VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

SYSTÉM PRO DOPORUČOVÁNÍ SKLADEB

MUSIC RECOMMENDATION SYSTEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **RADOSLAV PÁLENÍK**
AUTHOR

VEDOUCÍ PRÁCE — **doc.Ing. FRANTIŠEK ZBOŘIL, Ph.D.**
SUPERVISOR

BRNO 2022

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav inteligentních systémů (UITS) Akademický rok 2021/2022

Zadání bakalářské práce

Student: **Páleník Radoslav**
Program: Informační technologie
Název: **Systém pro doporučování skladeb**
Music Recommendation System
Kategorie: Umělá inteligence

Zadání:

1. Prozkoumejte kolekce trénovacích a testovacích dat pro systém, který má pro hudební skladby populární hudby rozpoznat jejich žánr. Dále vytvořte kolekci příkladů skladeb, které jsou pro jednu nebo více osob oblíbené či neoblíbené.
2. Nastudujte existující systémy pro rozpoznávání žánru či doporučování skladeb a zvolte vhodné metody pro jejich doporučování.
3. Vytvořte aplikaci používající metody zvolené v předchozím bodě, která jednak bude schopna trénovat takový model a jednak ji bude možné používat pro sledování zadaného souboru skladeb a vyhodnocování, zda-li by se některé mohly uživateli líbit.
4. Vyhodnoťte fungování vytvořeného systému a navrhněte její možné další vylepšení do budoucna.

Literatura:

- K. Maheshkumar, Music Genre Recognition Using Deep Neural Networks and Transfer Learning, Interspeech.2018-2045, 2018
- Vaidya, K.: Music Genre Recognition using Convolutional Neural Networks, online [10.18.2021] https://towardsdatascience.com/music-genre-recognition-using-convolutional-neural-networks-cnn-part-1-212c6b93da76, 2020

Pro udělení zápočtu za první semestr je požadováno:

- První dva body zadání

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Zbořil František, doc. Ing., Ph.D.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 11. května 2022
Datum schválení: 3. listopadu 2021

Abstrakt

Cieľom tejto práce je naštudovať a navrhnúť program, ktorý bude schopný rozoznávať základných 10 hudobných žánrov pomocou hlbokého učenia. Na klasifikáciu hudobných žánrov bola použitá navrhnutá konvolučná sieť, založená na frameworku Tensorflow. Táto sieť spracuje vstupný audio súbor po segmentoch vo forme spektogramov a na základe nájdených príznakov vráti percentuálnu pravdepodobnosť, s akou patrí nahrávka do jednotlivých žánrov.

Abstract

This bachelor thesis aims to study and design computer program, which will be able to recognise 10 essential music genres using deep learning. This classification was implemented by convolutional neural network based on Tensorflow framework. This network process audio file into segments in form of spectograms and returns percentual propability of record being classified into specific genre by features found in spectogram.

Kľúčové slová

Konvolučné neurónové siete, hudobný žáner, spektogram, segmentácia, GTZAN, TensorFlow, spracovanie obrazu

Keywords

Convolutional neural networks, music genre, spectogram, segmentation, GTZAN, TensorFlow, image processing

Citácia

PÁLENÍK, Radoslav. *Systém pro doporučování skladeb*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc.Ing. František Zbořil, Ph.D.

Systém pro doporučování skladeb

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána doc.Ing Zbořila, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Radoslav Páleník
11. mája 2022

Poďakovanie

Chcel by som poďakovať môjmu vedúcemu práce docentovi Františkovi Zbořilovi za jeho rady, ústretový a trpezlivý prístup a v neposlednom rade za vedenie tejto práce. Zároveň by som chcel poďakovať mojej rodine, ktorá ma podporovala nielen počas písania tejto práce, ale taktiež počas celého štúdia.

Obsah

Kapitola 1

Úvod

Ľudský život je prepojený s hudbou už od nepamäti. Je súčasťou nášho životného štýlu, prináša nám radosť, upokojuje nás, pomáha nám relaxovať, rozvíjať osobnosť, spájame si ju s konkrétnymi udalosťami, ale aj pocitmi. Pri pôvode hudby sa často diskutuje na tému, či nie je dokonca staršia ako naša samotná reč, pričom najstarší doteraz nájdený hudobný nástroj je viac ako 50 tisíc rokov starý.

Vo všeobecnosti môžeme hudbu definovať ako špecifickú ľudskú aktivitu usilujúcu o komunikáciu s pomocou v čase charakteristicky zoskupených tónov a zvukov využívajúcich rytmu, tempa a harmónií, čím sa snaží vyvolať určitý druh emócie u poslucháča.

Počas histórie sa hudba niekoľkokrát výrazne zmenila, vznikali nové hudobné nástroje, reflektovala náladu spoločnosti. No hudba ako ju dnes poznáme sa vyvinula len v posledných storočiach. Počiatky aktuálnej hudby síce siahajú až k ľudovej a klasickej hudbe, výrazný zlom modernej hudby však nastal na prelome 19. a 20. storočia, kedy vznikla tzv. populárna hudba - protichodný smer klasickej a ľudovej hudby určenej pre masy poslucháčov. S neskorším príchodom nových hudobných nástrojov sa jednotlivé smery hudobných umelcov začali výraznejšie líšiť, pričom sa začali kategorizovať na základe sebe podobných vlastností do určitých tried, ktoré dnes poznáme ako hudobné žánre.

A keďže hudba je veľmi subjektívne vnímaná forma prejavu, doniesli hudobné žánre so sebou aj vlnu rozporov. Práve subjektívne chápanie hudby môže vytvárať určité nejasnosti s tým, do ktorého z týchto žánrov daná skladba patrí. V poslednej dobe vzniká enormné množstvo nových žánrov odvodených od určitej skupiny nosných žánrov ako je napríklad rock, elektronická hudba, alebo hip-hop. Preto môže byť výber správneho, resp. dominantného žánru často pre poslucháča problematický alebo až nemožný. Momentálne je však možné tvrdiť, že žáner je definovaný použitými hudobnými technikami, nástrojmi, ale aj bližšie charakterizovaným kultúrnym a geografickým zasadením.

Cieľom tejto práce je návrh systému schopného rozoznať jednotlivé žánre a ich obsadenie vo vloženej audio nahrávke. Systém klasifikuje 10 žánrov uvedených v datasete GTZAN[20] pomocou modelu navrhnutej konvolučnej neurónovej siete. Vytváraný model je implementovaný v jazyku `Python` s pomocou knižnice `Tensorflow` a jej `API Keras`. Práca sa okrem návrhu neurónovej siete taktiež v kapitole 5 venuje analýze iných riešení z odboru získavania informácii o hudbe (Music Information Retrieval), ktoré používajú streamingové spoločnosti na predikciu individuálneho vkusu svojich zákazníkov a ponúknutie služby objavovania nových skladieb, ktoré by sa im mohli páčiť. Návrh modelu, ktorý by mohol byť schopný taktiež odporúčať skladby poslucháčovi na základe jeho preferencií je predstavený v kapitole 9.

Kapitola 2

Parametre hudby

Cieľom práce je návrh a implementácia neurónovej siete schopnej rozlíšiť najpopulárnejšie žánre. Takáto relatívne jednoduchá klasifikácia sa nejaví ako problém, ktorý by nebol človek schopný vyriešiť. Je síce možné polemizovať, že takáto klasifikácia je zo značnej časti založená na subjektívnom vnímaní každej osoby a teda je pravdepodobné, že vzniknú určité rozpory pri klasifikácii konkrétnych skladieb, no stále bude pri väčšine klasifikovaných ukážok figurovať zhoda. Takýto proces vnímania je značne založený na jednotlivých pocitoch, ktoré si osoba dokáže vybaviť pri počúvaní konkrétnych zvukov, alebo sekvencií tónov. Algoritmy strojového učenia si narozdiel od ľudí k hudbe nevyvíjajú žiadne citové väzby a ani nedokážu samostatne vnímať určité paradigmy, na základe ktorých si ľudia dokážu kategorizovať piesne do žánrov. Preto je potrebné jej modelu podať správne interpretované dáta, na základe ktorých sa bude pri rozhodovaní učiť, riadiť a zdokonaľovať. Navrhovaná neurónová sieť skúma poskytnuté hudobné ukážky ako zhluky dát, v ktorých je potrebné nájsť určité opakujúce sa vzory týkajúce sa najmä hudobnej teórie a technickej stránky súboru. Skúmanie je teda zamerané na technickú (hudobnú) časť diela, text poskytnutých piesní je teda pre model zanedbateľný a model s ním nepracuje. V tejto časti práce budú predstavené spôsoby zobrazovania zvukových signálov v časti kapitoly 2.1, následne spracovanie trénovacieho datasetu pre potreby práce v časti 2.2 .

2.1 Grafické zobrazenie zvukových súborov

Pokiaľ má byť práca schopná pracovať so zvukovým súborom v kontexte ľudského vnímania, je potrebné transformovať zvukový signál do vhodnej dátovej interpretácie. Pri interpretovaní takýchto dát je potrebné, aby bolo možné určovať špecifické vlastnosti na základe ktorých je možné spájať skúmané dáta vo vzťahu ku konkrétnemu hudobnému žánru.

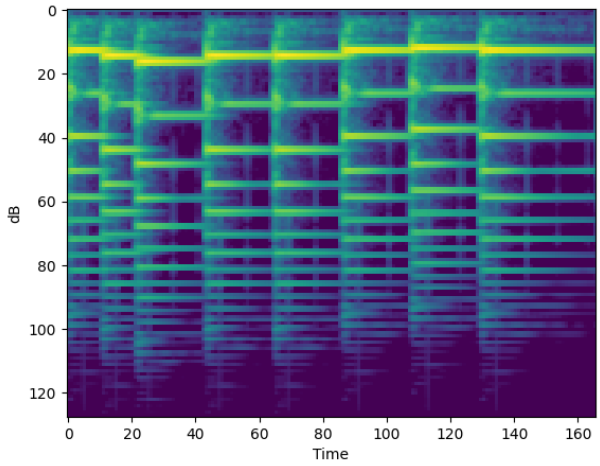
Pri hudobných ukážkach je možné pozorovať závislosť tempa (BPM) a najpoužívanejších frekvencií (pomocou heat-mapy v spektogramoch) v korelácii s hudobným žánrom. Okrem toho je možné pri značnej časti žánrov spozorovať rozdiel v komplexite diela, či už počtom použitých nástrojov(a teda aj šírkou využitého frekvenčného spektra), alebo aj použitými technikami pri hre na nástroje(napr. tremolo, vibrato). Taktiež je možné pozorovať koleráciu konkrétnych nástrojov len pri určitých žánroch (napr. banjo v country). Keďže cieľom práce je návrh modelu schopného využívať metódy počítačového videnia, konkrétne konvolučnú neurónovú sieť, je potrebné previesť zvukový súbor do analyzovateľného vizuálneho formátu.

2.1.1 Mel-spektogram

Pre potreby práce je audio dataset transformovaný do spektogramu *mel*. Jedná sa o spektogram v škále subjektívnej vnímanej výšky zvuku. Základnou jednotkou tejto škály je 1 *mel* - číselný ekvivalent frekvencie jedného tónu s intenzitou rovnou 40dB. [12] Túto škálu je teda možné konvertovať do frekvencie pomocou vzorca:

$$mel = 2595log_{10}(1 + \frac{f}{700})$$

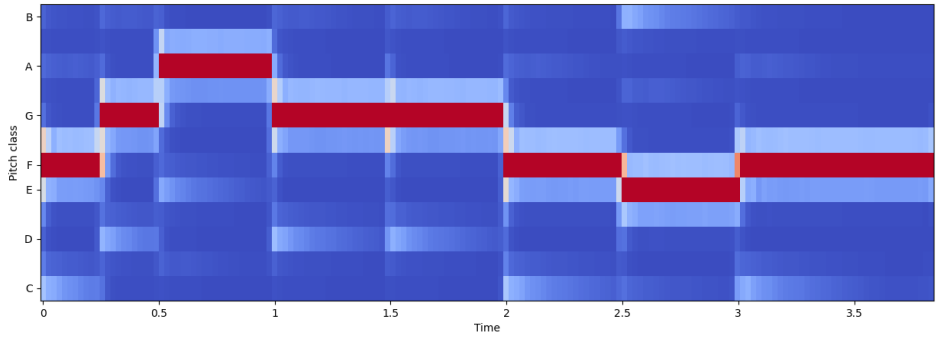
Vytváraný spektogram následne pozostáva z vyvedenia početnosti jednotiek *mel* pre každú analyzovanú vzorku obsiahnutú v audio nahrávke. Vo vygenerovaných spektogramoch z ukážok daného žánru je možné vyhľadávať určité vzory špecifické pre daný žáner. Na základe nájdených vzorov by mala výsledná neurónová sieť byť schopná detegovať, o aký žáner v skúmanej nahrávke sa jedná.



Obr. 2.1: Mel spektogram hry na piano

2.1.2 Chromagram

Okrem zobrazenia jednotlivých vnímaných frekvencií je možné audio dáta konvertovať aj do chromagramu zobrazujúceho intenzitu a priebeh zachytených tónov štandardnej stupnice v čase [24]. Jedná sa o spektogram upravený do formy zobrazovania jednotlivých tónov odvodený od základnej tónovej stupnice. Z takto zobrazených dát je možné získať harmonický postup(chord progression) [23], harmonický hudobný základ, na ktorom je založená moderná hudba západného sveta od čias klasiky po dnešné žánre.



Obr. 2.2: Chromagram rovnakej hry na piano

2.2 Vstupné dáta

Práca využíva dataset GTZAN obsahujúci 1000 ukážok piesní rovnomerne rozdelených do 10 hudobných žánrov. Výber datasetu na takýto typ práce je značne obmedzený hlavne autorskými právami na jednotlivé skladby. Z existujúcich datasetov bol GTZAN najlepšie využiteľný, keďže je členený iba do 10 nosných tried(žánrov) a všetky obsiahnuté skladby sú štandardizované na rovnakú dĺžku. Dá sa teda tvrdiť, že GTZAN je audio paralelou k populárnemu datatsetu MNIST, ktorý sa využíva hlavne na trénovanie rozpoznávania číslic konvolučnými neurónovými sieťami. Na tvorbu datasetu boli použité nahrávky z rôznych zdrojov, ako napríklad skladby z CD, alebo nahrávky rádia s cieľom reprezentácie rôznych podmienok nahrávacieho procesu. Žánre obsiahnuté v datasete je možné kvalifikovať nasledovne:

- **Blues** Vokálno-inštrumentálna hudba afroamerických komunít, ktorá vznikla v 60. rokoch 19. storočia v USA v okolí regiónu Deep South. Vychádza z pracovných piesní afro-amerických komunít, spirituálov a iných náboženských piesní. Jedná sa o zakladajúci žáner západnej modernej hudby. Formou sa jedná o sólový hlas doprevádzaný hrou na harmoniku a gitaru, počas spevu je charakteristické používanie modelu zvolanie-odpoveď [25].
- **Klasická hudba** Všeobecne referuje na formálnu hudobnú tradíciu západného sveta, ktorá sa považuje za odlišnú voči západnej populárnej alebo ľudovej hudby. Okrem svojej formálnosti sa vyznačuje zložitosťou svojej formy a harmonickou organizáciou, najmä s použitím polyfónie(mnohohlas s rytmickou samostatnosťou).Typickými sú pre ňu sláčikové nástroje, klavír alebo organ. Rozdeľuje sa na mnoho žánrov ako napríklad sonáta, symfónia alebo opera [26].

- **Country** Vzniklo v 20. rokoch minulého storočia na juhu a juhozápade USA. Vychádza z blues, kostolnej hudby a je výrazne ovplyvnená hudbou európskych prisťahovalcov. Typickými nástrojmi sú banjo, husle, kontrabas, gitara, alebo bicie [27].

