuusika arenes välja 1960ndatel. Rokkmuusikas kasutatakse võimendatud instrumente, eriti elektrikitarri ning elektribassi ja trummi. Rokkmuusikat iseloomustab tugev bass ning kiire rütm [5].

Rokkmuusika on tavaliselt agressiivsem kui popmuusika.[6]. Rokkmuusika on 60ndatest kuni tänaseni arenenud palju laiemaks žanriks [7]. Seetõttu võib närvivõrgul olla raskusi selle tuvastamisel.

2.3 Hiphop ja Räpp

Hiphop arenes välja 1970ndatel Ameerikas, mustanahaliste inimeste kultuurist . Hiphop muusikas on tavaliselt kasutusel ka räpp. Lugu koosneb taustamuusikast, kus kasutatakse digitaalset sämplimist, mida kutsutakse Hiphopiks ning laulja, kes laulab rütmilises kõnes [8].

Kuna hiphop arenes välja ilma teiste töös kasutatavate žanrite mõjuta, siis ei ole ka suurt ülekattumist teiste muusikastiilidega, mis teeb žanri klassifitseerimise lihtsamaks. Arvatavasti on hiphopi žanri klassifitseerimine täpsem, kui rokkmuusika ja popmuusika.

2.4 Instrumentaalmuusika

Instrumentaalmuusika on muusikalugu, kus ei esine vokaali [9]. Kasutatava andmestiku helifailidel on ainult 1 põhižanr. See tähendab, et helifailides, kus puudub vokaal on instrumentaalmuusika [10].

Instrumentaalmuusika võib sarnaneda igale teisele neljale žanrile, kuid vokaali puudumine peaks olema piisavalt suureks erinevuseks, et seda saaks teistest eristada ning klassifitseerida õigesti.

2.5 Folkmuusika

Selles töös kasutatav folkmuusika ei ole ainult traditsiooniline rahvusmuusika vaid selles žanris on ka folk-pop, folk-rokk ja *indie* folk muusikat[11]. Kuid alamžanrite ühine joon, on nende inspiratsioon tavalisest rahvusmuusikast.

Folk-pop ja folk-rokk on sarnased rokile [12],[13]. Sellest tulenevalt on võimalik, et žanri ära tundmine on raskem.

3 Närvivõrk

Käesolevas peatükis analüüsitakse erinevaid klassifitseerimise võimalusi ning antakse ülevaade närvivõrkudest.

3.1 Klassifitseerimise võimalused

Muusikažanrite klassifitseerimiseks on väga palju erinevaid võimalusi ning on uuritud erinevate masinõppe algoritmide efektiivsust selle ülesande lahendamiseks.

Uurimistöös „Music Genre Classification“ uuriti k-lähima naabri, k-keskmiste klasterdamise, tugivektormasinate ning närvivõrkude täpsust helifailide žanrite klassifitseerimisel. Nad kasutasid töös GTZAN andmestikku, helifailidest said MFCC abil klassifitseerida žanre. Leiti, et k-lähima naabri ja k-keskmiste klasterdamise täpsus oli väga sarnane, umbes 80%, mõlemal algoritmil oli raskusi jazz žanri tundmisega, kuid teiste žanrite täpsus oli kõrgem. Tugivektormasina täpsus oli 87%, mis on parem kui k-lähima naabri ja k-keskmiste klasterdamise täpsus, kuid jällegi oli tugivektormasinal raskusi jazz žanri tundmisega, mis viis keskmise täpsuse alla. Närvivõrgu täpsus oli 96% ning närvivõrk suutis 100% tuvastada jazz ja pop žanri, mis viis keskmise täpsuse võrreldes teiste meetoditega palju suuremaks [14].

Käesolevas töös on muusikalugude klassifitseerimiseks kasutusel närvivõrk just seetõttu, et närvivõrk on väga efektiivne lahendus sellele ülesandele.

3.2 Tehisnärvivõrk

Tehisnärvivõrkudega hakati tegelema aastal 1943, kui Warren McCulloch ja Walter Pitts avaldasid artikli, kuidas neuronid töötavad. Närvivõrgud suudavad isegi rasketest ülesannetest leida teatud mustrid ja trende, mida inimene ning teised arvutamise metoodikad ei suuda leida [15].

Tavalised algoritmilised lahendused jälgivad kindlat reeglistikku, et lahendada mingi probleem. Kui täpsed sammud probleemi lahendamiseks ei ole teada, siis ei ole võimalik seda lahendada [15].

Tehisnärvivõrk on disainitud inimese aju järgi, seal on teatud arv neurone, mis on üksteistega ühendatud. Kõikidel närvivõrkudel on sarnane topoloogia, mis koosneb kolmest erinevast kihist. Esiteks sisendkiht, mis on esimene kiht, kuhu andmed lähevad ning mis töötleb otseselt andmeid. Sisendkihte on ainult üks. Teine kiht on peidetud kiht, see kiht saab enda sisendid teistelt kihtidelt, sisendkihilt või teistelt peidetud kihtidelt. Peidetud kihte võib olla piiramatu arv. Viimane kiht on väljundkiht, mis saab enda andmed peidetud kihilt või selle puudumisel sisendkihilt. Väljundkiht ei anna enam infot edasi teistele neuronitele, vaid väljundkihi neuronite väärtus on kogu närvivõrgu lõplik vastus sisenditele [15].

Juhendatud õppes on nii sisendid kui ka väljundid treeningandmetel juba teada. Närvivõrk töötleb sisendid ning võrdleb saadud väljundeid soovitud väljunditega. Seejärel viga liigub mõõda närvivõrku tagasi ning neuronite kaale muudetakse, selle protsessi nimi on tagasilevi [15].

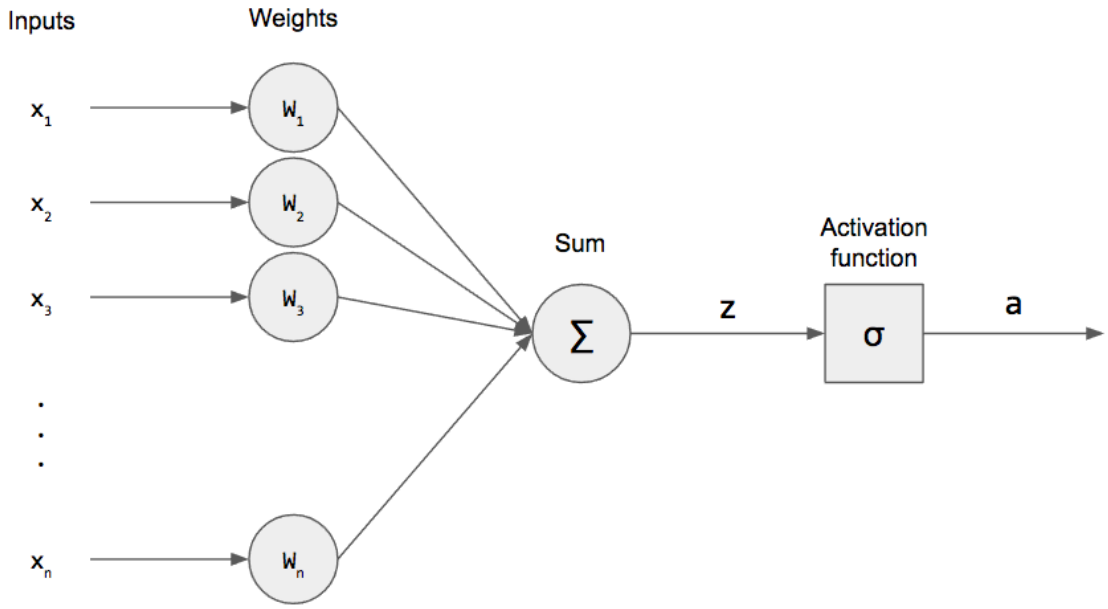
Õppimine ja üldistuste tegemine on närvivõrgu kõige olulisem ülesanne. Õppimine tähendab seda, et närvivõrk suudab andmetest leida mingisuguse loogika ning selle abil ennustada, milline on väljund. Närvivõrkudel on väga lihtne ka ülesobitada, mis tähendab, et närvivõrku täpsus treeningandmetel on väga suur, kuid uutel andmetel on madal. Ülesobitamise vastu on palju erinevaid võtteid, näiteks suurendada treeningandmete kogust, muuta närvivõrgu kihtide arvu või elementide arvu kihil. Lisaks on väga oluline õige närvivõrgu tüübi valik [16].

Närvivõrgu täpsuse tõstmiseks peab närvivõrk õppima. Selleks on vaja optimeerida kadu närvivõrgul, ehk viia närvivõrgu viga võimalikult väikeseks. Kaofunktsioon näitab närvivõrgu veamõõtu ja selle numbri vähenemisel muutub närvivõrk täpsemaks [17].