- **Disco** Druh tanečného hudobného žánru kombinujúci prvky soulu, funku a popu. Je charakteristická svojim *Four-on-the-floor* rytmickým vzorom. Jedná sa o stály, rovnomerný úder pedálom na basový bubon v 4/4 takte, z ktorého vychádza jeho pomenovanie. Okrem bicích sú pre žáner typickými nástrojmi aj basgitara, syntetizér alebo elektrická gitara. Žáner vznikol na prelome 60. a 70. rokov minulého storočia na scéne nočného života vo Philadelphii a New Yorku [28].

- **Hip-hop** Mestská hudobná štruktúra vznikajúca hlavne na prelome 70. a 80. rokov minulého storočia v africko-amerických a latinsko-amerických štvrtiach v USA. Jedná sa o spojenie jamajských a amerických hudobných žánrov ako napr. reggae, R&B alebo aj disca. Je špecifický pre svoj hlasový prejav(rap), beatbox, deejaying a syntetizovanie hudby [31].

- **Jazz** Začal vznikať začiatkom minulého storočia v meste New Orleans v USA. Žáner vznikol spojením blues a iných afroamerických piesní v kombinácii s európskymi tradíciami. Štýl sa vyznačuje silnou improvizáciou, používaním tónov z blues, alebo menením melorytmických pásiem. Typickými nástrojmi sú dychové nástroje ako saxofón, trúbka, trombón, ale aj klavír, bicie a gitara[1] [29].

- **Metal** Vznikol na prelome 60. a 70. rokoch minulého storočia v Spojenom Kráľovstve a USA. Korene žánru vychádzajú z hardrockovej hudby (rock kombinovaný s blues). Zvuk je postavený na skreslených tónoch elektrických gitár, hutnom zvuku bicích a živelných vokáloch, rovnako ako pri rocku je prítomná hra na basgitaru a taktiež je možné zakomponovanie klavíru alebo iných inštrumentov [30].

- **Pop** Začal vznikať ako žáner počas 50. rokov minulého storočia v Spojenom kráľovstve a USA. Je často zamieňaný za populárnu hudbu(pojem označujúci širokú škálu moderných hudobných žánrov). V dobe svojho vzniku zahŕňal pop rokenrolovú hudbu a iné žánre orientované na mládež. Až do konca 60. rokov sa používal pojem pop v synonymickom význame s pojmom rock pokiaľ, sa nezačali od seba čoraz viac vzďaľovať. Pop je často vnímaný ako hudba hitparád a často si vypožičiava prvky iných hudobných žánrov, napr. elektronickej hudby, rocku, alebo latina [33].

- **Reggae** Vzniklo na Jamajke počas 60. rokov minulého storočia. Je charakteristické prízvukom na neprízvučné doby(2. a 4. doba taktu) so špecifickou hrou na gitaru zdôrazňujúcu tretiu dobu. Štylisticky kombinuje elementy R&B, jazzu, jamajskú ľudovú hudbu a iné žánre [34].

- **Rock** Prvýkrát sa objavil koncom 50. rokov minulého storočia v USA. Zvuk je tradične postavený na zosilnených elektrických gitarách, sekundovaný elektrickou basgitarou a na bicích. Je tradične postavený na jednoduchých rytmoch špecifických prízvukom na ťažkej dobe a údermi do bicích v 2. a 4. dobe. Je silne ovplyvnený blues, jazzom, ale taktiež klasickou a ľudovou hudbou (súčasťou rockových kapiel býva často aj klavirista) [35].

[1] Trieda Jazz obsahuje chybnú ukážku č.54, opravená nahrávka je dostupná v odpovedi na: https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification/discussion/158649

2.2.1 Formát vstupných dát

Po stiahnutí je archív datasetu zložený zo 4 častí:

- **genres_original** - Kolekcia 1000 ukážok rovnomerne rozdelených do 10 žánrov v adresárovej štruktúre podľa prideleného žánru
- **images_original** - Predpripravená kolekcia mel-spektogramov vytvorených z genres_original, má rovnakú adresárovú štruktúru
- **features_30-_sec.csv** - CSV súbor obsahujúci výpis stredných a priemerných hodnôt rôznych extrahovaných zvukových čŕt celých, 30 sekundových, nahrávok
- **features_3-_sec.csv** - CSV súbor obsahujúci rovnaké parametre segmentovaných originálnych nahrávok po 3 sekundových intervaloch

Z datasetu je využívaná iba kolekcia nahrávok, keďže ostatné dáta nie sú pre navrhovanú neurónovú sieť použiteľné (nespĺňajú požadovanú formu, alebo obsahujú pre učenie irelevantné dáta). Každá ukážka z datasetu je dlhá presne 30 sekúnd, pričom sa jedná o jednokanálovú (mono) 16-bitovú nahrávku so vzorkovacou frekvenciou 22050Hz. Ukážky v datasete sú uložené v súborovom formáte `Waveform Audio File Format` (skrátene `WAV`). Na rozdiel od najpoužívanejšieho audio formátu `MPEG-1` (resp. `MPEG-2`) `Audio Layer III` (skrátene `MP3`) je WAV bezstratový, čo mu umožňuje zachytiť celé neupravené zvukové spektrum (MP3 je okrem kompresie orezaný pri hranici 20kHz), čím je možné poskytnúť pri trénovaní modelu lepšiu detekciu príznakov. Z každej ukážky sa vytvára 6 spektogramov vedených v stupniciach `mel` o dĺžke 5 sekúnd, čo predstavuje dĺžku ktorá by mala postačovať na rozoznanie jedného zo žánrov. Každý jeden takto spracovaný spektogram predstavuje 2D tenzor. Dáta ktoré takýto tenzor obsahuje sú vo formáte

$$\begin{vmatrix} a_{x,y} & a_{x+1,y} & \dots & a_{x+m,y} \\ a_{x,y+1} & a_{x+1,y+1} & \dots & a_{x+m,y+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{x,y+n} & a_{x+1,y+n} & \dots & a_{x+m,y+n} \end{vmatrix}$$

kde:

- **a** predstavuje usporiadanú množinu(r, g, b, a^2) kanálov pre daný bod, $r, g, b \in < 0, 255 >$, $r, g, b \in \mathbb{Z}^+$
- **x, y** predstavujú súradnice bodu v spektograme $x \in < 0, m), y \in < 0, n)$
- **m, n** predstavujú rozmery vytvoreného spektrogramu (pre 5 sekundový segment sa vytvorí obrázok o veľkosti 496*293 pixelov)

2.3 Alternatívne datasety

Okrem popísaného datasetu GTZAN existuje len malý počet datasetov určených pre oblasť získavania informácií zo zvuku. Z existujúcich datasetov nájdených počas písania práce bola väčšina tvorená len ako jeden alebo viac tabuľkových súborov (prevažne formátu CSV). Obsah týchto súborov zvyčajne tvoril zoznam extrahovaných príznakov podobný zoznamom

[2] Alfa kanál udávajúci priehľadnost bodu

uvedených na začiatku kapitoly 2.2.1. Po preskúmaní existujúcich riešení boli na výber 2 datasety GTZAN a FMA (Free Music Archive) [7]. Dataset FMA je síce narozdiel od GTZAN značne rozsiahly (v plnej veľkosti má vyše 105 tisíc ukážok spolu o veľkosti až 879 GiB), jeho nevýhodami však sú kompresný formát MP3, nerovnomerné rozloženie žánrov a priveľa špecifických podžánrov, vytvárajúcich rodičovskú stromovú štruktúru. Spomínaný dataset má aj svoju malú verziu s 8000 skladbami skrátenými na 30 sekúnd (podobne ako GTZAN), rozdelenými do 8 žánrov, no nie je vhodne štrukturovaný a kvôli veľkej veľkosti CSV súborov sa aj s najmenším z ponúkaných 4 veľkostí archívov z užívateľskej stránky pohľadu zle pracuje. Aj na základe týchto skutočností bol pre účely práce vybraný na trénovanie neurónovej siete dataset GTZAN.

Kapitola 3

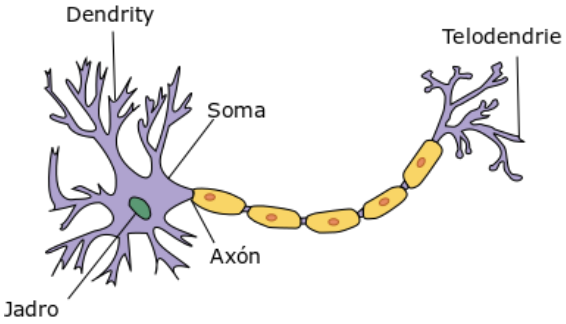
Neurón a neurónová sieť

Hlboké učenie je momentálne jednou z najpopulárnejších metód riešenia problémov v rámci odboru umelej inteligencie. Ako samostatný odbor umelej inteligencie siaha jeho história do roku 1950, no pozornosť si začal získavať až v posledných rokoch s príchodom neurónových sietí. V tejto kapitole je popísaný model biologického neurónu spolu s jeho abstrakciou pre účely strojového učenia a parametrami, ktoré ovplyvňujú ich rozhodovanie.

3.1 Biologický neurón

Neurón je základnou jednotkou periférnej nervovej sústavy. Jeho úlohou je prenášať informácie z celého tela a ovládať jednotlivé svaly na základe signálov z centrálnej nervovej sústavy, ktorú riadi mozog spolu s miechou. Tieto informácie sa prenášajú ku konkrétnym zakončeniam cez komunikačné siete tvorené neurónmi, ktoré sú rozšírené po celom ľudskom tele, pričom len ľudský mozog dospelého jedinca má približne 10^{10} neurónov. Na to, aby bol neurón schopný komunikovať so svojím okolím potrebuje byť zložený z týchto častí:

- **Telo (Soma)** obsahuje bunečné jadro, spracováva a riadi transfer signálov pripojených cez synapsiu.
- **Dentrit** Vstupný výbežok neurónu prijímajúci signál iných neurónov cez synapsiu.
- **Axón** Výstupný výbežok neurónu zakončený do synapsie pomocou vetvičiek označovaných ako telodendrie
- **Synapsia** Rozhranie, ktoré vytvára synaptickú väzbu najčastejšie medzi axónom a dentritom dvoch rozdielnych neurónov(môže prepájať rôzne kombinácie medzi axónmi, dentritmi a somami). Táto väzba môže byť excitačná (povzbudzujúca nejakú činnosť, napr. stiahnutie svalu) alebo inhibičná(potlačujúca). Dospelý jedinec disponuje $10^{14} - 5x10^{14}$ synapsiami.[32][13]



Obr. 3.1: Zloženie neurónu. Zdroj: [32]

3.2 Umelý neurón

Model prvého navrhnutého umelého neurónu predstavili Warren McCulloch a Walter Pitts v roku 1943. Návrh je možné rozdeliť do dvoch častí, rozhodovaciu a vstupnú časť. Pôvodný návrh pracoval na princípe prahovej logiky. Vstupná časť takéhoto modelu agreguje všetky excitačné(kladné) a inhibičné (záporné) vstupy vo vstupnej časti. Výsledkom agregácie je rozdiel súčtu kladných(E) a súčtu záporných(I) vstupov. Výstup rozhodovacej časti a teda celého neurónu je závislý na prekročení prahového parametru θ, ktorého stav je určený z výsledku agregácie vstupov vypočítanej vo vstupnej časti [4] [37].

$$g(x) = \Sigma E - \Sigma I$$

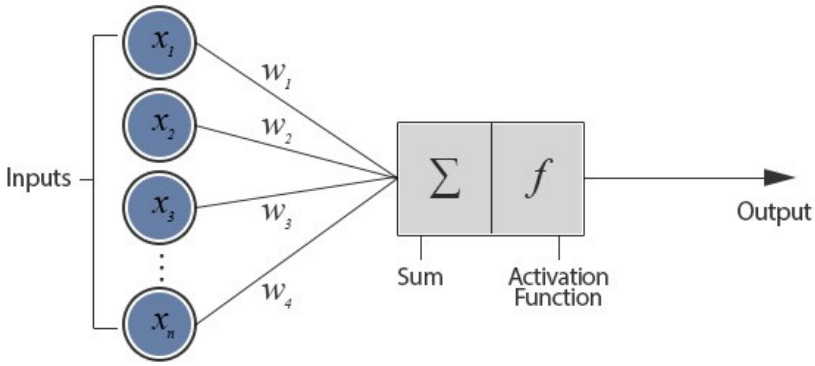
$$y = \begin{cases} 0 & \text{pre } g(x) < \theta \\ 1 & \text{pre } g(x) \geq \theta \end{cases}$$

Obr. 3.2: Prahová logika neurónu. Zdroj:[37]

Súčasný model umelého neurónu pracuje s množinou vstupov x a množinou im pridelených váh w. Umelý neurón simuluje silu synaptických väzieb pridaním váhy w_i pre každý vstup x_i. Tým dokáže definovať vplyv jednotlivých vstupov na konečný výstup neurónu. Vstupná časť tieto vážené hodnoty vstupov agreguje pomocou vybranej bázovej funkcie f, ktorá slúži na výpočet výstupného potenciálu neurónu u. Tento potenciál je ďalej predávaný do aktivačnej funkcie g, ktorá následne vyhodnocuje výstup neurónu výpočtom definovaným jej typom. Do výpočtu bázovej funkcie je taktiež možné pridať prah b ktorý predstavuje zápornú hodnotu prahového parametru θ. V práci sú využívané lineárne bázové funkcie, ktorých aktivačné funkcie sú rozoberané ďalej v kapitole 3.5.

$$y = g(u) = g(f(\overrightarrow{x}, \overrightarrow{w}))$$

Obr. 3.3: Výstupná funkcia neurónu. Zdroj:[37]



Obr. 3.4: Model umelého neurónu. Zdroj:[14]

3.2.1 Perceptrón

Perceptrón, najjednoduchšia forma neurónovej siete bol navrhnutý v roku 1958 Frankom Rosenblattom. Vychádza z pôvodného návrhu umelého neurónu McCullocha a Pittsa. Je tvorený jedným neurónom a používa lineárnu bázovú funkciu (LBF), ktorá umožňuje klasifikovanie vstupu do dvoch tried (hypotéz). Vnútorný potenciál perceptrónu je daný skalárnym súčinom $\vec{x} \cdot \vec{w}$. Vzhľadom na jeho stavbu je pomocou perceptrónu možné klasifikovať iba vzory, ktoré sú lineárne odlíšiteľné [37].

3.3 Zloženie neurónovej siete

Neurónová sieť je výpočtový model zložený z vrstiev, ktorých vstupom a aj výstupom sú dátové polia zložené z tenzorov. Pojmom tenzor rozumieme reprezentáciu multidimenzionálneho matematického objektu, ktorý udržuje vzťahy vo vektorovom priestore. Jedná sa o objekt, ktorý v sebe dokáže uchovávať skaláry, vektory, ale aj iné tenzory. Takýto objekt je abstrakciou informácii podávaných biologickým neurónom nachádzajúcom sa v ľudskej nervovej sústave. Pokiaľ chceme aby navrhovaná sieť vedela lepšie porozumieť ako zaobchádzať s vloženými dátami, je potrebné aby bola takáto sieť zložená z viacerých vrstiev, ktoré si medzi sebou predávajú pôvodné dáta upravené aktivačnou funkciou každého umelého neurónu, cez ktorý boli spracovávané. Každá vrstva takejto siete môže byť zložená z viacerých neurónov v závislosti od predpokladaného formátu vstupných dát pre vrstvu, pričom prvá (vstupná) vrstva je zložená toľkými neurónmi, aby pokryla každý jeden samostaný tenzor uchovávajúci časť zo vstupných dát. Počet neurónov v nasledujúcich, skrytých vrstvách sa môže líšiť v závislosti od typu použitých vrstiev. V predchádzajúcej kapitole 3.2.1 bola predstavená jednovrstvová neurónová sieť v podobe perceptrónu. V praxi sa ale oveľa viac používajú viacvrstvové siete. Ako už ich názov napovedá, jedná sa model siete, ktorý je tvorený niekoľkými za sebou spájanými vrstvami, ktoré sú medzi sebou prepojené, čo umožňuje predávanie si informácií medzi vrstvami.