Kao optimeerimisfunktsioonid proovivad leida miinimumi, et viia kao väärtus võimalikult väikeseks. Käesolevas töös on optimeerimisfunktsioonina kasutatud Adamit. Adam on väga sarnane RMSProp ja AdaGrad optimeerimisfunktsioonidele ning võtab mõlemast funktsioonist elemente, et olla veelgi efektiivsem. Eksperimendis, kus kasutati pildi tuvastamiseks konvolutsioonilist närvivõrku, on näha, et Adam oli kõige efektiivsem optimeerimisfunktsioon [18].

Närvivõrgu baaselement on neuron. See töötab sarnaselt bioloogilisele neuronile, mis saab teistelt neuronitelt teatud kaaluga elektrisignaali. Kui neuron on saanud piisavalt

suure signaali, siis annab signaali edasi teistele ühenduses olevatele neuronitele. Tehisnärvivõrgu neuron töötab väga sarnaselt (Joonis 1), see saab $\{x_1, x_2, .., x_n\}$ teistelt neuronitelt signaali, mille kaal on $\{w_1, w_2, .., w_n\}$, mis vastab igale sisendneuronile. Kõik signaalide kaalud liidetakse kokku ning antakse edasi aktiveerimisfunktsiooni. See funktsioon näitab, milliste väärtuste korral neuron annab signaali edasi ning määrab ära ka signaali suuruse [19]. Aktiveerimisfunktsioone on erinevaid ning nende valik mõjutab närvivõrgu efektiivsust.



Joonis 1. Ülevaade ühe neuroni ehitusest

Sigmoid aktiveerimisfunktsioon on üks kõige levinumaid aktiveerimisfunktsioone. See on defineeritud järgmise valemina [19].

$$g(x) = \frac{1}{1+e^{-x}}$$

Kus x kuulub piirkonda $(-\infty, \infty)$ ja g kuulub piirkonda (0,1). Tagasilevis on vaja kasutada aktiveerimisfunktsiooni tuletist. Sigmoidi tuletise arvutamine on lihtne, seetõttu kasutatakse tihti sigmoid funktsiooni närvivõrkudes, kus on vähe kihte [19].

Uuringus leiti, et ajus töötavad üks kuni neli protsenti neuroneid korraga. Kuid tehisnärvivõrgus, sigmoid funktsiooni korral töötasid umbes pooled. Sellepärast töötati välja ReLU aktiveerimisfunktsioon, mis peaks korraga töötavate neuronite arvu vähendama ning suurendama efektiivsust. Praeguse seisuga on ReLU üks kõige

populaarsemaid aktiveerimisfunktsioone. ReLU funktsioon on defineeritud järgmiselt [19]:

$$g(x) = \begin{cases} x, kui\ x \geq 0 \\ 0, kui\ x < 0 \end{cases}$$

ReLU funktsioon töötab palju efektiivsemalt kui sigmoid, sest arvutused on lihtsamad, lisaks korraga ei tööta nii palju neurone. ReLU sobib palju paremini juhendatud õppe ülesannetele [19].

Närvivõrgu tüüpe on väga erinevaid ning need on tavaliselt disainitud mingi täpsema probleemi lahendamiseks. Õige närvivõrgu tüübi valik mõjutab väga palju lõplikku täpsust treenitud mudelil. Näiteks Long-Short Term Memory närvivõrgud sobivad andmetele, mis on järjestatud [20], ja konvolutsioonilised närvivõrgud piltidele [21].

3.3 Konvolutsiooniline närvivõrk

Konvolutsiooniline närvivõrk on üks kõige arenenuimaid tehisnärvivõrgu liike, see on peamiselt kasutatud arvutinägemise ülesannetes. See on inspiratsiooni saanud loomade nägemise viisist. Konvolutsiooniline närvivõrk on võimeline õppima madala- ja kõrgetasemelisi mustreid [21].

Tavalises konvolutsioonilises närvivõrgus on kolm erinevat kihti: konvolutsiooniline kiht, ahenduskiht ning tavaline, täielikult ühendatud närvivõrgu kiht. Esimesed kaks kihti proovivad andmetelt leida tunnuseid ning viimane täielikult ühendatud kiht kaardistab saadud väärtused väljundkihti [21].

Pildil piksli väärtused on salvestatud kahemõõtmelisse (2D) massiivi. Konvolutsioonilises kihis käib *kernel* üle iga piksli ning proovib leida sealt tunnuseid. *Kernel'i* ning sisend 2D massiiviga sooritatakse elementide korrutus. Summa liidetakse kokku ning pannakse tunnuskaarti. Tavaliselt on ühe *kerneli* suurus 3x3, 5x5 või 7x7, mis on tavaliselt väiksem, kui sisse saadud massiiv. Seetõttu liigub *kernel* sammu võrra edasi ning teeb seda tehet uuesti, uute pikslitega. Konvolutsiooni väljund antakse edasi aktiveerimisfunktsiooni, mis on tavaliselt ReLU funktsioon [21].

Ahenduskiht võimaldab *downsample'*ida, mis vähendab saadud tunnuskaardi dimensioone. See vähendab õpitavate parameetrite ja arvutuste arvu, mis teeb närvivõrgu

efektiivsemaks. Ahenduskihis ei õpi närvivõrk midagi uut. Muudetavad parameetrid, nagu kihi suurus ja samm määratakse ära enne treenimist. Kõige tavalisem ahenduskiht on *max*-ahendus, kus tunnuskaart jagatakse vastavalt ahenduskihi suurusele ära ning igast osast võetakse selle maksimaalne väärtus ning pannakse uude tunnuskaarti [21].

Pärast konvolutsioonilisi- ja ahenduskihte on täielikult ühendatud kiht. Kõigepealt tehakse tunnuskaart ühemõõtmeliseks massiviks ning seejärel antakse see sisend täielikult ühendatud kihti. Täielikult ühendatud kiht omakorda on ühendatud väljundkihiga, mis näiteks klassifikatsiooni ülesannete korral näitab, millisesse klassi saadud sisend kuulub. Väljundkihi aktiveerimisfunktsioon on klassifitseerimise korral *softmax*, mis liidab saadud tulemused kokku ja normaliseerib [21].

4 Metoodika

Käesolevas peatükis tuuakse välja rakenduse kirjeldus, treenimiseks kasutatud andmestiku valik, helifailidest saadud andmete tüüp ja selle kasutamine. Lisaks kirjeldatakse andmete tükeldamist ning töös kasutatud konvolutsioonilist närvivõrku.

4.1 Rakendus

Töö käigus on disainitud rakendus, mis kasutades konvolutsioonilist närvivõrku, oskab määrata helifaili õige žanri, viie erineva žanri vahel. Rakenduse põhitöö koosneb neljast erinevast etapist.

Esimeses etapis on rakenduse UI kaudu võimalik leida helifaili asukoht. Helifail peab asuma kasutatava arvuti kõvakettal. Rakendus toetab .wav helifaile, sest rakenduses kasutatava helifaili tunnuste saamise teek, librosa toetab ainult .wav faile[22]. Kui helifaili aadress kõvakettal on teada, siis on võimalik vajutada nuppu, et teada saada helifaili žanr.

Teiseks, kui kasutaja vajutab nuppu, et teada saada helifaili žanr, annab rakendus helifaili aadressi järgmisele komponendile. Kasutades librosat avab programm helifaili ning teeb helifailist ühe suure MFCC (täpsemalt saab punktist 4.3 lugeda). Järgmisena tükeldatakse MFCC ära väiksemateks MFCCdeks. Tüki suurus oleneb treenitud närvivõrgust. Töös on kasutatud 30-sekundilisi ja 10-sekundilisi heliklippe ehk üks pikk helifail jagatakse mitmeks väiksemaks klipiks.

Kolmandas etapis antakse tükeldatud helifaili MFCCd edasi närvivõrku. Närvivõrk leiab iga üksiku MFCC klipi žanri. Iga MFCC klippile määratakse tõenäosus, millisesse žanrisse see võiks kuuluda. See tähendab, et ei anta kindel žanr vaid tõenäosused iga žanri kohta. Näiteks võib üks lugu kuuluda 70% hiphopi, 20% popmuusikasse ja 10% rokkmuusikasse.

Neljandas etapis liidetakse kõik MFCC žanrid kombineerimismeetodiga kokku. Selle tulemusena saab teada, millisesse žanrisse helifail kuulub.

4.2 Andmestik

Töös kasutatav andmestik on FMA andmestik[23]. Selles andmestikus on kokku 106 574 helifaili, mis on jagatud 16 peažanri vahel. See on suurim muusikaandmestik, kust on võimalik saada helifaile. Suuremad andmestikud, nagu Million Song on ainult muusikalugude metaandmed ja mitte helifailid. GTZAN andmestik oli üks esimesi avalikult kättesaadav muusika andmestikke ning seetõttu siiamaani üks kõige populaarseimaid. FMA andmestik on GTZANi andmestikust parem, sest FMA andmestikus on palju rohkem helifaile [10], millega on võimalik närvivõrku treenida.