Takýto model siete využíva definovaný počet vrstiev rozdelených do troch kategórií:

- **Vstupná vrstva** Prijíma vstupné dáta (v prípade tejto práce vytvorené spektogramy) a spracováva ich ako tenzory, ktoré posúva skrytým vrstvám. Vstupnou vrstvou môže byť aj vrstva z kategórie skrytých. Môže, ale nemusí mať zadefinovaný tvar vstupných dát.

- **Skryté vrstvy** Model neurónovej siete si pri trénovaní v skrytých vrstvách vytvára mapu príznakov, ktoré pomáhajú sieti generalizovať spoločné vzory pre danú triedu dát. Na základe nájdených príznakov je sieť neskôr schopná určiť, či požadovaný vstup spadá do jednej z určených kategórií. Pri rozhodovaní využíva každý neurón vo vrstve svoju aktivačnú funkciu, na základe ktorej sa rozhoduje o jeho výstupe. Počas prechádzania vrstvami upravuje model váhy jednotlivých neurónov. Nastavovaním váh sa určuje vplyv jednotlivých neurónov na výstup siete, pričom sieť hľadá ich optimálne vyváženie. Takýto proces nazývame učením neurónovej siete.
- **Výstupná vrstva** Výstupom poslednej vrstvy je najčastejšie pravdepodobnosť, s akou vložené dáta spadajú do určitej skupiny. Tvar výstupných dát je závislý na aktivačnej funkcii výstupnej vrstvy.

3.4 Optimalizácia pomocou gradientu

Pokiaľ má byť navrhovaný model neurónovej siete schopný správne klasifikovať vstupné dáta, musí vedieť rozoznať relevantné informácie z poskytnutých vstupných dát. Ako bolo spomenuté v kapitole 3.2, pre každý vstup je v neuróne pridelená váha, ktorou sa určuje jeho vplyv na okolie a teda aj na celkový výsledok klasifikácie. Hodnoty týchto váh sú zo začiatku náhodne inicializované na malé hodnoty, ktoré zo začiatku síce nedávajú zmysel, ale udávajú východiskový bod od, ktorého sa počas učenia vie model odraziť a následne ich tak môže začať postupne korigovať na optimálne hodnoty pomocou algoritmu spätnej propagácie chyby.[6]

Aby model vedel, či jeho učenie postupuje správnym smerom je potrebné mať metriku, ktorou sa bude pri nastavovaní váh neurónov riadiť. Túto metriku predstavuje stratová funkcia (loss function). Tú využívajú neurónové siete na meranie vzdialenosti medzi očakávaným a získaným výstupom (odhadom) modelu, čím hodnotia, ako dobre pracuje model s poskytnutými vstupnými dátami.

Nastavovanie váh je teda zložitý proces, a aby model nemusel počas učenia zmrazovať všetky váhy až na jednu, ktorej vplyv sa snaží aktuálne správne určiť, je potrebné aby všetky operácie používané v sieti boli diferenciovateľné. Pokiaľ spĺňa model túto podmienku, vie model získať optimálne hodnoty váh analyticky, a to tak že sa modelu umožní výpočítať gradient strát s ohľadom na parametre trénovania.

Gradientom rozumieme deriváciu funkcie s viacerými vstupnými premennými tvoriacimi vektor hodnôt, v prípade neurónových sietí sa jedná o deriváciu tenzorovej operácie pracujúcej s maticou váh, vstupným vektorom a požadovaným cieľom. S pomocou gradientu vieme optimalizovať sieť s cieľom minimalizovania stratovej funkcie pre daný model, napríklad pomocou gradientného zostupu. Gradient je teda možné chápať ako vyjadrenie zmeny hodnoty stratovej funkcie pri zmene nastavenia vnútorných váh jednotlivých neurónov v sieti.[6]

3.4.1 Algoritmus spätnej propagácie chyby (backpropagation)

Jedná sa o najznámejší a najpoužívanejší algoritmus pri nastavovaní synaptických váh siete. Jeho cieľom je minimalizovať výsledok stratovej funkcie, čo robí počítaním jej gradientu a jeho upravovaním s dôrazom na každú jednotlivú váhu použitú aktivačnými funkciami. Na zníženie hodnoty stratovej funkcie je potrebné posunúť váhy neurónov smerom klesajúceho gradientu. Ako už bolo spomenuté, pre umožnenie tejto operácie je potrebné aby boli všetky operácie v sieti diferenciovateľné. Backpropagation algoritmus sa pri výpočte gradientu riadi

reťazovým pravidlom, tzn. že počíta deriváciu gradientu vždy len na jednej vrstve, pričom postupuje výpočtami modelom od poslednej vrstvy k prvej.[6]

3.4.2 Gradientný zostup

Cieľom postupu gradientného zostupu je minimalizovanie stratovej funkcie modelu. Pokiaľ je funkcia diferenciovateľná, je teoreticky možné nájsť všetky jej minimá analyticky. Minimum funkcie je bod, v ktorom je derivácia hodnoty parametru nulová. Týchto bodov môže existovať viac, preto je pre optimálny výkon siete potrebné určiť jej globálne minimum. Globálnym minimom funkcie sa rozumie ten bod, ktorý pochádza z množiny bodov funkcie s nulovou deriváciou a zároveň je hodnota tejto funkcie z bodov množiny najnižšia. Optimalizácii zostupu sa venujú algoritmy ako napr.*SGD, Adagrad, RMSProp*, ktoré sa nazývajú optimalizátory a sú volané pri kompilácii modelu. Pri optimalizácii môžu nastať problémy s nájdením globálneho minima. Ich odstránenie dokáže riešiť optimalizátor pomocou:

- **Veľkosti kroku (learning rate)** Od ich veľkosti závisí rýchlosť a presnosť učenia, pokiaľ je krok príliš malý, je presnejší a pomalý, pričom pri veľkých (rýchlych) krokoch je možné prehliadnuť minimum.
- **Stratovej funkcie** Slúži na optimalizáciu veľkosti kroku, sleduje chybu predikcie od reálnej hodnoty parametru v danom bode a na základe rozdielu upravuje veľkosť kroku.
- **Hybnosti** Pri optimálnej veľkosti krokov môže optimalizátor uviaznuť v lokálnom minime. Hybnosť čerpá inšpiráciu z fyziky. Poslednú hodnotu optimalizátoru si je možné predstaviť ako malú guličku položenú na krivke, ktorá je schopná pri dostatočnej hybnosti vyliezť z aktuálneho minima a pokračovať v optimalizácii [6].

3.5 Aktivačná funkcia

Pri jednotlivých vrstvách siete je možné vybrať aktivačnú funkciu, ktorou je prechádzaný jej každý vstupný uzol, podobne ako pri modeloch uvedených v 3.2. Takáto funkcia slúži na vyhodnotenie výstupu pre daný uzol, ktoré je použité ako vstup pre ďalšiu vrstvu(určuje, či má byť uzol „aktivovaný“). Ich cieľom je odstrániť nelinearitu výstupných neurónov pre získanie komplexnejších informácií zo zdroja, a teda zmeniť resp. nakloniť udeľovanie váhových hodnôt pre daný neurón smerom, ktorým to daný model potrebuje.

- **ReLU (Rectified Linear Unit)** Jedná sa o momentálne najpopulárnejšiu aktivačnú funkciu, ktorej výstupom je obor hodnôt $(0, \infty)$. Hlavnou výhodou ReLU je, že automaticky neaktivuje všetky neuróny, ale len tie, ktoré spĺňajú podmienku z funkcie 3.5, kde x predstavuje vstupnú hodnotu vloženú na porovnávanie. To radí túto funkciu medzi vysoko efektívne a rýchle. Pri spätnom šírení môže ale nastať jav, kedy sa váhy pre určité neuróny (v predchádzajúcich epochách radené medzi deaktivované) neaktualizujú a vznikajú tzv. „mŕtve neuróny“. Ako riešenie na tento problém sa používa upravená funkcia Leaky ReLU.

$$y = max(0.0, u)$$

Obr. 3.5: Funkcia ReLU

- **Sigmoid** je funkcia s oborom hodnôt $(0, 1)$. Funkcia sigmoidu má negatívum, pre ktoré sa v poslednej dobe používa čoraz menej. Je ním problém miznúceho gradientu. Funkcia saturuje a zabíja gradient, čím spomaľuje učenie predchádzajúcich vrstiev pomocou algoritmu spätného šírenia. Predchádzajúce vrstvy sú vždy schopné zachytiť a rozoznať výraznejšie príznaky neurónov a ich spomalením sa výrazne spomaľuje celkový priebeh učenia.

$$y = \frac{1}{1+e^{-u}}$$

Obr. 3.6: Funkcia sigmoid

- **Hyberbolický tangens (Tanh)** je taktiež sigmoidnou funkciou, pracuje ale na obore hodnôt $(-1, 1)$. Oproti klasickému sigmoidu sú dáta rozložené cez bod $[0, 0]$, čím sa zaručí silnejší gradient a funkcia sa dokáže vyhnúť určitej predpojatosti(bias). Výhodou tejto funkcie pre prácu predstavuje jej obor hodôt $(-1, 1)$, ktorý je možné použiť pri práci s váhami počas zisťovania hudobných preferencií poslucháča.

$$y = tanh(u)$$

Obr. 3.7: Funkcia Tanh

- **Softmax** Funkcia normalizuje výstupné hodnoty všetkých vnútorných potenciálov m výstupných neurónov tak, aby ich výsledný súčet bol 1. V súčastnosti je táto funkcia veľmi používaná hlavne na riešenie klasifikačných problémov.

$$y_j = \frac{e^{u_j}}{\sum_{k=1}^{m} e^{u_k}}$$

Obr. 3.8: Funkcia Softmax

Kapitola 4

Typy viacvrstvových neurónových sietí

Po predstavení modelu samotného neurónu sa pred vedcov postavila nová, ešte náročnejšia úloha. Bol vytvorený základný stavebný blok, ktorý bolo potrebné vedieť správne prepojiť s ďalšími jemu rovnocennými blokmi za cieľom abstrakcie vnímania spojitostí. Podobne ako pri živom organizme sa začali vytvárať umelé nervové sústavy, ktoré poznáme pod pojmom viacvrstvové neurónové siete. Pomocou nich vieme abstrahovať nervový systém vytváraním vrstiev neurónov, ktoré medzi sebou komunikujú na základe presne zadefinovaných vzťahov.

4.1 Viacvrstvový perceptrón

Za prvý model viacvrstvovej siete je možné považovať `viacvrstvový perceptrón (MLP)`, ktorý pozostával z minimálne troch vrstiev: vstupnej, aktivačnej a výstupnej(pri väčších modeloch sa navyšuje počet aktivačných vrstiev). Pre každú vrstvu(okrem vstupnej) platí, že sa skladá z neurónov disponujúcich nelineárnou aktivačnou funkciou, vďaka čomu sa spolu v kombinácii s viacerými vrstvami odlišuje od klasického (lineárneho) perceptrónu, čo mu umožňuje rozlíšiť dáta, ktoré nie sú lineárne oddeliteľné. Jedná sa o sieť dizajnovanú predovšetkým na použitie s dvojrozmernými vstupnými dátami.

4.2 Konvolučná neurónová sieť

Konvolučné neurónové siete (CNN) sú inšpirované vizuálnym vnímaním živých organizmov. Takáto sieť si na vrstvách vytvára rôzne stupne abstrakcie vstupných, zvyčajne obrazových dát. Pri prehliadaní dát začína sieť hľadať zo začiatku najviditeľnejšie detaily s postupným prechodom do hlbších vrstiev modelu, kde sú vyhľadávané čoraz komplexnejšie koncepty. CNN si neukladá postup prehliadania do pamäte a každý vstup je spracovávaný samostatne, preto musia byť vstupné dáta vždy predspracované na štandardizovaný formát [6]. Tento typ sietí sa používa predovšetkým na rozpoznávanie a klasifikáciu obrázkov a klasifikáciu textu.

Pri implementovaní konvolučných neurónových sietí sa využívajú najmä vrstvy s týmito operáciami:

- Konvolúcia - Násobenie novej hodnoty nastavenými váhami pre každý bod pomocou filtrov

- Nelineárna aktivačná funkcia (napr. ReLU) - Zakončuje konvolučné vrstvy
- Pooling - Redukcia skúmaného priestoru
- Klasifikácia - Typicky reprezentovaná plne prepojenou vrstvou (Dense)

4.2.1 Vrstvy konvolučnej neurónovej siete

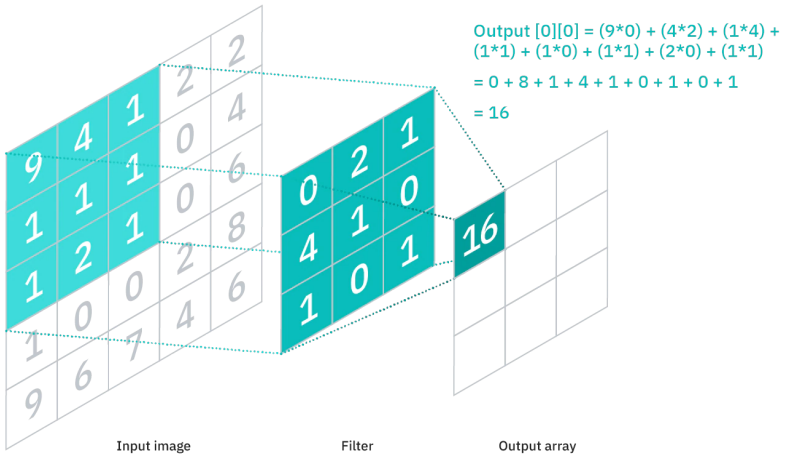
Konvolučné vrstvy

- Pri učení boli na vytvorenie príznakov použité konvolučné vrstvy. Tieto vrstvy sú schopné naučiť sa v dátach rozpoznať určité opakujúce sa lokálne príznaky(anglicky features), z ktorých vytvárajú ich mapy(feature maps). Keďže tieto mapy lokálnych príznakov sú spravidla rozmerovo menšie ako je rozmer pozorovanej dátovej jednotky, je možné neskôr takto naučený príznak rozpoznať na rôznych miestach po jej celej ploche - naučené vzory sú teda invariantné.[6]

Tieto vrstvy pracujú na princípe **konvolúcie**. V ponímaní neurónových sietí predstavuje konvolúcia lineárnu operáciu, ktorá násobí úsek matice vstupu (najčastejšie obrázok) s maticou váh (abstrakcia recepčného poľa neurónu, často nazývaná aj filter alebo kernel). Táto operácia násobenia je interpretovaná ako skalárny súčin, z čoho vychádza, že výsledkom operácie je vždy jedna hodnota. Samotný filter má menšie rozmery (spravidla veľkosť pár jednotiek v každom rozmere, napr. 3x3) ako spracovávaný obrázok, čo umožňuje opakované vyžitie filtra na prenásobenie všetkých hodnôt vloženého obrázku. Konvolučné vrstvy teda pri prechode opakovane prekladajú pôvodný obrázok maticou filtra podľa veľkosti nastaveného kroku, pričom prechádzajú celý obrázok zľava-doprava a zvrchu-dolu. Pokiaľ je filter dobre nastavený, je schopný zachytiť konkrétny príznak hocikde vrámci skúmaného obrázku. Takáto schopnosť zachytenia filtrom sa nazýva aj prekladová symetria (translation invariance) [3].