Käesolevas töös on kasutusel FMA *small* andmestik, kus on 8000 helifaili, 8 erinevas žanris, kuid töös on kasutusel ainult 5 žanri. Iga 5 žanri kohta on 1000 helifaili, mis annab usaldusväärse tulemuse žanri määramiseks. Selleks, et teada saada, millised helifailid kuuluvad millisesse žanri, on vaja kasutada metaandmeid. Selle lugemiseks on FMA andmestiku koostajatel tehtud Pythoni skript, mis suudab metaandmeid lugeda ning tagastada vajaliku info. Kasutades skripti on võimalik koostada eraldi failid iga soovitud žanri kohta. Nendes failides on teatud žanrisse kuuluvad helifaili ID-d.

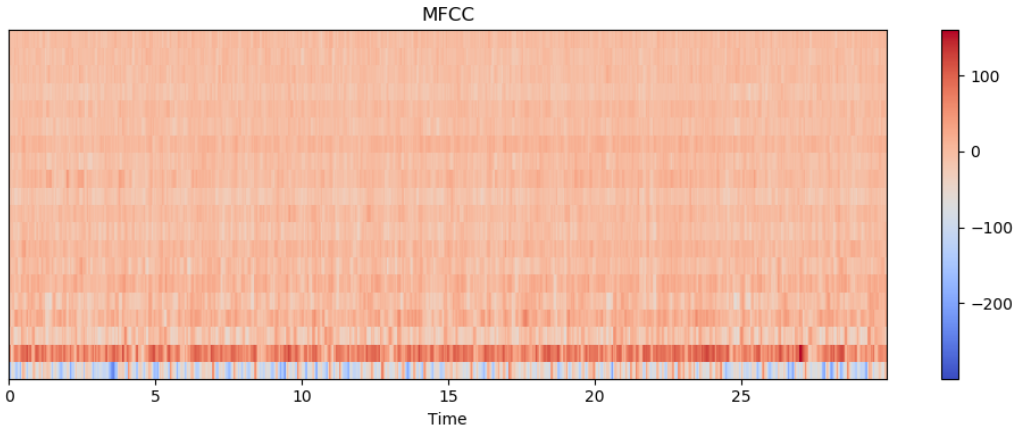
Enne kui faile saab töödelda, on vaja need õigesse formaati teisendada. Kõik FMA andmestiku helifailid on .mp3 formaadis, kuid librosa töötleb .wav faile. Seetõttu kasutades freac- free audio converterit[24] teisendati .mp3 formaadis helifailid wav formaati.

4.3 MFCC

Helifailist on vaja saada vajalik info, et närvivõrk suudaks aru saada, millisesse žanrisse helifail kuulub. Kasutades librosa teeki on võimalik helifailist palju erinevaid tunnuseid saada. Käesolevas töös on kasutatud *Mel-frequency cepstral coefficients* ehk MFCCd. See on kõige populaarsem viis muusika klassifitseerimiseks [25].

MFCC loomiseks jagatakse kasutatav helifail väiksemateks tükkideks ning igale tükile arvutatakse eraldi selle MFCC väärtus[26]. See tähendab, et igale helifaili tükile tehakse diskreetne Fourieri teisendus ning jäetakse alles ainult spektrumi amplituudi logaritm. Seejärel tehakse sellele *Mel scaling* ning ühtlustamine ehk viiakse sagedused kokku. Lõpuks kasutatakse *Discrete Cosine Transform*, et vähendada parameetrite arvu [27].

MFCCd on võimalik kujutada nagu pilti (Joonis 2) ning seda on võetud ka arvesse närvivõrgu valikul.



Joonis 2 MFCC

FMA andmestiku koostajad uurisid ka erinevate muusikatunnuste efektiivsust žanrite klassifitseerimisel. Nad testisid palju erinevad tunnused, nagu MFCC, *Zero-crossing rate*, *RMS energy*, *Chroma* ja veel mitmeid. Kõik tunnused testiti mitmel erineval klassifitseerimise viisil, näiteks nagu k-lähem naaber ja tugivektormasinad ning selgus, et kõige parema tulemuse andis MFCC [10].

Kui anda helifaili asukoht kõvakettal librosale, siis on võimalik helifail avada ning teha MFCC teisendus. Kasutades seda FMA andmestiku teisendatud .wav failide peal, saame librosalt MFCC teisendusest tagasi massiivi suurusega (20, 1291) või (20, 1293). Kõik saadud MFCC massiivid lühendatakse suurusele (20, 1291).

Töödeldes FMA andmesikku tuli välja, et kõik lood ei ole soovitud pikkusega. MFCC teisenduse järel oli kokku 4 massiivi liiga väikese pikkusega. Kolm neist olid hiphop MFCCd ja üks oli rokk MFCC, mis tähendab, et treenimiseks on hiphop MFCCsid 997 ja rokk MFCCsid 999. Üheski teises žanris ei leidunud selliseid MFCC massiive, mis oleksid olnud liiga väikesed.

4.4 Andmete tükeldamine

Käesolevas töös uuritakse, kas andmete tükeldamine tõstab närvivõrgu täpsust. Selle jaoks saadud 30-sekundilised MFCCd librosalt tükeldatakse väiksemateks 10-sekundilisteks MFCC klippideks.

Eelnevalt on uuritud, kas pikema loo tükeldamine ning pärast vastuste kombineerimine annab täpsema tulemuse. Selgus, et keskmiselt täpsus paranes 1% kuni 7% . Kuid seal töös kasutati tervet helifaili ning tükeldati see kolmeks 30. sekundiliseks klippiks, algusest, keskelt ning lõpust. Töös selgus, et tükeldamine võimaldab saada parema tulemuse.[28]

Kui tükeldada 30-sekundilised MFCCd 10-sekundilisteks MFCCks, kolmekordistatakse treeningandmete hulka, kuid maht jääb samaks. Saadud andmetega peaks saavutama ka suurema täpsuse, sest mudelit on võimalik kauem treenida ilma ülesobitamiseta. Kuna FMA andmestikus on väga palju helifaile, oleks võimalik võtta rohkem 30-sekundilisi helifaile, et suurendada andmehulka. See oluliselt suurendaks andmemahtu ning treenimise aega. Sellepärast peaks andmete tükeldamine olema palju efektiivsem, sest andmete maht jääb samaks.

Pärast tükeldamist ning iga žanri teada saamist on vaja vastused kuidagi kombineerida. Selleks on paar erinevat võimalust, kuid töös testiti kahte: hääletusmeetod ning liitmismeetod.

Kõik väiksemad MFCCd on tükeldatud ilma kattuvuseta, mis tähendab, et andmete mahtu ei ole muudetud.

Hääletusmeetodis iga sama MFCC 10-sekundiline klipp sai vastuseks žanri. Lõplikuks vastuseks valitakse kõige sagedamini esinenud žanr. Näiteks, kui 30-sekundiline rokk MFCC tükeldati ja kaks klippi said vastuses rokk žanri kuid üks instrumentaal žanri, siis lõplik vastus oleks ikka rokk žanr.

Liitmismeetod liidab kõik sama MFCC žanri vastused kokku. Liidetakse kõik tõenäosused kokku ning määratakse kõige suurema tõenäosusega žanr lõplikuks vastuseks.

4.5 Konvolutsiooniline närvivõrk

Peamine põhjus konvolutsioonilise närvivõrgu valikuks on MFCC andmete kuju. MFCC on must-valge pilt heli spektrist ning konvolutsioonline närvivõrk on peamiselt disainitud piltide ära tundmiseks. Konvolutsiooniline närvivõrk suudab pildi pealt üles leida teatud

tunnused, mida seostab mingi žanriga. Need tunnused võivad olla madalatasemelised nagu pildil olevad servad või kõrgetasemelised.

Konvolutsioonilise närvivõrgu korral valiti kihtide üldine ülesehitus, selle jaoks lähtuti mitmest erinevast aspektist.

Eelnevalt on uuritud, et ReLU ja dropout-meetod suurendavad närvivõrgu täpsust muusika klassifitseerimisel keskmiselt 1% - 2% võrreldes sigmoid aktiveerimisfunktsiooniga. Kuid dropout-meetodi kasutamine parandas täpsust ainult suuremates närvivõrkudes, väiksemates täpsus langes. Lisaks vähendab ReLU kasutamine märkimisväärselt treenimise aega, mis suurematel andmestikel on väga suur eelis [29].

Kuigi ReLUl on väga palju häid omadusi ning see on väga efektiivne aktiveerimisfunktsioon, siis peamine probleem ReLUga on kiire ülesobitamine. Selle vastu aitab dropout-meetodi lisamine närvivõrgu kihtidele. Dropout lisab närvivõrgu treenimisel müra, mis kustutab teatud protsendi aktiveerimisi kihilt. Peamine eelis selles seisneb väga keeruliste tunnuste tundmise ära kustutamisel, sest keerukamad tunnused võivad olulised olla ainult treenimisandmetel [29].

Lisaks on närvivõrgus kasutusel ka L2 regulariseerimine, mis vähendab parameetrite väärtust. See tähendab, et lihtsamad lahendused ei ülesobitu nii kiiresti [30]. Konvolutsiooniliste närvivõrkude korral tavaliselt regulariseerimise suurus on väga väike. Võimalik, et väike L2 regulariseerimine võimaldab närvivõrgul paremini õppida ning vähendab treenimise viga [31].

Töös kasutatavas närvivõrgus on üheksa kihti (Tabel 1). Kuus nendest on konvolutsioonilised kihid koos ahenduskihtidega ning viimased kolm on täielikult ühendatud kihid. Üldine arhitektuur konvolutsioonlilistel kihtidel on iga kihiga suurenev filtrite arv ning dropout-meetodi protsendi suurenemine suurematel kihtidel. Täielikult ühendatud kihtide suurus väheneb iga kihiga, kuni viimase kihini, kus on viis väljundit.

Tabel 1. Närvivõrgu kihtide jaotus

Närvivõrgu kihid
1. Konvolutsiooniline kiht, 24 filtrit, L2 regulariseerimine, aktiveerimisfunktsioon ReLU, kerneli suurus 3x3 Dropout-meetod 10%
2. Konvolutsiooniline kiht, 32 filtrit, L2 regulariseerimine, aktiveerimisfunktsioon ReLU, kerneli suurus 3x3 Dropout-meetod 10%, *max*-ahenduskiht suurus 2x2 ja samm 2x2
3. Konvolutsiooniline kiht, 48 filtrit, L2 regulariseerimine, aktiveerimisfunktsioon ReLU, kerneli suurus 3x3 Dropout-meetod 30%, *max*-ahenduskiht suurus 2x2 ja samm 2x2
4. Konvolutsiooniline kiht, 62 filtrit, L2 regulariseerimine, aktiveerimisfunktsioon ReLU, kerneli suurus 3x3 Dropout-meetod 40%, *max*-ahenduskiht suurus 2x2 ja samm 2x2
5. Konvolutsiooniline kiht, 128 filtrit, L2 regulariseerimine, aktiveerimisfunktsioon ReLU, kerneli suurus 3x3 Dropout-meetod 30%, *max*-ahenduskiht suurus 2x2 ja samm 2x2
6. Konvolutsiooniline kiht, 256 filtrit, L2 regulariseerimine, aktiveerimisfunktsioon ReLU, kerneli suurus 3x3 Dropout-meetod 30%, *max*-ahenduskiht suurus 1x2 ja samm 1x2
7. Täielikult ühendatud kiht 512 neuroni, L2 regulariseerimine, aktiveerimisfunktsioon ReLU, Dropout-meetod 40%
8. Täielikult ühendatud kiht 256 neuroni, L2 regulariseerimine, aktiveerimisfunktsioon ReLU, Dropout-meetod 20%
9. Täielikult ühendatud kiht 5 neuroni, aktiveerimisfunktsioon *softmax*

Iga konvolutsiooniline kiht kasutab L2 regulariseerimist, aktiveerimisfunktsioonina ReLUt, iga kihi *kerneli* suurus on 3x3. Igal kihil on äärise korral kasutatud valik sama.

See tähendab, et tunnuskaardi suurus ei vähene kihist kihti. Kõik kasutatavad ahenduskihid on *max*-ahenduskihid, nende suurus ja samm on 2x2. Viimase ahenduskihi suurus on 1x2, sest dimensioonid ei võimaldanud kasutada 2x2 ahenduskihti.

Viimased kolm kihti on täielikult ühendatud kihid. Kahel eelmisel täielikult ühendatud kihil on aktiveerimisfunktsioon ReLU ning kasutatakse ka dropout-meetodit. Žanrite järjekord viimase kihi väljundite suhtes sõltub sisse antud andmete järjekorrast. Viimasel kihil dropouti pole ning aktiveerimisfunktsioon on *softmax.*

Närvivõrgu treenimisel on kasutatud Adam optimeerimismeetodit ja selle õpisamm on seatud 0.001. See samm on piisavalt väike, et mudel suudab õppida ja treenimisaeg ei lähe liiga pikaks. Kaduna on kasutatud Kerase poolt kasutusel olev *categorical crossentropy*, mis on loodud klassifitseerimisülesannete jaoks. *Categorical crossentropy-i* jaoks peab ka treeningandmed viima õigesse formaati, ehk *onehot* kodeerimisse. See tähendab, et sisendile peab vastama soovitud väljund. Näiteks iga rokk MFCC puhul peab treenimisandmetes soovitud väljund olema [0, 0, 0, 1, 0].

4.6 Tehnoloogiad

Konvolutsioonilise närvivõrgu jaoks on kasutusel Keras ja Tensorflow. Keras annab väga hea kõrgetasemelise API Tensorflowle, mis võimaldab väga lihtsalt prototüüpida ning testida erinevaid närvivõrgu mudeleid. Keras võimaldab väga vähese koodiga käima panna lihtsa närvivõrgu. Lisaks on kihtide lisamine mugav.

Helifailidest andmete kättesaamiseks on kasutusel librosa. Librosal on võimalik otse helifaili lugeda ning teha see MFCCks. Librosa kasutab helifailide lugemiseks audioread teeki, mis vaikimisi toetab ainult .wav faile. Seetõttu on ka vajalik FMA andmestikust .mp3 failid teisendada .wav failideks.

Freac- free audio converteriga teisendati .mp3 formaadis helifailid, õigesse .wav formaati.

Närvivõrgu treenimiseks on kasutatud Google'i poolt tehtud Pythoni veebiarenduse keskkonda Colab research. Kuna Colab töötab Google'i serverite peal, siis kõik arvutused toimuvad pilves, mis teeb treenimisele kuluva aja oluliselt lühemaks. Colab kasutab failide lugemiseks ja kirjutamiseks kasutaja Google Drive'i, mis on mugav, sest treenimisandmed on suured ning Driveis on piisavalt ruumi nende hoidmiseks. Kui

treenida konvolutsioonilist närvivõrku, on vaja kasutada GPU kiirendust, sest ainult CPU peal võtab treenimine liiga kaua aega. Kuid peamine probleem GPU kasutamisel on liiga väike mälu, kuid Colabi kasutades ei teki sellega probleeme.

Kasutajaliidese tegemiseks on kasutatud PyQT. PyQT võimaldab väga kiiresti disainida programmile kasutajaliidese. Lisaks on võimalik kasutada QT Designerit, mis võimaldab disainimise ajal näha, milline kasutajaliides välja näeb. PyQTs toimub UI ja rakenduse sidumine väga lihtsalt.

5 Tulemused

Selles peatükis toimub tulemuste analüüs. Võrreldakse 30-sekundiliste helifailidega ja 10-sekundiliste helifailidega treenitud närvivõrke. Lisaks vaadatakse ka 10-sekundiliste helifailidega kombineerimismeetodeid.

5.1 Treenimine ja valideerimine

Mõlemate, 30- ja 10-sekundiliste helifailide korral kasutatakse sama konvolutsioonlist närvivõrku, mida analüüsiti 4.5 peatükis. Ainuke parameeter, mida muudetakse, on epohhide arv, sest andmekogus on mõlemal juhul erinev, ning ülesobitumine toimub erinevatel aegadel.

Mõlemate mudelite korral on andmed jagatud 8:1:1 suhtes, ehk 80% andmeid on kasutatud treenimiseks, 10% valideerimiseks ning 10% testimiseks. Lisaks on eraldi testimiseks tehtud 500-st uuest 30-sekundilisest helifailidest MFCCd. Need 500 MFCCd on saadud FMA *medium* andmestikust. Neid helifaile ei ole närvivõrk enne kasutanud. Igast žanrist on 100 MFCCd, see on piisavalt suur andmehulk, et saada teada närvivõrgu täpsus. Eraldi testimine on tehtud sellepärast, et testida mõlemat mudelit samade andmete peal, et näha, kumb saab suurema täpsuse ning näha erinevusi mudelites.

Õigete epohh arvude määramine on väga tähtis. Kui see arv on liiga väike, siis mudel ei õpi piisavalt. Samas, kui see on liiga suur, siis mudel on ülesobitatud ning täpsus on seetõttu väike. Õigete epohhide arvu määramiseks treenitakse väga suure epohhide arvuga ning selgitatakse, millal hakkab mudel ülesobitama, ning seejärel treenitakse õigete epohhidega.

Täpsus on defineeritud kui tõsipositiivsed vastused jagatud tõsipositiivsed vastused pluss väärpositiivsed vastused [32].

Saagis on defineeritud kui tõsipositiivsed vastused jagatud tõsipositiivsed vastused pluss väärnegatiivsed vastused [32].