Pri každej konvolučnej vrstve sa dá okrem veľkosti recepčného poľa nastaviť, aké parametre budú využívať jej konvolučné filtre. Veľkosťou kroku (anglicky stride) rozumieme celočíselnú hodnotu predstavujú o aký počet pozícií v sa v matici vstupných dát(v prípade tejto práce počet pixelov) posunie konvolučný filter jedným smerom. S krokom súvisí aj parameter „vypchávky“. Predpokladajme, že vrstva siete pracuje s konvolučnými filtrami o veľkosti 3x3. V takom prípade sa po prechode konvolučným filtrom zmenší prehliadaná plocha $n \times n$ prvkov na veľkosť $n-2 \times n-2$. Toto zmenšenie je spôsobené tým, že sa konvolučný filter nemá ako dostať k hraničným bodom vstupného poľa. Pokiaľ chceme zachovať rovnakú veľkosť vstupných dát aj pre ďalšiu vrstvu a teda aby bol konvolučný parameter vypočítaný aj pre krajné oblasti, je potrebné pre filter nastaviť vypchávku, tzv. padding. Padding pridáva okolo vstupu vhodný počet riadkov a stĺpcov do mapy príznakov, aby bolo možné umiestniť pôvodné okraje mapy príznakov ako stredový bod pre každý z filtrov pracujúcich na danej vrstve [6].

Pre každú vrstvu je možné zadefinovať počet filtrov, pri obrázkoch sa nastavuje počet aspoň na 8, aby bola schopná prekryť celé farebné spektrum na jednej vrstve obrázku. Tento počet sa medzi vrstvami nemusí zhodovať, no pri prechode hlbších vrstiev sa zvyšuje komplexita jednotlivých vzorov a teda je vhodné zvýšiť ich počet na vylepšenie presnosti detekcie malých detailov, čím je možné zvýšiť celkovú upešnosť správnej klasifikácie siete.



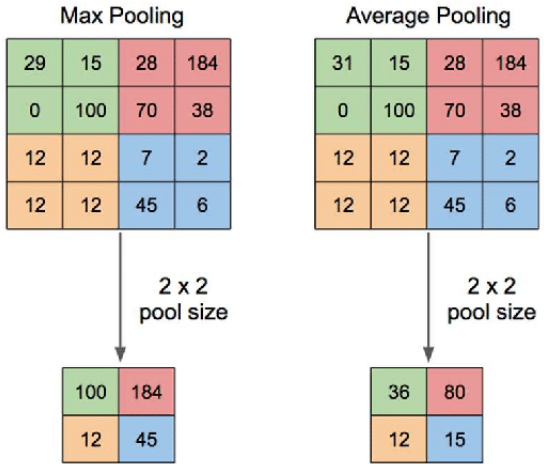
Obr. 4.1: Operácia konvolúcie. Zdroj: [8]

Poolingové vrstvy

Poolingové (združovacie) vrstvy sa pri učení neurónovej siete využívajú na zredukovanie vstupného poľa a teda na zrýchlenie výpočtov a lepšie zviditeľnenie príznakov. Takýto typ vrstvy zvyčajne nasleduje za konvolučnou vrstvou, pričom sa zvykne takáto postupnosť niekoľkokrát opakovať na zvýšenie efektivity modelu. Často sa pri nich používa filter o rozmere 2x2 a veľkosť kroku 2, čím sa veľkosť dvojrozmerného vstupu redukuje z $n \times n$ na $n/2 \times n/2$.

Výber poolingovej vrstvy záleží na funkcii, ktorá má byť použitá. Najčastejšie používané ale sú:

- **Max Pooling** Vyberá z pozorovanej oblasti najvyššiu hodnotu, teda najsilnejší príznak.
- **Average Pooling** Vytvára priemer všetkých príznakov v danej pozorovanej oblasti.



Obr. 4.2: Poolingové operácie. Zdroj: [36]

Plne prepojené vrstvy

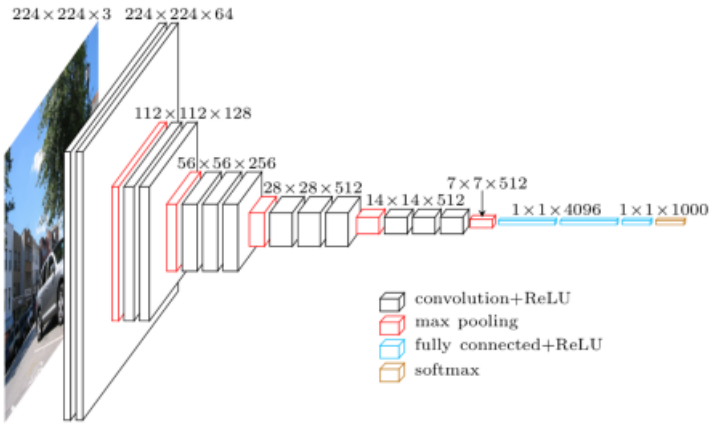
Jedná sa o najbežnejšiu vrstvu používanú pri učení. Každý jeden neurón tejto vrstvy je prepojený so všetkými neurónmi predchádzajúcej vrstvy. Dense vrstvy vykonávajú operáciu násobenia(dot) matíc vstupných dát(input) s maticou váh vytvorených vrstvou (kernel), pričom je možné tieto váhy siete natrénovať pomocou backpropagation algoritmu. Takéto vrstvy sú výpočtovo náročné, preto sa zvyčajne používajú len ako posledné vrstvy modelu siete . Pokiaľ sa jedná o výstupnú vrstvu, tak sa používa najmä aktivačná funkcia softmax [9][18].

$$output = activation(dot(input, kernel) + bias)$$

Obr. 4.3: Operácia Dense

4.2.2 Model VGG16

Model konvolučnej siete VGG16 bol vytvorený na oxfordskej univerzite vedcami Karenom Simonyanom a Andrewom Zissermanom. Myšlienka na vznik modelu bola navrhnutá v roku 2013 a samotný model bol predstavený vrámci súťaže ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). Tvorcovia tejto siete vyhodnotili existujúce modely a rozhodli sa zväčšiť hĺbku pomocou veľmi malých konvolučných filtrov veľkosti 3x3 s veľkosťou kroku 1 v kombinácii s poolingovými vrstavami veľkosti 2x2 s veľkosťou kroku 2, čo ukázalo významné zlepšenie pri momentálnom stave strojového učenia. Jedná sa o sieť určenú na detekciu a kvalifikovanie objektov v obraze s úspešnosťou klasifikácie 92.7% pri klasifikovaní 1000 obrázkov z 1000 rôznych kategórií [19].



Obr. 4.4: Model VGG16. Zdroj: [19]

4.3 Rekurentná neurónová sieť

Tento typ sietí sa často používa ako asociatívne pamäte. Takáto pamäť je vnímaná ako ľudská schopnosť zapamätať si vzťahy medzi nepríbuznými pojmami, ako napr. medzi slovami a predmetmi(ich pomenovaním), alebo predmetmi a ich chuťou. Častým využitím sietí tohoto typu je predikcia nasledovnej časti vstupných dát alebo ich analýza, ako napríklad predpoveď nasledujúceho slova uvedeného textu, alebo rozoznávanie reči. Ako konkrétny príklad chovania takejto pamäte je možné uviesť čítanie textu. Počas čítania spracováva čitateľ text postupne slovo za slovom, pričom si udržuje spomienky na slová ktoré už v blízkej minulosti prečítal. Tým si vytvára priebežnú reprezentáciu toho, čo už prečítal. Na základe tejto reprezentácie je čitateľ schopný s určitou presnosťou predpovedať nasledujúce slovo textu ešte pred tým, ako mu bola táto informácia ponúknutá. Doposiaľ popisované siete pracujú vždy len so spracovávanou dávkou vstupných informácii, čo im neumožňuje vnímať sekvenčné dáta ako celok. To je spôsobené tým že nedisponujú žiadnou formou pamäte, do ktorej by bol model schopný ukladať doposiaľ zistené informácie zo spracovávanej sekvencie. Pokiaľ by mali byť zmieňované siete schopné korektne spracovať celú sekvenciu, bolo by potrebné aby dáta obsahujúce informácie o sekvencii neboli vkladané do modelu po dávkach, ale aby boli vložené ako celok. Narozdiel od nich popísaná biologická inteligencia jedinca spracováva informácie postupne a zároveň udržuje interný model toho, čo už v kontexte spracovala z predchádzajúcich dávok informácií, pričom sa tento model neustále aktualizuje pomocou novo nadobudnutých informácií.

Rekurentná neurónová sieť spracováva sekvenciu iterovaním cez jej jednotlivé prvky, pričom udržuje stav obsahujúci informácie vzťahujúce sa k tomu, čo doteraz v rámci rovnakej sekvencii videla. Na rozdiel od CNN má vnútorný cyklus, ktorý aktualizuje doposiaľ nadobudnutý stav, ktorý sa po ukončení spracovávania nezávislej sekvencie resetuje [6].

Kapitola 5

Existujúce riešenia

Aktuálne riešenia v oblasti vyhodnocovania hudby pomocou umelej inteligencie sa zameriavajú hlavne na odporúčanie nových skladieb pre používateľa audio streamingových platformiem ako sú napríklad Spotify, Apple Music alebo Youtube Music. Uvedené platformy ponúkajú používateľovi pravidelne nový (alebo aktualizovaný) zoznam skladieb v týždenných intervaloch na základe už používateľom obľúbených skladieb. Najdlhšiu históriu z aktuálnych riešení má spoločnosť Songza. V roku 2014 ju kúpila spoločnosť Google, ktorá využíva ich implementácie aktuálne na platforme Youtube Music.

5.1 Prístup Apple

Apple na klasifikáciu hudobných žánrov používa vlastný SoundAnalysis framework, ktorý má zabudovaný klasifikátor zvukov pri hudbe používaný ako extraktor príznakov. Ten je schopný v rámci analýzy audio súboru rozpoznať v ňom obsiahnuté jednotlivé zdroje zvuku(ako napr.: gitara, bicie, hlas...). Výstupom tohoto klasifikátoru je zoznam jednotlivých detekovaných zdrojov a k ním priradené *confidence score*, teda skóre v intervale $< 0; 1 >$ vyjadrujúce istotu algoritmu v prítomnosť daného zdroja. Pre skladbu je potom možné hlavne na základe tohoto skóre predikovať, o aký žáner sa jedná na základe splnenia podmienok, ktoré sú preň definované. Tieto podmienky sú definované ako prahy pre jednotlivé zložky audia a sú pre každý zdroj individuálne.[10]

5.2 Prístup Spotify

Spotify na rozdiel od Apple-u klasifikuje skladby na základe 12 koeficientov nimi definovaných príznakov. Medzi príznaky patria napríklad:

- **Key (Tónina)** Odhaduje použitú tóninu v skladbe, ktorú potom mapuje na číslo, napr.: 0 = C, 1 = C#, 2 = D
- **Loudness (Hlasitosť)** Priemerná hlasitosť nahrávky v decibeloch (dB). Hlasitosťou sa vyjadruje kolerácia medzi psychologickým vnímaním úrovne zvuku a ľuskou silou. Zvyčajne je v rozsahu medzi -60 a 0 db.

- **Speechiness (Rečnosť)**[1] Detekuje prítomnosť a pomer reči v rámci nahrávky. Na základe tohoto koeficientu je možné rozlíšovať, či sa jedná o hudobnú skladbu alebo aj o hovorené slovo(audio knihy, talk show . . .)
- **Liveness (Živosť)**[1] Detekuje prítomnosť živého publika pri nahrávaní. Pokiaľ hodnota koeficientu presahuje 0.8, je možné tvrdiť že sa jedná o nahrávku živého vystúpenia
- **Danceability (Tanečnosť)**[1] Reprezentuje ako veľmi je nahrávka vhodná na tanec. Pri tanečnosti sa zohľadňujú základné hudobné elementy ako je tempo, rytmus, alebo pravidelnosť.
- **Tempo** Udávané v BPM(počet úderov za minútu)
- **Duration** Dĺžka skladby (v milisekundách)[17]

Na základe týchto koeficientov je možné klasifikovať skladby aj nasledovne:

- Rap: Vysoká rečnosť
- Rock: Veľmi nízka tanečnosť.
- EDM: Vysoké tempo
- R&B: Dlhé nehrávky
- Latinská hudba: Vysoká tanečnosť
- Pop: Nezapadá výrazne do žiadnej kategórie

5.3 Shazam

Na rozdiel od vyššie zmienených spoločností pracuje firma Shazam na hľadaní zhody dvoch skladieb. Táto zhoda je hľadaná medzi užívateľom nahraným úsekom skladby a záznamom na serveri spoločnosti za účelom poskytnutia informácii ako je názov skladby a meno interpreta Aplikácia nahráva cez mikrofón zariadenia krátky, 15 sekundový audio súbor, ktorý následne prepošle na server pre rozpoznávanie piesne.

Pri porovnávaní zvukových signálov vytvára technológia „odtlačok“ audia, čo je proces extrahovania reprodukovateľných[2] tokenov obsahujúcich porovnateľné črty nahrávky. Tieto odtlačky sú porovnávané s odtlačkami skladieb uloženými v databáze, pričom sa hľadajú kandidátske skladby, pri ktorých v ďalšom priebehu rozpoznávania sleduje presnosť zhody. Medzi parametre známe pre porovnávací algoritmus firmy Shazam patria časová lokalita, prekladová symetria, robustnosť, a dostatočná entropia.

Časová lokalita určuje, aby každý odtlačok bol vypočítaný na základe vzoriek nachádzajúcich sa v blízkosti uvedeného bodu(vzdialené vzorky neovplyvňujú skúmanú vzorku). **Prekladová symetria** značí, že skúmaný odtlačok odvodený z konkrétnej vzorky je reprodukovateľný nezávisle od polohy v nahrávke pokiaľ časová lokalita, z ktorej boli vypočítané tokeny, je obsiahnutá v súbore (skúmaná vzorka môže pochádzať z ktorejkoľvek časti skúmaného audio súboru). **Robustnosťou** sa rozumie schopnosť reprodukovateľnosti z pôvodnej

[1] Nadobúda hodnotu na intervale $< 0; 1 >$.

[2] Reprodukovateľnosť vyjadruje schopnosť softvéru vytvoriť nahrávku bez šumu a skreslenia, ktorá je použiteľná na porovnávanie skladieb.

nahrávky. Jednotlivé tokeny by mali byť **dostatočne entropické** pre účel minimalizovania šance falošnej zhody v nekorešpondujúcich miestach medzi databázou a porovnávanou vzorkou. Nedostatočná entropia vedie k nadmerným a falošným zhodám mimo korešpondujúcich miest v skladbe, čo vedie k väčšiemu vyťaženiu zdrojov potrebných na správne rozoznanie piesne. Pri príliš veľkej entropii zas hrozí krehkosť jednotlivých tokenov, čo môže pri prítomnosti hluku a skreslenia viesť k nevyprodukovaniu zhody pri porovnávaní.[22] V roku 2018 bola spoločnosť Shazam odkúpená spoločnosťou Apple Inc.

Kapitola 6

Návrh implementácie

Práca na implementáciu používa CNN aj napriek tomu, že na sekvenčné dáta sa v realite používajú prevažne rekurentné siete. RNN sa pri sekvenčných dátach využívajú skôr na predikciu nasledujúceho prvku v skúmanej sekvencii, čo pri potrebnej klasifikácii dát do tried nie je potrebné. Ako bolo opísané v kapitole 4.3, konvolučná sieť je schopná spracovať sekvenčné dáta, pokiaľ jej bude celý skúmaný kontext poskytnutý ako súbor jednej dávky. Takouto dávkou je možné rozumieť aj formu v podobe obrázku, ktorá je pre CNN najčastejšou formou vstupných dát. Vkladané segmentované vstupné dáta síce pochádzajú z pôvodnej množiny dát tvoriacej pre každú nerozdelenú ukážku vlastný kontext, v realite je pre tento typ problému možné rozoznať takúto klasifikačnú triedy(hud. žáner) za zlomkový čas z trvania ukážky, typicky pár sekúnd. [21] Vstupné dáta majú štandardizovaný formát(spektrum mel) a veľkosť, čo taktiež dopomáha k lepšiemu výkonu CNN. Tá pracuje s fixne zadefinovaným vstupným a výstupným formátom dát oproti RNN, pri ktorej sa môže dĺžka sekvencie líšiť a taktiež si pri práci neukladá stav, ktorý dokáže tiež ovplyvniť jej výkon.