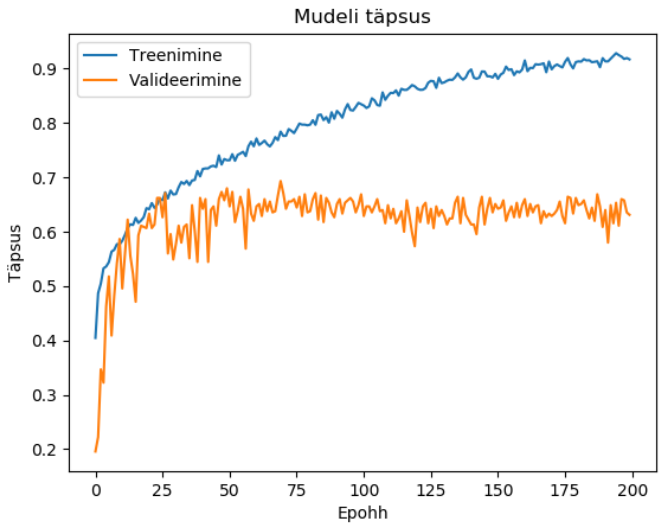
Enamus närvivõrke, mis treenitakse GTZAN andmestiku peal saavutavad 10 žanriga umbes 80% saagise [29]. Kuid enamus närvivõrke, mida on treenitud FMA andmestiku peal saavutavad palju madalama saagise. See võib olla tingitud mitmest erinevast

aspektist, nagu näiteks FMA andmestikus olevad helifailid võivad olla mitmele žanrile väga sarnased. Enamus töid, mis on kasutanud FMA andmestikku, saavutavad 16 žanriga 60% saagise [33]. Käesolevas töös, oodatav tulemus umbes 70% juures, sest kasutusel on 5 žanri.

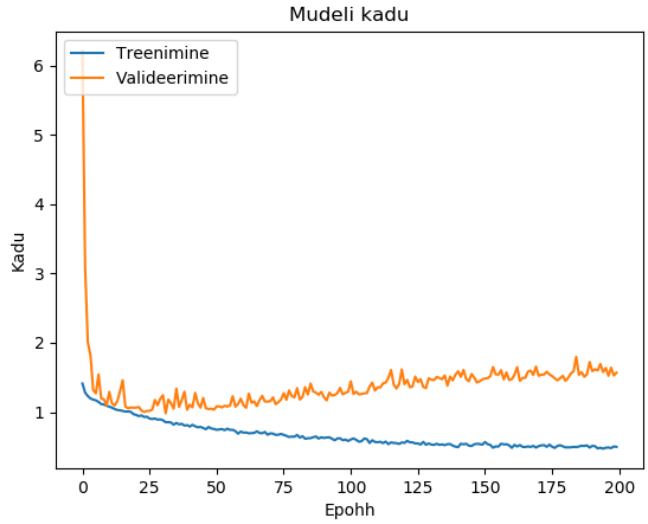
5.2 30-sekundiliste helifailidega treenitud mudeli analüüs

Selles peatükis analüüsitakse 30-sekundiliste helifailidega treenitud konvolutsioonilise närvivõrgu mudelit. Epohhide arv on määratud vaadates, millal mudel hakkab ülesobitama.

Nagu on näha jooniselt 3, siis mudeli valideerimise täpsus on jäänud 60% juurde ning jooniselt 4 on näha, et kadu on pärast 60 epohhi tõusma hakanud. Mudeli treenimiseks valiti 60 epohhi, see peaks olema piisav, et mudel õpiks soovitud andmed ja saavutaks hea täpsuse.

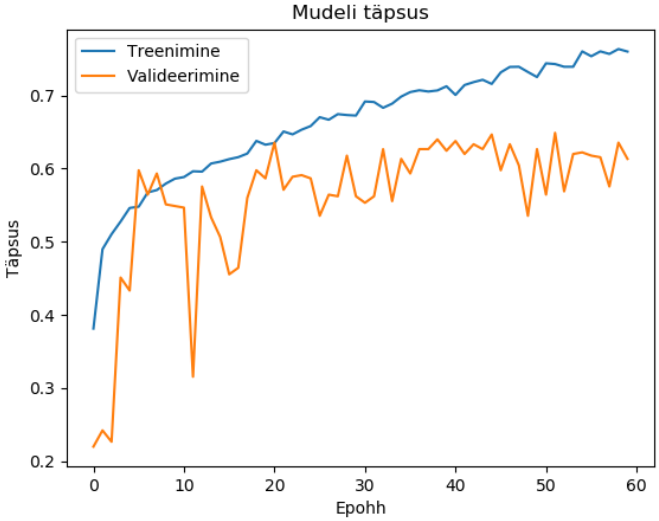


Joonis 3. 30-sekundiliste helifailidega treenitud täpsus 200 epohhiga

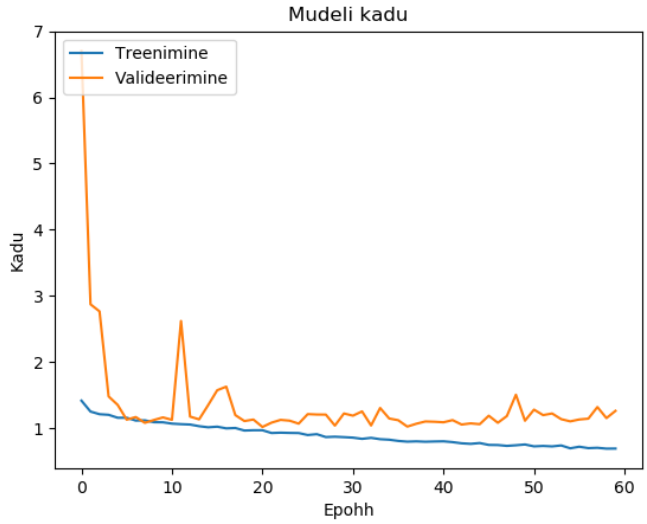


Joonis 4. 30-sekundiliste helifailidega treenitud kadu 200 epohhiga

Pärast mudeli treenimist testis närvivõrk ennast 10% andmetega ning saavutas 61.6% täpsuse ning 1.1 kadu väärtuse. Jooniselt 5 on näha, et mudel saavutab umbes 40 epohhiga 60% valideerimistäpsuse, mis on mudeli maksimaalse täpsuse juures. Jooniselt 6 on näha, et mudeli kadu on jõudnud miinimum oleku juurde.



Joonis 5. 30-sekundiliste helifailidega treenitud täpsus 60 epohhiga



Joonis 6 30-sekundiliste helifailidega treenitud kadu 60 epohhiga

Testides eraldi 500 MFCCga, mis eelnevalt saadi FMA *medium* andmestikust saadi kokku saagis 62% ning täpsus 64% (Tabel 3).

Tabel 2. Segadustemaatriks 30-sekundiliste helifailidega treenitud mudelile

	Folk	Hiphop	Pop	Rokk	Instrumentaal
Folk	59	0	20	10	11
Hiphop	2	66	3	25	4
Pop	11	0	58	11	20
Rokk	21	3	13	50	13
Instrumentaal	2	2	5	15	76

Tabel 3. Täpsus ja saagis 30-sekundiliste helifailidega treenitud mudelile

	Täpsus	Saagis
Folk	62%	59%
Hiphop	93%	66%

	Täpsus	Saagis
Pop	59%	58%
Rokk	45%	50%
Instrumentaal	61%	76%
Kokku	64%	62%

Folkmuusika täpsus on 62% ja saagis 59%. Selle korral on näha, et kõige rohkem ajas närvivõrk folkmuusikat segamini popmuusikaga (Tabel 2). Kordagi ei arvanud mudel folkmuusika korral hiphopi, see on arvatavasti seetõttu, et žanritel on suur erinevus. Mudel klassifitseeris folkmuusikat rokkmuusikana 10 korda ning instrumentaalmuusikana 11 korda.

Hiphopi täpsus on 93% ning saagis 66%. Saagise ja täpsuse suur vahe tuleneb sellest, et mudelil on lihtsam määratleda, kui MFCC ei kuulu hiphopi žanrisse. Hiphopi MFCCde korral ajas mudel neid peamiselt segamini just rokkmuusikaga (Tabel 2). Mudel ei oska rokkmuusikat väga hästi klassifitseerida, mis võimaldab valesid tulemusi saada.

Popmuusika täpsus on 59% ning saagis 58%. Folkmuusika korral pakkus popmuusikat 20 korda valesti (Tabel 2). Popmuusika korral arvas folkmuusikat ainult 10 korda valesti. Kõige rohkem valesti arvas mudel instrumentaalmuusikat. See võib tuleneda sellest, et peamine vahe nendel on vokaalmuusika ning mudel ei tunne täpselt seda ära. Popmuusika korral hiphopi ei arvanud mudel kordagi. Rokk- ja folkmuusikat arvas mudel mõlemat 11 korda valesti.

Rokkmuusika korral saagis on 50% ning täpsus 45%. See on üllatav, sest see žanr peaks olema äratuntav, mis tõttu võiks täpsus olla parem. Kõige rohkem ajas mudel rokkmuusikat segamini folkmuusikaga (Tabel 2), mis on samuti üllatav, sest popmuusika tundub rokile palju lähemal olev, kui folkmuusika. Hiphopiga pakkus mudel valesti 3 helifaili. Pop- ja instrumentaalmuusikaga pakkus mudel mõlemat 13 korda valesti.