6.1 Učenie CNN

Navrhovaná CNN na rozpoznanie žánrov získava základnú vedomosť o rozdelení dát pomocou učenia s učiteľom(Supervised machine learning). Jedná sa o proces, kedy učiteľ disponuje vedomosťou, ako by mali byť vstupné dáta správne naviazané na výstup modelu. Tieto vedomosti sa učí model na základe učiteľom poskytnutého dostatočne veľkého a kvalitného datasetu.[15] Pri takomto datasete je dôležitá generalizácia dát, resp. poskytnutie takých dát, z ktorých dokáže model siete vyhodnotiť (alebo nájsť nové) príznaky(generalizovať dáta), ktoré sú pre danú triedu špecifické.[2] Dataset a jeho generalizácia sú preto jedny z najdôležitejších častí celého strojového učenia.

6.2 K-násobná krížová validácia

Správnosť siete je možné overiť rozdelením na trénovaciu a validačnú množinu. Po trénovaní bude model s novými poznatkami spustený na validačnej množine, pričom sa z tohoto procesu vypočíta validačné skóre, na základe ktorého vieme pri úpravách sledovať zlepšenie, resp. zhoršenie modelu. Z dôsledku malého počtu vstupných dát môže vyplývať aj malá validačná množina. Malý počet validačných dát dokáže v určitých prípadoch negatívne ovplyvniť výsledok validácie a teda aj konečnú štruktúru modelu. Takýmto prípadom

zabraňuje k-násobná krížová validácia. Pri jej použití je dátová množina rozdelená na K rovnako veľkých častí, z ktorých sa $k-1$ použije na trénovanie a posledná časť bude validačná. Cyklus validácie sa opakuje k krát, pričom pri každom behu je validačná časť zamenená za doteraz ešte na tento účel časť nepoužitú.

6.3 Podučenie a preučenie siete

Jedná sa o javy, ktoré vedú k nízkej výkonnosti modelovanej siete. Zatiaľ čo podučenie je menej častý jav spojený najmä s použitím nedostatočne vhodného datasetu na generalizáciu, oveľa väčší problém môže spôsobiť preučenie. To nastáva, pokiaľ sa model na trénovacích dátach začína učiť už nepodstatné detaily z prvkov datasetu alebo začína do vedomostí modelu zakomponovávať ich šum[2] . Oba javy je možné spozorovať na grafoch tréningovej a validačnej straty pre dataset. Podučenie sa dá spozorovať ako kontinuálne zlá výkonnosť počas celej fázy trénovania. Pri preučení začína v jednom bode pri stále znižujúcej sa trénovacej strate stagnovať strata validačná, prípadne môže začať aj rásť, čo značí zvýšenú chybovosť pri rozpoznávaní dát rovnakej triedy. Model takéto prvky nebol schopný správne začleniť, keďže nespĺňali dostatočné množstvo generalizovaných príznakov z danej triedy. Preučenie je možné riešiť napríklad použitím krížovej validácie alebo pridaním vrstiev typu `Dropout` spomenutých v 7.3. Medzi metódy krížovej validácie patrí aj K-násobná validácia.

6.4 Testovacia množina

Dáta do testovacej množiny správnej klasifikácie modelu sú získavané z viacerých zdrojov. Prevažne sa jedná o výberové CD nosiče, alebo populárne playlisty pre daný žáner. Pre zvyšné testy boli vytvorené menšie testovacie množiny, určené na testovanie prítomnosti konkrétnych príznakov špecifických danej problematike.

Keďže problematika ako správne rozdeľovať skladby do hudobných žánrov je veľmi subjektívna, počas testovania sa dohliada aj na čiastočnú úspešnosť klasifikácie. Tá sa zisťuje pri skladbách, ktoré neboli modelom klasifikované do správneho žánru, no predpokladaná hlavná zložka skladby tvorí aspoň 15% z výsledku klasifikácie modelu.

Pri testovaní je taktiež zavedená množina falošnej pozitivity. Množina pozostáva zo zoznamu o veľkosti klasifikačných tried, ktoré napĺňa testovací skript pomocou výsledkov získaných z klasifikátora. Pokiaľ nemá očakávaná trieda majoritné zastúpenie, pripočíta sa k indexu klasifikovanej majoritnej triedy nový výskyt. Falošná pozitivita sa do zoznamu nezapočíta len pod podmienkou, ak výsledok klasifikácie nadobudol maximálne 3 triedy s presne rovnakým klasifikačným podielom, pričom jednou z týchto tried je práve očakávaný výstup klasifikácie.

Kapitola 7

Implementácia

Táto kapitola obsahuje popísané jednotlivé nástroje, triedy a techniky použité pri implementácii modelu. V kapitole 7.1 je uvedený súhrn pre prácu najdôležitejších komponentov. Následne je kapitola rozdelená na 3 tématické okruhy venujúce sa vlastnej problematike. V podkapitole 7.2 je rozobrané spracovávanie datasetu na použiteľnú formu vstupných dát pre implementovaný model. Ďalšou časťou je opis architektúry implementovaného modelu v podkapitole 7.3 aj spolu s problémami ktoré sa vyskytli počas trénovania. Pre aplikáciu bolo vytvorené užívateľské rozhranie, spolu s jej komunikáciou s analyzačným skriptom je popísaná v podkapitole 7.5.

7.1 Prostredie a použité nástroje

Aplikácia bola implementovaná pomocou programovacieho jazyka `Python 3.8` s knižnicou pre grafické rozhranie `PyQt5`. Aplikáciu je možné spustiť cez konzolu zadaním „python main.py“ v domovskom adresári. Pre tvorbu modelu neurónovej siete a jeho trénovanie sa použil framework TensorFlow[1] s Keras API[5]. O plotovanie dát (vytváranie spektogramov) poskytnutých pre model CNN sa stará knižnica Librosa. Aplikácia bola testovaná na operačnom systéme Windows 10. Model bol trénovaný pomocou grafickej karty NVIDIA GeForce GTX 1060 s dedikovanou pamäťou 6 GB. Na získanie potrebných metrík pre optimalizáciu využívania zdrojov bola použitá knižnica `pympler`.

7.2 Spracovanie vstupných dát

Vstupné dáta pre trénovanie vytvára trieda `soundExtractor.py`. Tá prechádza jednotlivé adresárované triedy datasetu, pričom pre každú nájdenú ukážku segmentuje a pre jednotlivé časti vytvára spektogramy o stanovenej dĺžke 5 sekúnd.Na základe dĺžky skladby sa cyklicky spúšťa funkcia `plot_audio`, ktorá vytvára spektogramy jednotlivých segmentov a vkladá ich do patričného adresáru pre spracovávaný žáner.

Vytváranie spektogramov plotovaním audio súboru zabezpečuje knižnica `Librosa` pomocou funkcie `melspectogram()` z jej modulu `feature`. Pri spracovávaní audio signálu používa plotovacia knižnica `PySoundFile` (wrapper pre C knižnicu `libsndfile`). Pokiaľ sa analyzovaný audio formát nenachádza v tejto knižnici (ako napríklad MP3), použije Librosa knižnicu `audioread`, ktorá je pomalšia, ale poskytuje podporu rozsiahlejšieho počtu audio formátov. Predvolený výstup plotovaných dát obsahuje pre učenie siete nepodstatné údaje, napr. v podobe vyznačenia osí. Všetky možnosti pridania dodatočných informácii

pre výsledný graf sú teda vypnuté. Tým sa veľkosť trénovacích dát síce viditeľne zmenší, eliminujú sa však možné zdroje šumu, čo by malo dopomôcť k lepšiemu učeniu neurónovej siete.

7.2.1 Rýchlosť spracovania nahrávok

Pri takomto vytváraní spektogramov sa dĺžka práce na jednom spektograme úmerne zväčšuje s dĺžkou audio súboru. Je to zapríčinené hlavne pomalým správaním `garbage collectoru`, ktorý pri väčšom objeme zahadzovaných dát nestíha uvoľňovať pamäť pre beh programu dostatočne rýchlo. Toto správanie je možné odpozorovať pri plotovaní nového audio súboru. Zatiaľ čo pri prvotných snímkoch pracuje plotovacia funkcia s rýchlosťou na vytvorenie spektogramu pod jednu sekundu, pri posledných spektogramoch sa vytvára jeden spektogram za viac ako 20 sekúnd v závislosti od dĺžky súboru (uvažujeme štandardnú dĺžku skladby v priemere 3 minút) a teda od počtu opakovaní spustenia plotovania.

7.2.2 Urýchlenie procesu

Toto spomalenie je možné minimalizovať pomocou troch procedúr. Prvou je mazanie premenných pomocou kľúčového slova `del` v jazyku Python. Takto vymazané objekty z menného priestoru nezapĺňajú garbage collector tak rýchlo, aby dokázal obmedzovať rýchlosť behu programu pre nedostatok miesta vo vyrovnávacej pamäti. Druhou procedúrou na urýchlenie tohoto procesu je definovanie konkrétneho renderovacieho jadra pracujúceho v back-ende, pomocou ktorého má knižnica `matplotlib` vykresľovať a zapisovať jednotlivé spektogramy do súborov. Ako predvolené jadro používa `matplotlib` knižnicu Agg[1]. Pre urýchlenie vytvárania spektogramov bola použitá knižnica `TkAgg`, ktorá spracováva dáta rovnako, no pre vykresľovanie používa prvky grafickej knižnice `Tkinter`. Treťou procedúrou je multiprocessing. Keďže jednotlivé spektogramy sú od seba nezávislé dáta, je možné rozdeliť priebeh programu do viacerých procesov. Knižnica multiprocesingu v jazyku Python definuje `pool` tzv. *workerov*, počet pracujúcich paralelizovaných procesov komunikujúcich pomocou implementovaného API. V prípade vytvárania spektogramov je použitý asymetrický multiprocessing, pri ktorom je vytvoreným procesom prideľovaná práca tzv. master-procesom, ktorým je hlavné vlákno programu. Týmto sub-procesom je prideľované spracovanie jednotlivých segmentov audio súboru nerovnomerne a teda každý ďalší nespracovaný segment je pridelený novo uvoľnenému vláknu bez ohľadu na ich celkový podiel práce.

7.3 Architektúra neurónovej siete

Na vytvorenie CNN bola použitá knižnica Tensorflow spolu so svojím rozhraním pre programovanie aplikácii (z angl. API, Application programming interface) Keras. Jedná sa o jednu z najpoužívanejších knižníc pre strojové učenie a je využívaná hlavne na extrakciu príznakov zo zvukových, obrazových a textových súborov, čo predstavuje značné možnosti využitia konvolučných a rekurentných neurónových sietí.

[1] Anti-Grain Geometry, multiplatformová 2D grafická knižnica v C++

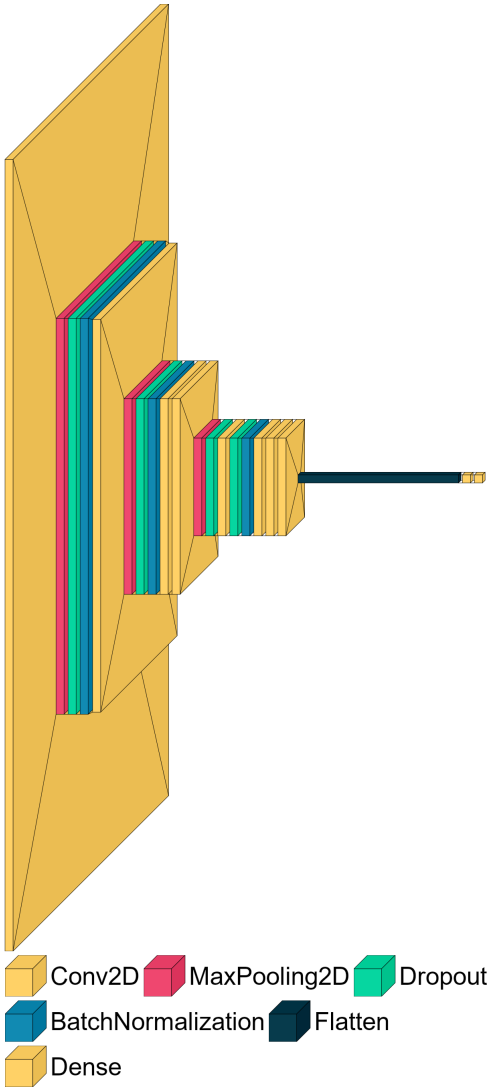
7.3.1 Model neurónovej siete

Keďže navrhovaná neurónová sieť pracuje s obrazovými dátami, je jej model postavený na využívaní konvolučných vrstiev spomínaných v sekcii 4.2.1. V rozhraní Keras predstavuje model objekt, do ktorého je možné vkladať inštancie tried jednotlivých už implementovaných vrstiev využívaných pri strojovom učení. Samotný objekt pri tom slúži ako organizačná štruktúra, z ktorej je možné napríklad vytvárať organizačné bloky vrstiev, alebo volať jednotlivé vrstvy.

Navrhovaný model má štruktúru lineárneho zásobníku (podľa triedy `Sequential`) a prijíma vstupné dáta vo forme 2D tenzoru (matice) - obrázku. Vstupnou vrstvou pre model je prvá z konvolučných vrstiev, ktorá spracováva tenzor v jeho pôvodnej veľkosti a následne ho posúva ďalším vrstvám. Za každým balíkom konvolučných vrstiev je umiestnená poolingová vrstva typu `max pooling`, ktorá pomáha prechádzať modelu postupne do hĺbky a teda s každým prechodom hľadať menšie príznaky špecifické pre konkrétnu triedu. Model obsahuje takisto vrstvy typu `dropout` slúžiace k minimalizovaniu preučenia modelu. Ich úlohou je meniť vstupné dáta jednotlivých tenzorov na polia vyplnené 0, čo povedie k zabráneniu aktivácie funkcie, v prípade tejto práce sa jedná o ReLU (na základe výpočtu vzorcu z 3.5), a následnému „vyhodeniu“ daného tenzoru z učenia. Po prechode jednotlivými balíkmi prichádza tenzor k poslednej, výstupnej vrstve typu `Dense`, ktorá vyhodnotí prítomnosť jednotlivých hľadaných tried a pravdepodobnosť pridruženosti vstupných dát k definovaným triedam pomocou svojej aktivačnej funkcie typu softmax.

Implementovaný model neurónovej siete predstavuje zjednodušenú verziu populárneho modelu VGG16 ukázaného v kapitole 4.2.2, obohatený o k-násobnú validáciu. Dá sa predpokladať, že model uvedený v [21] bol tiež vytvorený ako jeho jednoduchšia verzia. Pri vytváraní jednotlivých vrstiev už nie je potrebné určovať veľkosť recepčných polí, ani veľkosť kroku, keďže predvolené hodnoty týchto parametrov pri vytváraní konovolučných a poolingových vrstiev sú v API Keras totožné s modelom VGG16.

Výstupom modelu je teda jednorozmerný zoznam o veľkosti počtu definovaných žánrov datasetom GTZAN (10, definované v 2.2), ktorého obsahom je pre každý žáner vypočítaná pravdepodobnosť pridruženia vstupu k triede.



Obr. 7.1: Návrh modelu pre CNN

7.4 Trénovanie CNN pre rozpoznávanie hudobných žánrov

Trénovanie konvolučnej siete zabezpečuje trieda `training.py`, v ktorej je konfigurovaný model pomocou parametrov funkcie `compile`. Ako optimalizačný algoritmus bol vybraný `Adam`, aktuálne populárny vďaka jeho efektivite, ktorú získali jeho autori kombinovaním algoritmov `AdaGrad` a `RMSProp`, čím dosiahli získavanie dobrých výsledkov za krátky čas.

Samotný dataset GTZAN neobsahuje pre potreby tejto práce dostatok trénovacích dát, preto je potrebné dostupné dáta patrične upraviť pre získanie väčšej trénovacej množiny. Keďže výsledok spektogramov je štandardizovaný na fixný pomer (dB za vzorku), nie je možné transformovať získané obrázky(napr. transformovaním obrázku cez os `y`). Viac vstupných dát pre trénovací model preto zabezpečuje segmentácia ukážok piesní na viac spektogramov. Pre každý žáner je dataset rozdelený na 4 rovnomerne veľké časti. Takto rozdelené dáta slúžia ako trénovacia a validačná množina zároveň za použitia K-násobnej validácie.

Na optimalizáciu práce s pamäťou boli pre trénovacie a validačné dáta použité generátory `DataImageGenerator` z API Keras. Generátory počas trénovania uchovávajú v pamäti iba jednu trénovaciu dávku, čím predchádzajú vyčerpaniu pridelenej pamäte pre trénovanie. Implementácia využíva knižnicu `NumPy`, z ktorej využíva operácie na prípravu a spracovanie matíc.

7.4.1 Trénovanie s K-násobnou validáciou

Vrámci triedy `soundExtractor.py` prebieha po vytvorení všetkých spektogramov ich rozdelenie do 4 rovnako veľkých častí vo funkcii `divide_spectograms`, ktoré sa využijú pri trénovaní s pomocou k-násobnej validácie. Funkcia získava zoznam všetkých spektogramov prehliadaním adresára určeného pre konkrétnu triedu klasifikácie. Následne funkcia zoznam zamieša a vytvára trénovacie balíky (adresáre). Štruktúra takéhoto balíku pozostáva z 2 častí, trénovacej a validačnej, pričom pomer uložených spektogramov v nich je 1:3. Pri tvorbe týchto balíkov sú spektogramy do nich kopírované, čím je umožnené následné iné rozdelenie pôvodného datasetu spektogramov.

Pre využitie k-násobnej validácie je trénovacia funkcia `fit` z API Keras obalená v cykle spoločne spolu s generátormi trénovacej a validačnej množiny. Obsah týchto generátorov je menený každý jeden cyklus, aby sa zabezpečilo správne fungovanie tejto procedúry.

Pri trénovaní siete sa zbierajú štatistiky o strate a presnosti validácie a trénovania. Pri každom cykle trénovania sa ukladajú zaznamenávané metriky do polí, ktoré po dokončení trénovania spracováva knižnica `NumPy`. Tá priemeruje každú jednu metriku a pomocou knižnice `Librosa` sa po ukončení testovania z priemerovaných metrík vytvárajú grafy pomocou funkcie `create_training_graphs`.

7.4.2 Dávková normalizácia

Keďže sa klasifikácia implementovaného modelu týka štandardizovaných dát v podobe zastúpenia jednotlivých frekvencií v škále mel ako bolo popísane v kapitole 2.1.1 je možné takýto vstup vhodne normalizovať bez obáv zo straty dôležitých príznakov, čím je možné predísť presaturovaniu jednotlivých výrazných príznakov. Prílišná saturácia môže vzniknúť ako vedľajší efekt pri používaní filtrov jednotlivých vrstiev tak, že od začiatku výrazné príznaky budú vďaka váhovým maticiam jednotlivých filtrov ešte viac výraznejšie, čím môže model dospieť k ignorovnaniu určitej sady príznakov. Dáta mohli síce prísť do vstupnej vrstvy štandardizované, nie je však pravdepodobné že po použití sád filtrov jednotlivých vrstiev budú dáta štandardizované aj na jej výstupe.

Dávková normalizácia (anglicky batch normalization) využíva rôzne metódy(napríklad preškálovaním výstupných dát okolo hodnotu 0 a nastavením odchlýlky na 1, čím sa vytvára normálne rozloženie výstupu) aby takémuto správaniu modelu predišla. Tento druh vrstvy dokáže zlepšiť generalizáciu modelu,čo v konečnom výsledku zlepšuje správnosť klasifikácie. Takúto vrstvu je preto vhodné dať vždy pred použitím prvej aktivačnej funkcie z bloku vrstiev pracujúcich s filtrami.

7.4.3 Preučenie siete

Pri trénovaní modelu s pomocou k-násobnej validácie sa podarilo modelu dosiahnuť 100% validačnú úspešnosť už počas posledných epoch 2. zo 4 balíkov určených na trénovanie. Preto bolo potrebné zmenšiť počet epoch a pridať konvolučné vrstvy, ktorými prechádza model počas trénovania. Pri následnom testovaní siete sa to viditeľne prejavilo na výsledkoch.

Naviac na preučenie trpela trieda blues, pri ktorej sa úspešnosť klasifikácie pohybovala na úrovni 27%. Väčšinu z testovacej množiny klasifikoval model ako rock, ktorý s blues zdieľa mnoho čŕt. Miera misklassifikácie na úkor rocku bola až 67% percent. Model z [21] dokázal byť vhodne natrénovaný za dobu 70 epoch, nevyužíval však krížovú validáciu, ktorá dokáže redukovať počet potrebných epoch v jednom validačnom behu na približne štvrtinu.

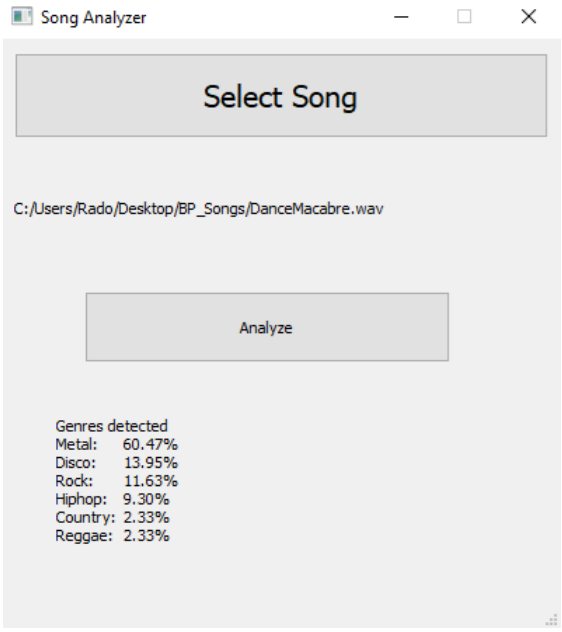
7.5 Užívateľské rozhranie

Pre zjednodušenie práce s modelom pri vkladaní nahrávok užívateľom bola vytvorená jednoduchá aplikácia s užívateľským rozhraním. Aplikácia bola implementovaná v programovacom jazyku `Python 3.8` s implementovaným grafickým užívateľským rozhraním založeným na prvkoch knižnice `PyQt5`. Do implementovanej aplikácie je možné vložiť audio nahrávku a spustiť analýzu súboru, ktorá po skončení analýzy nahrávky modelom vyhodnotí percentuálnu pridruženosť danej nahrávky k jednotlivým definovaným žánrom.

Rozhranie aplikácie bolo navrhnuté pomocou aplikácie Qt Designer a je načítavané ako súbor `window.ui` pomocou skriptu v `main.py`.

Po spustení aplikácie načíta trieda `UI` hlavné okno, v ktorom je možné vkladať jednotlivé nahrávky na analýzu navrhnutou sieťou. Užívateľ klikne na tlačidlo „Select Song“, ktoré otvorí dialógové okno pre výber súboru. Toto okno je nastavené tak, aby ukazovalo ako prvotnú lokáciu pre výber koreňový adresár `C://`. Toto výberové okno obmedzuje výber súborov na audio formáty WAV, FLAC a MP3, na ktorých bola aplikácia testovaná a zároveň sú aj najpoužívanejšími. Po výbere súboru sa odomkne možnosť spustiť analýzu aktivovaním tlačidla „Analyze“.

Pri spúšťaní analýzy je pre správny chod aplikácie potrebné vytvoriť nové vlákno, aby sa zabezpečil plynulý chod grafického rozhrania, ktorý by bol inak závažne obmedzený behom analýzy, keďže sa jedná o výpočtovo náročný proces. To sa vytvorením nového vlákna úplne eliminuje. V novom vlákne je spúšťaný nezávislý skript `songPredictor.py`, ktorého spúšťací parameter `s` je získaný z dialógového okna a určuje cestu k súboru určeného na analýzu.



Obr. 7.2: Vzhľad implementovanej aplikácie

7.5.1 Komunikácia aplikácie s analyzačným skriptom

Vlákno aplikácie komunikuje so skriptom pomocou implementovaných `socketov` naviazaných na `localhost` zariadenia. Vrámci tejto komunikácie sa vytvára spojenie typu klient-server, v ktorom predstavuje vlákno aplikácie server a spúšťaný skript klientskú časť. Po užívateľovom stlačení tlačidla „Analyze“ sa vo vlákne určenom na analýzu vytvorí ďalšie vlákno venujúce sa vytvoreniu a previazaniu socketu s bežiacou aplikáciou, zatiaľ čo sa v rodičovskom vlákne volá skript `songPredictor.py`, ktorý vykonáva klasifikáciu nahrávky. Toto vlákno si vyberá na komunikáciu so skriptom port z rozsahu mimo well-known ports (interval $< 1024, 65535 >$), na ktorom očakáva odpoveď s výsledkami klasifikácie. Po ukončení klasifikácie sa v skripte vytvorí socket, ktorý odošle výsledky klasifikácie späť vedeniu aplikácie, ktoré ich ďalej spracováva.

7.6 Spracovanie vloženej nahrávky analyzačným skriptom

Aplikácia spracováva poskytnutú nahrávku rovnako ako nahrávky z datasetu v sekcii 7.4. Keďže sa jedná o dočasné dáta, ukladá aplikácia spektogramy do priečinku určeného pre dočasné súbory, ktoré sú odstraňované vždy po ukončení analýzy nahrávky. Každé jedno volanie skriptu vytvorí nový priečinok, ktorý je pomenovaný podľa čísla portu (pokiaľ bol port definovaný ako spušťací parameter), alebo sa vyberie náhodné číslo v intervale $< 80000, 99999 >$, čo dovoľuje súčasný beh viacerých inštancií skriptu (odporúča sa len pri silnom výpočtovom výkone zariadenia). Pokiaľ je skript spúšťaný s definovaným portom, je platné obmedzenie, aby sa nejednalo o well-known port, a teda aby sa predišlo zlyhaniu komunikačného rozhrania.

Spektogramy jednotlivých segmentov sa po ukončení ich vytvárania posúvajú modelu CNN ako parameter do funkcie `analyse_song`, ktorá pre každý segment vypočíta pravdepodobnosť klasifikácie do konkrétnej triedy pomocou aktivačnej funkcie `softmax` spomínanej

v kapitole 3.5. Každý detekovaný žáner vrámci nahrávky má k sebe pripísaný svoj percentuálny podiel zo segmentu, pričom sa ako finálny žáner segmentu považuje práve ten, ktorého podiel je v množine klasifikovaných žánrov najvyšší. Údaje o tom, v ktorom segmente je aký žáner dominantný, sú potom uchovávané v zozname, ktorý funkcia vracia ako návratovú hodnotu ďalej vedeniu programu. Zo získaného zoznamu sa vytvára slovník napĺňaný pomocou knižnice `NumPy`, ktorá v pôvodnom zozname zisťuje početnosť výskytu jednotlivých žánrov. Klasifikácia v takto vytvorenom slovníku je zoradená podľa početnosti jednotlivých detekovaných žánrov. Ten je buď predávaný naspäť vo forme poslania správy socketu definovanému ako parameter pri spustení, alebo je vypísaný do konzolovej aplikácie z ktorej bol spustený. Výsledok klasifikácie poskytuje skript vo forme slovníku, kde je ako kľúč vyjadrený slovný názov klasifikačnej triedy, ktorý je v dvojici s percentuálnym podielom klasifikácie tejto triedy. V prípade odosielania odpovede pomocou socketu je využitá knižnica `pickle`, pomocou ktorej sú dáta enkódované do formy bajtového streamu.

Kapitola 8

Testovanie

Testovanie správnosti navrhnutého modelu prebiehalo s pomocou skriptu **genre_test.py**. Najdôležitejším testovaním je test správneho určenia v kapitole 8.1. Vzhľadom na vek datasetu je zaujímavé sledovať výsledky nových skladieb trénovaných na starších dátach v kapitole 8.3.

8.1 Test správneho určenia

Testom správneho učenia rozumieme schopnosť modelu neurónovej siete správne klasifikovať analyzovanú nahrávku do správnej klasifikačnej triedy. Pre potreby testu boli vytvorené sady ukážok populárných skladieb každého klasifikovateľného hudobného žánru z datasetu GTZAN. Testované ukážky majú rôznu dĺžku, čím sa testuje schopnosť modelu vysporiadať sa s nepravidelnou veľkosťou vstupu. Očakávaným výsledkom testov je vyhodnotenie jedného žánru so správnou predikciou klasifikácie s očakávaným žánrom tvoriacim modus majoritných výsledkov zo segmentov. Pre prípadné skúmanie čiastočnej zhody žánru sa očakáva jeho prítomnosť v prvých troch najpočetnejších klasifikovaných tried spolu so zastúpením minimálne 15%. Spojením čiastočnej zhody so zhodou istou sa definuje *kombinovaná klasifikačná presnosť*. Tento pojem bude využívaný v neskorších podkapitolách testovania. Bol vytvorený na pokrytie konkrétnych prípadov pri skladbách, ktoré sú špecifické určitým žánrovým prekrytím, ktoré je často ťažko rozlíšiteľné, pričom práve takéto prekrytie zvyčajne vytvára problematiku správneho zaradenia spojenú so subjektívnym vnímaním poslucháča.

Najviac problematickým sa stal pre trénovanie žáner blues. Z testovacej množiny bol model schopný vyhodnotiť správne len vyše 5% testovaných skladieb. Falošnú pozitivitu pri väčšine chybnej klasifikácie tvorila klasická hudba a country(žáner s čiastočným pôvodom z blues). Rovnako málo presvedčivé výsledky poskytlo testovanie žánru jazz. Pri ňom bolo možné pozorovať ako hlavný žáner podľa klasifikácie klasickú hudbu na všetkých testovaných skladbách, ktoré neboli vyhodnotené ako primárna klasifikácia zhody.

Prekvapivým bolo aj testovanie množiny pre klasickú hudbu, ktorá obstála bez chyby počas celého testovania. Na žánri klasickej hudby bolo taktiež testovanie podpory pre formát FLAC. Spolu s jazzovými skladbami sú klasické diela špecifické svojou dĺžkou, ktorá môže presiahnuť aj dobu 10 minút. Takáto dĺžka taktiež analyzačnému skriptu neprekážala.

Počas testovania reggae si počas testovania bolo možné všimnúť relatívne rovnomernú distribúciu medzi triedy pri falošnej zhode. Podelili sa o ňu disco, hip-hop a pop, teda 3 žánre, ktoré si môžu byť podobné v nižších frekvenčných zónach určených pre basové linky.