Instrumentaalmuusika saagis oli kõige suurem 76%, kuid täpsus oli 61%. See näitab, et kuigi mudel sai hea saagise instrumentaalmuusikaga, pakkus mudel seda väga palju. See

viis instrumentaalmuusika täpsuse alla. See võib tuleneda sellest, et mudel ei saa täpselt vokaalmuusika puudumisest hästi aru. Peamine žanr millega mudel instrumentaalmuusika korral valesti pakkus on rokkmuusika.

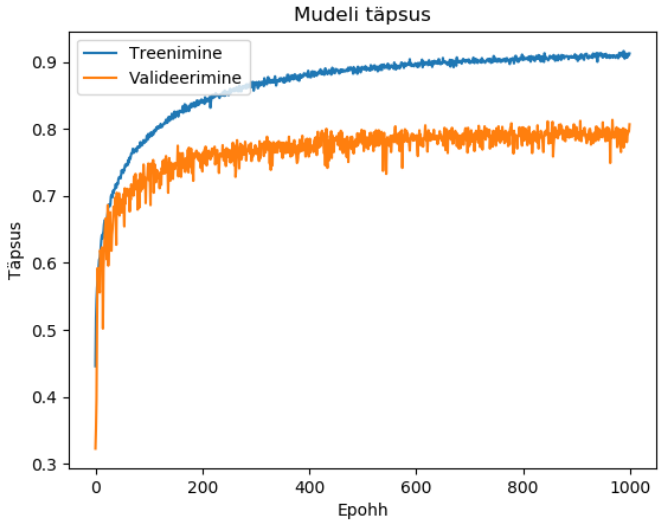
Üldiselt on näha, et kuigi instrumentaalmuusika saagis on kõige suurem, siis mudel arvab seda väga palju, mistõttu on selle täpsus ainult 61%. See-eest mudel saab hiphopist palju paremini aru, sest selle täpsus on 93%. Samas, kõige halvemini sai mudel aru rokkmuusikast. Seetõttu on ka selle täpsus ja saagis teistest halvem. Ülejäänud kahe žanri puhul on tulemused sarnaselt 60% juures.

Kokku saavutati 64% täpsus, kuid loodetud täpsus oli 70%. Peamine põhjus, miks tulemused on oodatust halvemad on kiire ülesobitamine. Närvivõrk sai treenida ainult 60 epohhi, sest peale seda hakkas toimuma ülesobitamine ning mudel ei õppinud rohkem. Närvivõrgu arhitektuuris on kasutusel palju meetodeid selle vähendamiseks, kuid peamine probleem on andmete vähesus. Kui oleks rohkem andmeid võimaldaks see närvivõrgul ka kauem treenida, mis aitaks ülesobitamise vastu.

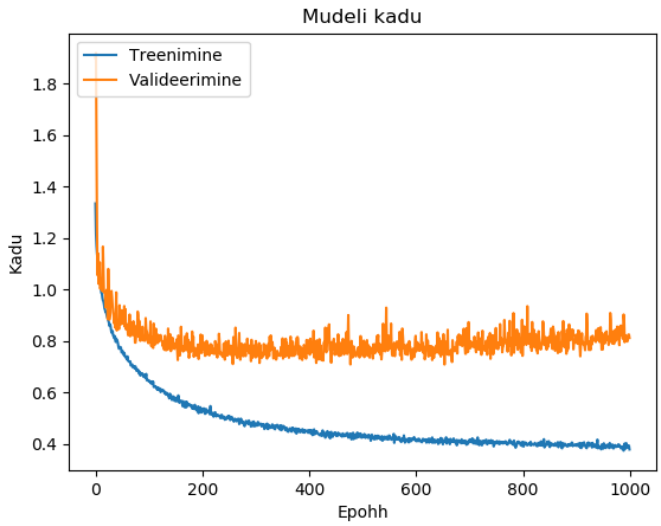
5.3 10-sekundiliste helifailidega treenitud mudeli analüüs

Käesolevas peatükis analüüsitakse 10-sekundiliste helifailidega treenitud konvolutsioonilist närvivõrgu mudelit. Esmalt on samuti vajalik määrata epohhide arv, ning selleks treenitakse uuesti mudel väga suurte epohhidega, et näha, millal hakkab toimuma ülesobitamine.

Nagu jooniselt 7 ja 8 näha hakkab ülesobitamine toimuma palju hiljem, kui 30-sekundiliste helifailidega treenitud mudelil. Seda on hästi näha joonisel 6, kus kadu on palju hiljem tõusma hakanud, kui joonisel 4. See on tingitud treenimisandmete suurendamisest, sest kõik ülejäänud parameetrid on jäänud samaks. 30-sekundiliste helifailidega treenitud mudelil kasutati 60 epohhi. Treeningandmete suurendamine võimaldab 10-sekundiliste helifailidega treenitud mudelil kasutada 200 epohhi. 200 epohhi on piisav, et mudel saaks suure täpsuse ning mille treenimine ei võtaks liiga kaua aega.

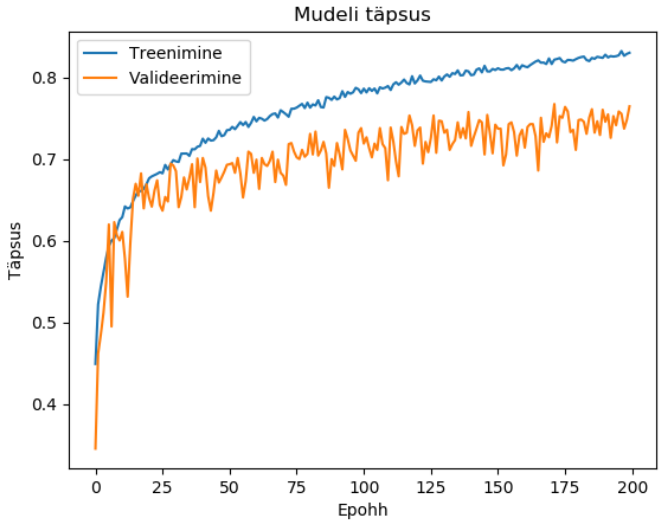


Joonis 7 10-sekundiliste helifailidega treenitud täpsus 1000 epohhiga

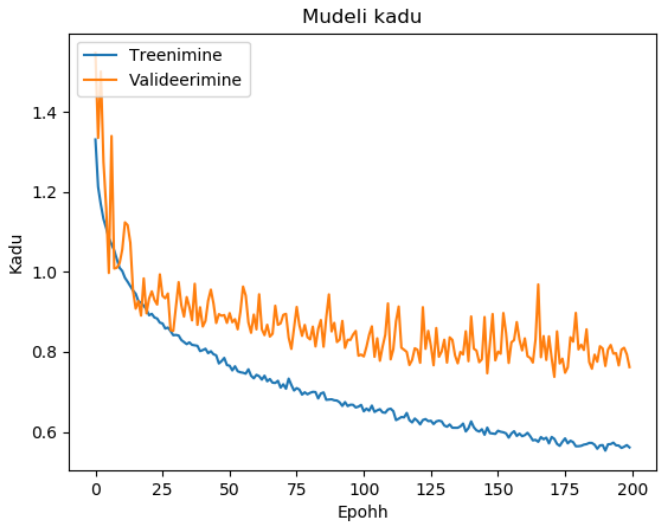


Joonis 8 10-sekundiliste helifailidega treenitud kadu 1000 epohhiga

Pärast 200. epohhi testiti mudelit 10% 10-sekundiliste MFCCdega. Mudel saavutas 76.5% täpsuse ning 0.75 kadu. Võrreldes 30-sekundiliste helifailidega treenitud mudeliga on see väga hea tulemus. Jooniselt 9 ja 10 on näha, et mudeli täpsus ja kadu koguaeg, terve 200 epohhi peale paranesid. Viimased 50 epohhi paranesid täpsus ja kadu aeglasemalt. See läheb ka jooniste 7 ja 8 andmetega kokku, sest sealt on näha, et pärast 200 epohhi ei ole täpsus ja kadu paranenud.



Joonis 9. 10-sekundiliste helifailidega treenitud täpsus 200 epohhiga



Joonis 10. 10-sekundiliste helifailidega treenitud kadu 200 epohhiga

Samuti testiti seda mudelit eraldi sama 500 MFCCga, millega 30-sekundiliste helifailidega mudelit testiti. 10-sekundiliste helifailidega treenitud mudelil oli testimine erinev võrreldes eelmise mudeliga. Üks 30-sekundiline MFCC jagatakse kolmeks 10-sekundiliseks MFCCks. Igale kolmele MFCCle saadakse mudeli poolt tõenäosus, millisesse žanrisse MFCC võiks kuuluda. Järgmisena oleks vajalik kolm vastust kombineerida üheks, et teada saada lõplik žanr. Selleks on töös kasutusel kaks erinevat meetodit, mida analüüsiti peatükis 4.4: liitmismeetod ja hääletusmeetod. Esmalt analüüsitakse liitmismeetodit.