Žáner	Zhoda [%]	Čiastočná zhoda [%]	Chyba [%]
Blues	5.56	0	94.44
Klasická hudba	100	0	0
Country	85.71	14.29	0
Disco	56.25	12.5	31.25
Hip-hop	21.05	31.58	47.37
Jazz	23.53	23.53	52.94
Metal	62.5	18.75	18.75
Pop	70.59	11.76	17.65
Reggae	47.62	19.05	33.33
Rock	46.67	20	33.33

Tabuľka 8.1: Klasifikácia testu správneho učenia

Počas testovania implementácie jednotlivých architektúr bolo možné sledovať správne určenie konkrétnych žánrov, spravidla sa jednalo o dvojicu klasickej hudby a metalu. Táto dvojica tried mala vysokú mieru správnej klasifikácie (nad 70% výsledkov bolo správne klasifikovaných) takmer pri každej iterácii vývoja modelu, nezávisle na tom či bola daná sieť pretrénovaná, alebo podtrénovaná.

8.2 Žánrové prekrytie

V dnešnej dobre je stále častejšie medzi umelcami prekrývanie jednotlivých hudobných žánrov ako napríklad pop kombinovaný s elektronickou hudbou, alebo hip-hop spojený metalom. Pri tomto testovaní sledujeme, či bude analyzovaná nahrávka obsahovať vo vyhodnotení klasifikácie minimálne 2 majoritné triedy, ktoré sa budú zhodovať s predpokladaným výstupom. Pri takomto testovaní je možné uvažovať pri analyzovaní výsledkov kombinovanú klasifikačnú presnosť, pokiaľ bude jeden zo žánrov, ktorým sa predpokladá majoritný podiel potrebovať pre úspešnú validáciu takú časť podielu, ktorú mu vie jeho komplementárny žáner doplniť. Komplementárnym žánrom rozumieme, taký žáner, ktorý zo špecifikovaného vychádza, alebo ho svojou tvorbou značne ovplyvňuje, napr. spojenie rocku a metalu. Na rozdiel od príkladov uvedených v 8.3 sa jedná o viac priamy a vedomý cieľ prekrytia techník z dvoch kategórií.

Príkladmi takýchto piesní môžu byť napríklad:

- Kombinácia elektronickej hudby a pop-u

  Jedná sa o jednu z momentálne najčastejších kombinácii, ktorú je možné sledovať od dôb vzniku disca, kedy sa začali vo veľkej miere používať syntetizátory. Postupom času sa z disca vyvinul jeho podžánrov vyvinula elektronická tanečná hudba (EDM, z angl. electronic dance music). Tento žáner sa stal populárnym medzi mladými ľuďmi na prelome milénia a doteraz má veľké zastúpenie aj napriek tomu, že nie je tak výrazný ako pár rokov dozadu. Príkladom na kombináciu týchto žánrov môže byť skladba švédskeho zoskupenia Swedish House Mafia, ktorí v kooperácii s umelcom The Weeknd vyprodukovali skladbu *Moth To A Flame*. Z výsledkov obsiahnutých v tabuľke 8.2 je možné pozorovať okrem najväčšieho zastúpenia popu aj výraznú stopu disca, z ktorého základov vychádza práve EDM.

Zložka	Klasifikačný podiel[%]
Pop	30
Country	26
Disco	20
Iné	24

Tabuľka 8.2: Klasifikácia skladby Moth To A Flame

- Kapela Ghost

  Ghost je momentálne jednou z najpopulárnejších novodobých metalových kapiel. Medzi fanúšikmi tohoto žánru však často vzbudzuje kontroverziu práve z pohľadu, ako ktorý žáner by mala byť klasifikovaná. Pri kompozícii piesni totiž táto švédska kapela využíva často prvky disca, čím vytvára veľmi špecifickú kombináciu, ktorá sa nemusí každému páčiť.

Názov skladby	Metal [%]	Disco [%]	Majoritný podiel [%]
Dance Macabre	25.58	13.95	Hip-hop (30.23)
Hunter's Moon	25.64	15.38	Pop (33.33)
Year Zero	38.81	8.96	Metal (38.81)

Tabuľka 8.3: Klasifikácia skladieb kapely Ghost

  Na výsledkoch uvedených v tabuľke 8.3 je možné vidieť vždy vysoké percentuálne obsadenie metalu, čo značí vždy úspešnú detekciu predpokladaného majoritné žánru, a teda sa naplňuje minimálne podmienka klasifikácie skladieb ako čiastočnej zhody. Okrem správne klasifikovaného metalu je možné sledovať vždy aj neprehliadnuteľnú prítomnosť disca. To sa dokázalo vždy klasifikovať vždy do 2 priečok po zoradení klasifikácie podľa pomeru zložiek. Faktom taktiež je, že práve disco zdieľa veľa podobných čŕt s popom a hip-hopom, ktoré dokázali pri prvých 2 ukážkach predbehnúť metal a obsadiť prvú priečku v rámci výskytu. Z prezentovaných výsledkov je teda možné vskutku potvrdiť značnú prítomnosť elementov tohoto žánru v spojení s metalovou kapelou.

- Nu-metal

Zložka	Klasifikačný podiel[%]
Hip-hop	64.29
Metal	19.05
Rock	9.52
Iné	7.14

Tabuľka 8.4: Klasifikácia skladby Rollin'

  Na prelome milénia bol populárny metalový podžáner s názvom *nu-metal*. Po hudobnej stránke špecifický kombinovaním typického skresleného zvuku elektrických gitár spojených so „scratchovaním“ platní a samplovaním zvukom[1]. Jednou z najvýraz-

[1] metóda prekrývania nahraných zvukov

nejších kapiel bol Limp Bizkit, ktorých pieseň *Rollin'* bola klasifikovaná modelom nasledovne:

- Taylor Swift
  Speváčka Taylor Swift je jednou z napopulárnejších country umelkýň, ktorej značná časť tvorby je považovaná popovú hudbu. V uvedenej tabuľke 8.5. Zastúpenie týchto žánrov je pre jej skladby tak silné, že v prípade kombinovanej klasifikačnej presnosti tento koeficient prejavuje pri jej tvorbe najvýraznejšie, a to v priemere viac ako 70% všetkých klasifikovaných segmentov.

Názov skladby	**Pop** [%]	**Country** [%]
Our Song	70	25
Love story	40.82	30.61
Sparks Fly	38.46	17.31

Tabuľka 8.5: Klasifikácia skladieb kapely Ghost

- Reggae verzia Take Me Home, Country Roads (country)
  Medzi testovanými skladbami z množiny zameranej na reggae sa náchádzal cover na pieseň od Johna Denvera, ktorá sa dá považovať za najpopulárnejšiu z celého žánru country. Táto pieseň v pôvodnom znení bola správne klasifikovaná modelom s podielom pre country vo výške 71.79%. Pri cover verzii do štýlu reggae určil klasifikátor určil žáner ako pop, pričom je možné pozorovať aj výraznú zložku country, s viac ako štvrtinovým podielom. Model neklasifikoval pre reggae žiadny z vytvorených segmentov.

Zložka	**Klasifikačný podiel** [%]
Pop	53.66
Country	26.83
Hip-hop	9.76
Iné	9.75

Tabuľka 8.6: Klasifikácia Reggae verzie Take me home, country roads

8.3 Evolučné problémy žánrov

Podobne ako pri biologických organizmoch platí pre hudbu, že sa stále vyvíja. Tak, ako raz prešiel trend vývoja z ľudovej hudby do klasickej, aj stále vznikajúce nové žánre populárnej hudby čerpajú z už existujúcich predlôh. Aktuálne je možné uvažovať za najstaršie žánre populárnej hudby práve blues a síce mladší, ale stále aj z neho silne vychádzajúci jazz. Takéto prepojenie vytvára predpoklad, že sa pri klasifikovaní konkrétnych žánrov môže objaviť ich zamenenie pri klasifikácii trénovaným modelom, keďže pri kompozícii skladieb využívali ich autori rovnaké alebo podobné techniky, ktoré môže model vyhodnotiť ako príznak klasifikácie viacerých tried.

Je teda možné hypotetizovať, či je možné vidieť túto zámenu napr. medzi žánrami rock a metal, ktorý sa vyvinul práve z rocku. Z výsledkov testovania uvedených v tabuľke 8.1 je možné pri rocku vidieť vyššiu kombinovanú klasifikačnú presnosť, ktorú spôsobila klasifikácia skladieb práve zamenením týchto 2 žánrov.

Takéto vyhodnotenie môže byť spôsobené práve úmyselným prekrývaním nosných hudobných žánrov opísaných v 8.2. V prípade klasifikácie uvedenej testovacej množiny bola falošná pozitivita zaznamenaná práve pri novších skladbách, konkrétne z podžánru alternatívneho rocku, ktorý často preberá agresívnejší zvuk práve od metalu.

Príkladom takéhoto prepojenia môže byť aj podžáner EDM, house. Pri posluchu housových skladieb nie je pochýb že by nesplňovali predpoklady byť elektronickou hudbou vychádzajúcou z disca čo navodzuje určitý predpoklad jeho prítomnosti, no model CNN dokázal pieseň klasifikovať pre tak značnú časť segmentov ako pop. House totiž často tvorí základnú basovú linku pre popové skladby, čo sa odrazilo aj na výsledkoch testovania. Pri tomto teste bola vytvorená špecifická množina skladieb z EDM, na ktorej bola testovaná klasifikácia pomocou žánru disco. Výsledky testovania tejto množiny udávajú najvyššiu zhodu práve so žánrom pop, pričom klasifikácia skladieb do žánru disca bola úspešná len pri približne 12% prípadov.

Rovnaké výsledky ako v testovaní žánru disca tvoril najväčší podiel falošnej pozitivity práve pop, ktorý z neho preberá hlavne prvky tvoriace basovú linku v skladbách. Takáto zlá klasifikácia môže byť do určitej miery spôsobená neaktualizovaním použitého datasetu.

Pri tejto kolekcii je potrebné podotknúť, že bola vydaná v roku 2001, pričom nebola odvtedy aktualizovaná. Preto je asi vhodnejšie na dnešné pomery vymeniť kategóriu Disco za žánre elektronickej hudby, ktoré sa z neho postupne časom vyvinuli do dnešnej podoby

8.4 Testovanie vplyvu spevu na žáner

Ako bolo zmienené v 2, text skladieb neovplyvňuje výsledok klasifikácie nahrávky modelom. To ale neznamená, že vytvárané spektogramy neobsahujú zložku signálu vytvorenú práve hlasivkami speváka. V tomto teste sa preto uvažuje o hudobnom prejave speváka ako o rozdielovej zložke klasifkácie, pričom sa skúma vplyv spevu na výsledky získane z modelu. Pre testovanie boli vytvorené 2 množiny náhravok rovnakých skladieb. Okrem štúdiových nahrávok v prvej množine klasifikuje model tzv. „karaoke“ nahrávky. Jedná sa o typ nahrávok, v ktorých je signálová zložka spevu odfiltrovaná, alebo nebola pri kompozícii nahrávky prítomná.

Skladba	Sledovaná zložka	Klasifikačný podiel	Podiel v inštrum. verzii	Rozdiel	Najväčsia zmena
Dr. Alban - It's My Life	Disco	17.39	22.45	+5.06	Hip-hop(-14.51)
Timbaland - Apologize	Pop	67.57	75.68	+8.11	Klasická hudba(-8.1)
50 Cent - In Da Club	Hip-hop	30.61	45.45	-14.84	Pop(+25.6)
Alan Walker - Faded	Pop	61.9	30.95	-30.95	Pop (-30.95)
Metallica - Master of Puppets	Metal	80.39	76.7	-3.69	Disco(+5.82)
Dua Lipa - Physical	Pop	82.22	32.5	-49.72	Disco(+23.23)

Tabuľka 8.7: Klasifikácia originálnych skladieb s filtrovanou stopou hlasu

Z testovania je možné odpozorovať veľmi silný vplyv spevu pri popových skladbách z posledných rokov (Faded, Physical). V prípade odfiltrovania zvukovej stopy hlasu, padla klasifikácia žánru pop na polovičné hodnoty. Takýto pokles už ale nebolo možné pozorovať pri piesni Apologize z roku 2009. V nahrávke s odfiltrovanou hlasovou stopou klesol hlavne podiel klasifikácie country a klasickej hudby, ktorú nahradila skoro rovnako veľkou zložkou detekcia popu. V tejto piesni sa nachádzajú výrazné úseky hry na piano a sláčikových nástrojov (hlavne na úvode a v závere). O značnej prítomnosti country sa dá hypotetizovať, že je podmienená práve prítomnosťou nástrojov používaných najmä pri klasickej hudbe.

Z definície v kapitole 2.2, je možné určiť prítomnosť všetkých typických nástrojov pre tento žáner až na banjo, čo môže práve pri neprítomnosti hlasovej stopy spôsobiť zámenu pri klasifikácii klasickej hudby za country.

Kapitola 9

Preferencie počúvajúceho a Odporúčanie piesní

Okrem návrhu a implementácie modelu na kvalifikáciu hudobných žánrov sa práca venuje aj návrhu predikcii hudby pre užívateľa. V nasledovnej kapitole je prezentovaný návrh, čo by mal daný systém obsahovať a ako by bolo možné dosiahnuť jeho implementácie.

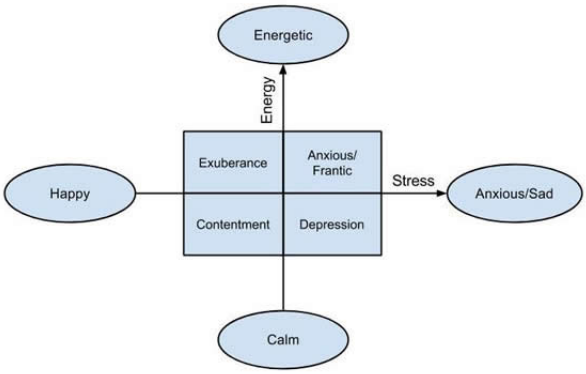
Ako bolo spomenuté v 5, využitie neurónových sietí v odvetví hudobného priemyslu je viac smerované na poskytnutie nových skladieb poslucháčovi na základe jeho preferencií. Je však možné uvažovať, že samotné preferencie poslucháča sa taktiež odvíjajú od konkrétneho žánru. V takom prípade je možné na túto problematiku naviazať práve s použitím rozpoznávania jednotlivých žánrov. Ako bolo spomenuté v 5.2, pri riešení tejto problematiky je taktiež prítomná extrakcia príznakov naviazaných na konkrétne žánre.

Hlavným rozdielom je však iný typ problému, ktorý model neurónovej siete rieši. Zatiaľ čo sa implementovaný model zaoberá správnou klasifikáciou do viacerých tried s jednou označenou triedou, je pri preferenciách poslucháča označiť týchto tried viac. Model určený pre takúto klasifikáciu potrebuje okrem správnej klasifikácie žánrov pracovať aj s inými mapami príznakov. Pri obdržaní dávky vstupných dát by mal byť model schopný okrem klasifikovania napr. hernej techniky, alebo špecifického frekvenčného pásma vytvárať aj špecifické asociácie medzi nimi, čím by sa mu otvorili nové možnosti vnímania, ako napríklad rozpoznávanie jednotlivých nástrojov. Z takýchto novo nadobudnutých informácií by sa potencionálne mohlo umožniť okrem získavania dovtedy nepoznaných vedomostí vytvoriť pre model novú úroveň chápania medzi už predtým naučenými asociáciami a tak si dopomôcť k presnejšej klasifikácii.