5.3.1 Tulemused kombineerides liitmismeetodiga

Liitmismeetodiga saavutati 73% täpsus ning 71% saagis (Tabel 5), mis on parem tulemus, kui 30-sekundiliste helifailidega treenitud mudelil. Tabelist 4 on näha, et folk ja hiphop muusikat määras mudel väga täpselt ning popmuusikat määras kõige halvemini.

Tabel 4.Segaduste maatriks 10-sekundiliste helifailidega treenitud mudel liitmismeetodiga

	Folk	Hiphop	Pop	Rokk	Instrumentaal
Folk	82	0	8	8	2
Hiphop	4	82	0	12	2
Pop	14	1	53	21	11
Rokk	12	3	5	74	6
Instrumentaal	7	2	6	22	63

Tabel 5. Täpsus ja saagis 10-sekundiliste helifailidega treenitud mudel liitmismeetodiga

	Täpsus	Saagis
Folk	69%	82%
Hiphop	93%	82%
Pop	74%	53%
Rokk	54%	74%
Instrumentaal	75%	63%
Kokku	73%	71%

Folkmuusika puhul on näha, et täpsus paranes 7% ning saagis 23% (Tabel 3 ja 5). Segamini ajab seda žanrit pop- ja rokkmuusikaga. See on sarnane 30-sekundiliste helifailidega treenitud mudeliga.

Hiphopi korral täpsus jäi 93%, kuid saagis tõusis 16% (Tabel 3 ja 5) võrra. Kõige rohkem arvas valesti rokkmuusikat, mis on oodatav, sest sama juhtus 30-sekundiliste helifailidega treenitud mudelil.

Popmuusika saagis langes 5% (Tabel 3 ja 5), mis on väga üllatav, sest selle saagis oli juba madal võrreldes teiste žanritega. See-eest on tõusnud popmuusika täpsus 15% (Tabel 3 ja 5), mis tähendab, et mudelil ei ole enam nii palju valesid vastuseid popmuusikaga. Kõige rohkem ajab popmuusikat segi rokkmuusikaga, kuid see on arvatavasti tingitud instrumentaalmuusika täpsuse suurenemisest.

Rokkmuusika täpsus suurenes 9% ning saagis tõusis 24% (Tabel 3 ja 5), mis on kõige suurem saagise tõus. Jällegi ajab mudel kõige rohkem segamini folkmuusika, seejärel pop- ning instrumentaalmuusika, täpselt nagu 30-sekundiliste helifailidega treenitud mudelil. Kuigi saagis tõusis, on rokkmuusika ebatäpseim võrreldes teiste žanritega

Instrumentaalmuusika korral saagis langes 13%, mis on ka kõige suurem saagise langus, kuid samas tõusis täpsus 14%. Jällegi ajab instrumentaalmuusikat kõige rohkem segamini rokkmuusikaga, täpselt nagu 30. sekundiliste helifailidega treenitud mudelil.

Üldiselt on selgelt näha, et MFCCde tükeldamine on andnud väga suure täpsuse ja saagise suurenemise võrreldes 30-sekundiliste helifailidega treenitud mudeliga. See peamiselt tuleneb treeningandmete suurenemisest, mis vähendab ülesobitust. Keskmiselt kasvas täpsus ja saagis 9%, mis on väga märkimisväärne kasv.

5.3.2 Tulemused kombineerides hääletusmeetodiga

Hääletusmeetodiga saavutati kokku täpsus 71% ja saagis 69% (Tabel 7), mis on 2% halvem, kui liitmismeetodiga saadud täpsus ja saagis. Üldiselt on näha (Tabel 6), et määratud MFCCd on väga sarnased liitmismeetodi tulemustele (Tabel 4).

Tabel 6. Segaduste maatriks 10. sekundiliste helifailidega treenitud mudelil hääletusmeetodiga

	Folk	Hiphop	Pop	Rokk	Instrumentaal
Folk	83	0	8	8	1
Hiphop	7	81	0	11	1
Pop	14	1	55	19	11

	Folk	Hiphop	Pop	Rokk	Instrumentaal
Rokk	17	2	6	70	5
Instrumentaal	11	4	8	20	57

Tabel 7 Täpsus ja saagis 10-sekundiliste helifailidega treenitud mudel hääletusmeetodiga

	Täpsus	Saagis
Folk	63%	83%
Hiphop	92%	81%
Pop	71%	55%
Rokk	55%	70%
Instrumentaal	76%	57%
Kokku	71%	69%

Tabelis 6 on näha, et MFCCde jaotus žanritesse on peaaegu täpselt sama, nagu liitmismeetodis (Tabel 4). See on oodatav, sest erinevad kombineerimismeetodid ei avalda suurt mõju lõpptulemusele. Enamus žanrites täpsus ja saagis langes, kuid rokk- ja instrumentaalmuusika täpsus paranes 1% ning folk- ja popmuusika saagis paranes 1%. Siiski kokku langes täpsus ja saagis 2%, võrreldes liitmismeetodiga(Tabel 5 ja 7).

Üldiselt on selge, et liitmismeetod annab täpsemad tulemused. Seetõttu on see kombineerimismeetod kasutatud ka rakenduses. See tuleneb arvatavasti sellest, et liitmismeetod annab suurema paindlikkuse kui hääletusmeetod. Kui mudel ei leia täpset žanrit MFCCle, siis hääletusmeetodis on valitud žanril suurem mõju kui liitmismeetodis.

5.4 Kasutajaliides

Rakendusele on ka disainitud kasutajaliides, millega on võimalik valida .wav helifail ning saada tagasisidet, millisesse žanrisse see kuulub. Lisaks on ka näha igasse žanrisse kuulumise tõenäosus, mis annab hea ülevaate mudeli tööst.

Kasutajaliides on tehtud kasutades PyQT teeki, mis võimaldab väga mugavalt ja lihtsalt disainida kasutajaliideseid. Esmalt on kasutatud PyQT poolt saadaval olevat *designer*, mis võimaldab kasutajaliidese luua ja seda samal ajal näha. See teeb kasutajaliidese tegemise lihtsamaks ja kiiremaks.

Järgmisena on vaja siduda kasutajaliides mudeliga, selleks annab kasutajaliides saadud loo asukoha edasi helifaili lugejale. See lugeja teeb helifailist MFCC ning tükeldab kõik MFCC sobivaks suuruseks. Järgmisena antakse tükeldatud MFCCd mudelile ning mudel arvab iga MFCC žanri tõenäosused. Viimasena kombineeritakse saadud vastused lõplikuks vastuseks. Sellel mudelil on kasutusel liitmismeetod, sest see osutus täpsemaks. Kui vastus käes, kuvatakse saadud vastus kasutajaliidesele.

Joonisel 11 on näha rakenduse kasutajaliides ning vastuse kuvamise viis. Hiphop laulu korral on näha, et kõige rohkem arvab, et laul on 57.8% hiphop žanris ning 31% rokk žanris. Sarnane tulemus kajastub ka tabelis 4, kus kõige rohkem ajab mudel hiphopi segamini rokkmuusikaga.

Joonis 11.Rakenduse kasutajaliides, kus on näidatud mudeli vastus laulule Eminem Puke

Rakendus annab töö käigus disainitud närvivõrgu mudelile ka reaalse kasutusvõimaluse. See teeb mudeli kasutamise ning testimise palju mugavamaks.

5.5 Edasiarendus

Antud töös kirjeldatud rakendusele on mitmeid erinevaid edasiarenduse viise. Autor toob välja mõned võimalused.

Lisada rakendusele võimalus kasutada erinevas formaadis olevaid helifaile, näiteks .mp3.

Katsetada tükeldamist ülekattuvate MFCC lõikudega. See suurendaks andmete mahtu, kuid samas oleks ka rohkem treenimsandmeid. Lisaks leida parim tükeldamise suurus MFCCle, mis annaks parema tulemuse, kui 10 sekundiliste MFCCdega.

6 Kokkuvõte

Käesoleva töö käigus koostati konvolutsiooniline närvivõrk, mis suutis tuvastada helifaili žanri 73% täpsusega.

Töös on kasutusel FMA andmestik *small*, kus on igale žanrile 1000 helifaili, mis on 30 sekundit pikad. Saadud helifailidest tehti librosa teekiga MFCCd. Töös uuriti, kas antud MFCC tükeldamine väiksemateks osadeks tõstab närvivõrgu täpsust. 30-sekundilised MFCCd tükeldati väiksemateks, 10-sekundilisteks MFCCdeks. See suurendas treenimisandmeid ning seetõttu vähendas närvivõrgu ülesobitumist. Lisaks testiti ka erinevaid vastuste kombineerimismeetodeid.

Töös testiti kõik mudelid samal 500 helifailil, mis saadi FMA *medium* andmestikust. See võimaldab saada võimalikult täpse tulemuse, ning näha selgeid erinevusi mudelite vahel.