9.1 Klasifikácia nálady v skladbách

Nový rozmer pre túto problematiku môže pridať tradičný model nálady od Roberta Thayera. Podľa neho je možné rozdeliť skladby do 8 nálad rozdelených pomocou osí energie a stresu. Na základe uvedeného modelu je možné klasifikovať skladby do 4 tried vytvorených z extrémov týchto osí (energetickosť - kľudnosť, radosť - smútok) a 4 tried vytvorených ich prienikom pre konkrétny kvandrant (bujarosť, úzkosť, spokojnosť a depresia). Pri takejto kategorizácii skladieb sa pomocou akustickej analýzy neprizerá na konkrétne hudobné praktiky, ale pracuje sa skôr s príznakmi zvukového signálu ako dynamický rozsah zvukového signálu alebo jeho tempo, teda príznakmi ktoré sa konvolučné filtre jednotlivých vrstiev CNN nemôžu naučiť, napr. z dôvodu hľadania invariantných vzorov, ako bolo popísané

v kapitole 4.2.1 [16] Pre takúto klasifikáciu by však bolo možné použiť dáta extrahované do súborov CSV, ktoré využíva väčšina datasetov, ako bolo popísané v kaptiole 2.3.



Obr. 9.1: Rozdelenie nálady podľa R. Thayera. Zdroj:[16]

9.2 Návrh implementácie získania preferencií

Základnou metrikou pri zisťovaní preferencií počúvajúceho môžu byť váhy v intervale $< -1, 1 >$. Užívateľ dostane na začiatku pool náhodne vybraných ukážok z vytvoreného trénovacieho datasetu skladieb, pri ktorých bude musieť určiť jednu zo 4 možností na základe jeho preferencií: skladba sa páči, skôr páči, skôr nepáči, alebo nepáči. Tieto hodnotenia budú predstavovať váhové ohodnotenia 1, 0.5, -0.5 a -1. Ako aktivačnú funkciu je vhodné pre tento prípad použiť hyperbolický tangens opísaný v 3.5. Na základe ohodnotení bude model schopný detegovať, ktoré žánre sa inkriminovanej osobe páčia, prípadne nájsť rôzne vzory medzi skladbami, na základe ktorých bude vedieť odporúčať nové, nepočuté skladby.

Pri takomto ohodnocovaní skladieb je dôležité zachovať vyváženosť prítomnosti jednotlivých žánrov vrámci vybraného poolu skladieb (ale aj trénovacieho datasetu) z dôvodu zachovania objektivity a teda eliminovania prípadov, v ktorých by mohlo dôjsť k prehliadnutiu konkrétneho žánru. Preto by mali byť skladby do ponúknutého poolu vybrané náhodne a v rovnakom počte pre každú triedu obsiahnutú v datasete. Takéto zloženie by malo vytvoriť rovnomerné rozloženie výskytu jednotlivých tried a teda zaručiť objektivitu vytvoreného poolu.

Pre potreby vytvorenia validného profilu používateľských preferencíi je preto vhodné využiť metódy preferenčného učenia. Tým sa rozumie vytvorenie takého modelu, ktorý bude schopný sa naučiť z kontextu vyplývajúce preferencie. Tieto preferencie môže model následne využiť pri procese zbierania vhodných alternatív prepojených s pôvodným vstupom pomocou množiny spoločných čŕt.

Ako je popísané v [11], jednotlivé hudobné žánre majú síce podobný dynamický rozsah, no v určitých frekvenčných pásmach sú špecifické. Podobne ako model implementovaný v tejto práci, by mal byť navrhovaný preferenčný model schopný získavať okrem pravdepodobnostnej klasifikácie žánrov aj informácie o prvkoch priamo nezávislých na generalizovaných triedach jednotlivých žánrov. Príkladom takéhoto správneho učenia by mohlo byť rozpoznávanie konkrétnych hudobných nástrojov v skladbách, ako napríklad hranie na saxofón. Na základe učenia by teda mohlo byť predpovedané s akou pravdepodobnosťou sa bude poslucháčovi páčiť jeho prítomnosť v popových skladbách.

Pri preferenčnom učení je potrebné získať od užívateľa dostatočne veľkú množinu správnych výsledkov, z ktorých bude vedieť model poskladať množiny schopné uspokojiť preferencie poslucháča, pričom je zároveň potrebné sledovať ich vývoj, aby bolo možné v čase udržiavať toto poskytovanie na konzistentnej úrovni jeho uspokojenia. Po poskytnutí preferencií je potrebné získať množiny spoločných čŕt obsiahnutých v užívateľom novo vybraných skladbách pre presnejšie odporúčania. Takýmito znakmi sa myslia konkrétne štruktúry, ktoré je schopný model vyextrahovať ako samostatné príznaky triedy spomínané v kapitole 9.2. Samostatným určením žánru a tempa bude totiž profil užívateľa nedostatočný a teda nebude možné presne určiť napr. konkrétny preferovaný podžáner.

Kapitola 10

Záver

Cieľom práce bolo navrhnutie a implementovanie modelu neurónovej siete schopného klasifikovať jednotlivé hudobné žánre uvedené v datasete GTZAN. Z prečítaných internetových fór a stránok zaoberajúcimi sa učením neurónových sietí zameraných na extrakciu príznakov z hudby vyplýva buď použitie datasetu GTZAN, alebo vytvorenie vlastného. Autorský zákon vytvára pri riešení práci s takýmto obsahom problém, ktorý sa týka jeho možného šírenia. To nie je možné zabezpečiť tak, aby obsahované autorské diela neboli zneužité na iné účely.

Hlavným problémom pri tvorbe tejto práce teda bol nedostatok vstupných dát pre trénovanie z dôvodu autorských práv, preto boli získané ukážky obsiahnuté v datasete segmentované na malé časti v kombinácii s k-násobnou validáciou, čo dopomohlo k lepšiemu učeniu aj presnosti navrhovanej siete.

Vytvorená sieť by mala byť schopná rozpoznať príznaky uvedených žánrov z datasetu a na základe nadobudnutých informácií percentuálne určiť, do ktorých žánrov a akým podielom sa s nimi poskytnutá nahrávka zhoduje na úrovni špecifických čŕt získaných počas trénovania siete.

Pre pochopenie a návrh neurónovej siete bolo potrebné preštudovať problematiku neurónových sietí, ich súčastí a hlavne prvky konvolučných neurónových sietí, na ktorých je výsledný model založený. Ako predloha pre výsledný model slúžil článok [21]. Tento model bol značne upravený, aby boli minimalizované klasifikačné problémy spomínané v 8. Pre vytváranie podkladov pre učenie modelu a analýzu vkladaných nahrávok bolo potrebné naštudovať tvorbu spektogramov pomocou knižnici `Librosa`, čo dokázalo značne urýchliť vytváranie spektogramových podkladov.

Okrem návrhu modelu neurónovej siete bola preštudovaná problematika aktuálne existujúcich a najpoužívanejších riešení v rámci odboru získavania informácií o hudbe, ktoré sú využívané na vytváranie personalizovaných playlistov pre používateľa. V kapitole 9 sa uvádza návrh, ako by mohla vypadať takto vytváraná sieť na predikciu hudobného vkusu.

Možným rozšírením práce by mohla byť integrácia s API niektorej zo spomínaných streamingových služieb v 5 na zlepšenie trénovacích dát.

Pri testovaní sa nakoniec ukázalo, že navrhnutá sieť nedokáže správne určiť žáner, pokiaľ závislosť týchto žánrov na ich predkoch, resp. ich potomkoch, je veľká do miery preberania najzákladnejších čŕt, ktoré sa snaží model generalizovať. Tento problém bude pravdepodobne spôsobený aj vekom knižnice, čo bolo dokázané na testoch v kapitole 8.3, ktorá opisuje zmenu vnímania jednotlivých žánrov za posledné obdobie.

Literatúra

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Dostupné z: https://www.tensorflow.org/.

[2] Brownlee, J. *Overfitting and Underfitting With Machine Learning Algorithms*. Aug 2019. Dostupné z: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/.

[3] Brownlee, J. *How do convolutional layers work in Deep Learning Neural Networks?* Apr 2020. Dostupné z: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/.

[4] Chandra, A. L. *McCulloch-Pitts Neuron - mankind's first mathematical model of a biological neuron*. Towards Data Science, Nov 2018. Dostupné z: https://towardsdatascience.com/mcculloch-pitts-model-5fdf65ac5dd1.

[5] Chollet, F. et al. *Keras*. 2. vyd. 2015. Dostupné z: https://keras.io.

[6] Chollet, F. *Deep learning v jazyku Python : knihovny Keras, Tensorflow*. 1. Praha: Grada Publishing, 2019. ISBN 978-80-247-3100-1.

[7] Defferrard, M., Benzi, K., Vandergheynst, P. a Bresson, X. FMA: A Dataset for Music Analysis. In: *18th International Society for Music Information Retrieval Conference (ISMIR)*. 2017. Dostupné z: https://arxiv.org/abs/1612.01840.

[8] Education, I. C. *What are convolutional neural networks?* Oct 2020. Dostupné z: https://www.ibm.com/cloud/learn/convolutional-neural-networks.

[9] Ezra, K. *Dense layer in tensorflow*. OpenGenus IQ: Computing Expertise &; Legacy, Oct 2021. Dostupné z: https://iq.opengenus.org/dense-layer-in-tensorflow/.

[10] Inc., A. *Discover built-in sound classification in soundanalysis - WWDC21 - videos*. 2021. Dostupné z: https://developer.apple.com/videos/play/wwdc2021/10036/.

[11] Kirchberger, M. a Russo, F. A. Dynamic Range Across Music Genres and the Perception of Dynamic Compression in Hearing-Impaired Listeners. *Trends in Hearing*. 2016, zv. 20, s. 2331216516630549. DOI: 10.1177/2331216516630549. PMID: 26868955. Dostupné z: https://doi.org/10.1177/2331216516630549.

[12] Krčmová, M. *6.4 Percepce a porozumění řeči*. 2007. Dostupné z: https://is.muni.cz/elportal/estud/ff/js07/fonetika/materialy/ch06s04.html.

[13] Medalová, K. *Neurón a jeho stavba*. 2015. Dostupné z: https://www.mentem.sk/blog/neuron/.

[14] Mukanga, E. T. A neural networks adoption framework for predicting stock market trends : case of the Zimbabwe Stock Exchange. *Business & Social Sciences Journal*. 2017, zv. 2, č. 2, s. 27–60. DOI: 10.10520/EJC-933fc4ffb. Dostupné z: https://journals.co.za/doi/abs/10.10520/EJC-933fc4ffb.

[15] Muráň, I. J. *Algoritmy strojového učenia I. - Učenie s učiteľom*. Jul 2020. Dostupné z: https://umelainteligencia.sk/algoritmy-strojoveho-ucenia/.

[16] NUZZOLO, M. *Music mood classification*. 2015. Dostupné z: https://sites.tufts.edu/eeseniordesignhandbook/2015/music-mood-classification/.

[17] RPubs. *Spotify Project-Predicting Genre for better User-Recommendation*. 2021. Dostupné z: https://rpubs.com/puneet1193/832604.

[18] Team, K. *Keras documentation: Dense layer*. 2022. Dostupné z: https://keras.io/api/layers/core_layers/dense/.

[19] Thakur, R. *Step by step VGG16 implementation in Keras for beginners*. Towards Data Science, Nov 2020. Dostupné z: https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c.

[20] Tzanetakis, G., Essl, G. a Cook, P. *Automatic Musical Genre Classification Of Audio Signals*. The International Society for Music Information Retrieval, 2001. Dostupné z: http://ismir2001.ismir.net/pdf/tzanetakis.pdf.

[21] Vaidya, K. *Music genre recognition using convolutional neural networks (CNN)part 1*. Towards Data Science, Dec 2020. Dostupné z: https://towardsdatascience.com/music-genre-recognition-using-convolutional-neural-networks-cnn-part-1-212c6b93da76.

[22] Wang, A. L. An industrial-strength audio search algorithm. In: *ISMIR 2003, 4th Symposium Conference on Music Information Retrieval*. 2003, s. 7–13. ISBN 097461940X.

[23] Wikipedia. *Chord progression — Wikipedia, The Free Encyclopedia*. 2021. [Online; navštívené 20.12.2021]. Dostupné z: http://en.wikipedia.org/w/index.php?title=Chord%20progression&oldid=1060656655.

[24] Wikipedia. *Chroma feature — Wikipedia, The Free Encyclopedia*. 2021. [Online; navštívené 20.12.2021]. Dostupné z: http://en.wikipedia.org/w/index.php?title=Chroma%20feature&oldid=1011214694.

[25] Wikipedia. *Blues — Wikipedia, The Free Encyclopedia*. 2022. [Online; navštívené 06.05.2022]. Dostupné z: http://en.wikipedia.org/w/index.php?title=Blues&oldid=1083207781.

[26] Wikipedia. *Classical music — Wikipedia, The Free Encyclopedia*. 2022. [Online; navštívené 06.05.2022]. Dostupné z: http://en.wikipedia.org/w/index.php?title=Classical%20music&oldid=1086766452.

[27] Wikipedia. *Country — Wikipedia, The Free Encyclopedia*. 2022. [Online; navštívené 06.05.2022]. Dostupné z: http://sk.wikipedia.org/w/index.php?title=Country&oldid=7335392.

[28] Wikipedia. *Disco — Wikipedia, The Free Encyclopedia*. 2022. [Online; navštívené 06.05.2022]. Dostupné z: http://en.wikipedia.org/w/index.php?title=Disco&oldid=1086711846.

[29] Wikipedia. *Džez — Wikipedia, The Free Encyclopedia*. 2022. [Online; navštívené 06.05.2022]. Dostupné z: http://sk.wikipedia.org/w/index.php?title=D%C5%BEez&oldid=7230886.

[30] Wikipedia. *Heavy metal — Wikipedia, The Free Encyclopedia*. 2022. [Online; navštívené 06.05.2022]. Dostupné z: http://sk.wikipedia.org/w/index.php?title=Heavy%20metal&oldid=7300774.

[31] Wikipedia. *Hip-hop (hudba) — Wikipedia, The Free Encyclopedia*. 2022. [Online; navštívené 06.05.2022]. Dostupné z: http://sk.wikipedia.org/w/index.php?title=Hip-hop%20(hudba)&oldid=7101759.

[32] Wikipedia. *Neurón — Wikipedia, The Free Encyclopedia*. 2022. Dostupné z: http://sk.wikipedia.org/w/index.php?title=Neur%C3%B3n&oldid=7345982.

[33] Wikipedia. *Pop (hudobný žáner) — Wikipedia, The Free Encyclopedia*. 2022. [Online; navštívené 06.05.2022]. Dostupné z: http://sk.wikipedia.org/w/index.php?title=Pop%20(hudobn%C3%BD%20%C5%BE%C3%A1ner)&oldid=7027755.

[34] Wikipedia. *Reggae — Wikipedia, The Free Encyclopedia*. 2022. [Online; navštívené 06.05.2022]. Dostupné z: http://sk.wikipedia.org/w/index.php?title=Reggae&oldid=7200119.

[35] Wikipedia. *Rock — Wikipedia, The Free Encyclopedia*. 2022. [Online; navštívené 06.05.2022]. Dostupné z: http://sk.wikipedia.org/w/index.php?title=Rock&oldid=7333673.

[36] Yani, M., Irawan, S. a Setianingsih, C. Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail. *Journal of Physics: Conference Series*. 1. vyd. Máj 2019, zv. 1201, s. 012052. DOI: 10.1088/1742-6596/1201/1/012052.

[37] Zbořil, F. *Soft computing* [online]. 2020 [cit. 2022-05-08]. Dostupné z: https://www.fit.vutbr.cz/study/courses/SFC/private/20sfc_1.pdf.

Príloha A

Obsah priloženého CD

- **README.md** Užívateľská príučka obsahujúca pokyny na spustenie programu.
- **xpalen05.pdf** Elektronická podoba odovzdávanej bakalárskej práce.
- Adresár **bp_source** obsahujúci zdrojové súbory pre generovanie textu práce.
- Adresár **source** obsahujúci zdrojové kódy spolu s potrebnou adresárovou štruktúrou .

Dataset a model nie sú na médiu prítomné z kapacitných dôvodov. Odkaz na ich získanie je uvedený v príručke.