Parimaks tulemuseks oli 73% täpsus ning 71% saagis, mille andis 10-sekundiliste helifailidega treenitud mudel, kasutades kombineerimiseks liitmismeetodit. 30-sekundiliste helifailidega treenitud mudeli täpsus oli 64% ja saagis 62%, mis tähendab, et MFCCde tükeldamine suurendas mudeli täpsust ja saagist 9%. Lisaks katsetati töös hääletusmeetodil kombineerida vastuseid. Selle täpsus oli 71% ja saagis 69%, mis selgelt näitab, et liitmismeetod on parem kombineerimis viis.

Kõige paremini said mudelid aru hiphop žanrist, mis on arvatavasti seetõttu, et see on teistest erinev. Kõige ebatäpsem oli rokkmuusika, selle võimalik põhjus on popmuusika sarnasus rokkmuusikale.

Kokkuvõtlikult on näha, et rakenduses kasutatava närvivõrgu täpsust oli tunduvalt võimalik suurendada MFCCde tükeldamisega 10-sekundilisteks MFCCdeks. Valminud rakendus suudab hea täpsusega tuvastada helifaili žanri, viiest eeldefineeritud žanrist.

Kasutatud kirjandus

[1] Tom LH. Li, Antoni B. Chan and Andy HW. Chun " Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network“, researchgate , 2019. [Online]. Available: https://www.researchgate.net/profile/Antoni_Chan2/publication/44260643_Automatic_Musical_Pattern_Feature_Extraction_Using_Convolutional_Neural_Network/links/02e7e523dac6bb86b0000000/Automatic-Musical-Pattern-Feature-Extraction-UsingConvolutional-Neural-Network.pdf. [Accessed: 26- Feb- 2019].

[2]"What does music genre mean?", *Definitions.net*, 2019. [Online]. Available: https://www.definitions.net/definition/music+genre. [Accessed: 08- May- 2019].

[3]"What Is Pop Music?", *ThoughtCo*, 2019. [Online]. Available: https://www.thoughtco.com/what-is-pop-music-3246980. [Accessed: 08-Apr- 2019].

[4] A. Shin et al., "MELODY GENERATION FOR POP MUSIC VIA WORD REPRESENTATION OF MUSICAL PROPERTIES", *Arxiv.org*, 2019. [Online]. Available: https://arxiv.org/pdf/1710.11549.pdf. [Accessed: 22- Apr- 2019].

[5]“rock music", TheFreeDictionary.com, 2019. [Online]. Available: https://www.thefreedictionary.com/rock+music. [Accessed: 10- Apr- 2019].

[6]"Rock Music's Origins: A 1940s Blending of Country and Blues", *ThoughtCo*, 2019. [Online]. Available: https://www.thoughtco.com/what-is-rock-music-2898293. [Accessed: 08-Apr- 2019].

[7]"Rock Music | Encyclopedia.com", Encyclopedia.com, 2019. [Online]. Available: https://www.encyclopedia.com/history/encyclopedias-almanacs-transcripts-and-maps/rock-and-roll. [Accessed: 11- Apr- 2019].

[8]"Rap | music", Encyclopedia Britannica, 2019. [Online]. Available: https://www.britannica.com/art/rap. [Accessed: 16- Apr- 2019].

[9]"instrumental wiki | Last.fm", Last.fm, 2019. [Online]. Available: https://www.last.fm/tag/instrumental/wiki. [Accessed: 17- Apr- 2019].

[10] M. Defferrard, K. Benzi, P. Vandergheynst and X. Bresson, "FMA: A DATASET FOR MUSIC ANALYSIS", *Arxiv.org*, 2019. [Online]. Available: https://arxiv.org/pdf/1612.01840.pdf. [Accessed: 24- Apr- 2019].

[11] "Free Music Archive", *Freemusicarchive.org*, 2019. [Online]. Available: http://freemusicarchive.org/. [Accessed: 07- May- 2019].

[12] "Folk Rock",Rateyourmusic, 2019, [Online], Available: https://rateyourmusic.com/genre/Folk+Rock/ [Accessed: 17- Apr- 2019].

[13] "Folk Pop",Rateyourmusic, 2019, [Online], Available: https://rateyourmusic.com/genre/Folk+Pop/ [Accessed: 19- May- 2019].

[14] M. Haggblade, Y. Hong and K. Kao, "Music Genre Classification", *Cs229.stanford.edu*, 2019. [Online]. Available: http://cs229.stanford.edu/proj2011/HaggbladeHongKao-MusicGenreClassification.pdf. [Accessed: 07- May- 2019]

[15] "Research Paper on Basic of Artificial Neural Network", Ijritcc.org, 2019. [Online]. Available: http://www.ijritcc.org/download/Research%20Paper%20on%20Basic%20of%20Artificial%20Neural%20Network.pdf. [Accessed: 18- Apr- 2019].

[16] "A study of Neural Networks for Classification: An Survey", *Pdfs.semanticscholar.org*, 2019. [Online]. Available: https://pdfs.semanticscholar.org/d3b0/ae5354bb663ac11f6e159c32a98fef08639e.pdf. [Accessed: 19- Apr- 2019].

[17] Y. Srivastava, V. Murali and S. Dubey, "A Performance Comparison of Loss Functions forDeep Face Recognition", Arxiv.org, 2019. [Online]. Available: https://arxiv.org/pdf/1901.05903.pdf. [Accessed: 21- Apr- 2019].

[18] D. Kingma and J. Ba, "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION", Arxiv.org, 2019. [Online]. Available: https://arxiv.org/pdf/1412.6980.pdf. [Accessed: 22- Apr- 2019].

[19] B. Ding, H. Qian and J. Zhou, "Activation functions and their characteristics in deep neural networks - IEEE Conference Publication", Ieeexplore.ieee.org, 2019. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8407425. [Accessed: 21- Apr- 2019].

[20] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN)and Long Short-Term Memory (LSTM) Network", Arxiv.org, 2019. [Online]. Available: https://arxiv.org/pdf/1808.03314.pdf. [Accessed: 20- Apr- 2019].

[21] R. Yamashita, M. Nishio, R. Kinh Gian Do, K. Togashi, „Convolutional neural networks: an overview and application in radiology“, Link.springer, 2019, [Online], Available: https://link.springer.com/article/10.1007/s13244-018-0639-9 [Accessed: 20- Apr- 2019].

[22]"librosa/librosa", *GitHub*, 2019. [Online]. Available: https://github.com/librosa/librosa. [Accessed: 23- Apr- 2019].

[23] "mdeff/fma", GitHub, 2019. [Online]. Available: https://github.com/mdeff/fma. [Accessed: 22- Apr- 2019].

[24] "fre:ac - free audio converter", *Freac.org*, 2019. [Online]. Available: https://www.freac.org/. [Accessed: 25- Apr- 2019].

[25] C. Yeh and Y. Yang, "Supervised Dictionary Learning for Music Genre Classification", *http://delivery.acm.org*, 2019. [Online]. Available: https://dl.acm.org/citation.cfm?id=2324859. [Accessed: 25- Apr- 2019].

[26] C. Lee, J. Shih, K. Yu and J. Su, "Automatic Music Genre Classification using Modulation Spectral Contrast Feature - IEEE Conference Publication", *Ieeexplore.ieee.org*, 2019. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4284622. [Accessed: 25- Apr- 2019].

[27] B. Logan, "Mel Frequency Cepstral Coefficients for Music Modeling", *Musicweb.ucsd.edu*, 2019. [Online]. Available: http://musicweb.ucsd.edu/~sdubnov/CATbox/Reader/logan00mel.pdf. [Accessed: 25- Apr- 2019].

[28] Silla Jr, C., Kaestner, C. and Koerich, A. (2019). *Automatic music genre classification using ensemble of classifiers - IEEE Conference Publication*. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4414136 [Accessed 26 Apr. 2019].

[29] S. Sigtia and S. Dixon, "Improved music feature learning with deep neural networks - IEEE Conference Publication", *Ieeexplore.ieee.org*, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/6854949. [Accessed: 27- Apr- 2019].

[30] "ACHIEVING STRONG REGULARIZATION FOR DEEP NEURAL NETWORKS", *Openreview.net*, 2019. [Online]. Available: https://openreview.net/pdf?id=Bys_NzbC-. [Accessed: 28- Apr- 2019].

[31] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *Papers.nips.cc*, 2019. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf. [Accessed: 28- Apr- 2019].

[32] "Precision-Recall — scikit-learn 0.21.1 documentation", Scikit-learn.org, 2019. [Online]. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html. [Accessed: 17- May- 2019].

[33] J. Kim, X. Serra, M. Won and C. Liem, "Transfer Learning of Artist Group Factors to Musical Genre Classification", *http://delivery.acm.org*, 2019. [Online]. Available: https://dl.acm.org/citation.cfm?doid=3184558.3191823. [Accessed: 27- Apr- 2019].