



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS
INFORMATIKAI KAR



SZAKDOLGOZAT

OE-NIK
2020.

Hallgató neve:
Hallgató törzskönyvi száma:

Hörömpő András
T/005125/FI12904/N

Tartalom

1.	A megoldandó feladat megfogalmazása. Bevezetés.....	2
2.	Irodalomkutatás és módszertanvizsgálat	4
2.1	Hasonló megoldások	4
2.2	A felismerni kívánt zenei stílusokról	11
2.3	Egy hasonló, népszerű alkalmazás – Shazam	15
2.4	Gépi tanulás, neurális hálózatok, a konvolúciós neurális hálózatokról	18
2.5	Kész, működő, bizonyítottan pontos konvolúciós hálózatok	23
3.	Feladat és specifikáció.....	29
3.1	A tanuláshoz vett adatok. Mi használható, mi nem használható?.....	29
3.2	Zenéből kiolvasható és megszerezhető tulajdonságok, azok elemzése	29
3.3	Specifikáció	39
4.	Tervezés	41
5.	A fejlesztés folyamata	45
6.	Tesztelési dokumentáció	64
7.	Felhasználói dokumentáció.....	72
8.	Kutatási-fejlesztési lehetőségek	80
9.	Összefoglalás	81
10.	Summary	84
	Ábrajegyzék	87
	Irodalomjegyzék	91

1. A megoldandó feladat megfogalmazása. Bevezetés

A mai világban, szinte kivétel nélkül mindenki, de sokan szabadidős tevékenységként szívesen hallgatnak különbnél különb zenétet. Legyen az hagyományos rádión keresztül a kedvenc csatornán, legyen az egy népszerű zenestreamelő felületen keresztül (mint például Spotify-on, Deezer-en, vagy iTunes-on) online vagy offline változatban, vagy éppen amit a videómegosztó oldalak felkínálnak meghallgatásra. Sokan hallgatnak mindenféle zenét, ténylegesen, amit épp a rádió ad, azok számítanak „mai slágereknek”. Ellenben nem mindenkinek tetszhet egy adott zeneszám, nem biztos, hogy találhatóak benne olyan elemek, olyan „hozzávalók”, amelyek elnyerik az ő zenei ízlésének tetszését, viszont másoknak lehet, hogy pont az a zeneszám jelent nagyon sok mindent, legyen az köszönhető a zeneszámban fellelhető mondanivalónak, dalszövegnek, vagy éppenséggel a zenében megtalálható, harmonikus zenei összjátéknak, ami érzéseket, tetszést fejez ki hallgatójában.

Gyakran elhangzó kérdés szokott lenni, főleg egy új társaság megismerése közben, hogy „mégis milyen zenét kedvelsz?”, amire adott válaszok annyi félek lehetnek, ahány embertől ez meg van kérdezve. A több kimenetel közül kiemelendő néhány. Van, aki egyszerűen „én mindenfajta zenét szeretek” módon reagál, amiből lehet következtetni, hogy rádiót, avagy a különféle zenei szolgáltatók véletlen módját használja. Van, aki kijelenti, hogy „én x és y stílust szeretem, mert...” mondattal nyit, és kifejti, hogy adott stílust milyen oknál fogva szereti, esetleg megnevezi egy-két kedvenc, adott stílusban igen sokat alkotó előadóját. A kettő közt szokott lenni az a lehetőség, hogy „nincs kimondottan kedvenc stílusa, de szereti az olyan, zenétet, amiben van...” kezdetű reakció, és ezt követően egy olyan, felsorolásjellegű leírást ad, amelyekből akár ki lehet következtetni, hogy a személynek mi vagy mi lehet az a zenestílus, amely igazán az ínyére való, holott ő maga még nem tudja ezt pontosan, csak azt sejti, vagy nem követi annyira a zenei világban zajló stílusbeli jegyek alakulását.

Az alkalmazás elkészítésének célja, hogy ezen bizonytalanságon áthidalva, egy adott zenét, akár kedvenc zenét betöltve meghatározza, hogy milyen zenei stílus jegyeinek felel meg legvalószínűbben ez a zene, mi a zene legvalószínűbb zenei stílusa, és mi lehet akár ezen, stílusban bizonytalan személy számára az a stílus, ami tetszhet számára. Az alkalmazás elkészülését gépi tanulás fogja segíteni, amihez különböző, a zenékben fellelhető tulajdonságot fogja alapul venni.

A fentebb említett, „a stílus konkrét meghatározásában bizonytalan személy” is minden bizonnyal fel tud sorolni néhány olyan dolgot, amelyek közösek lehetnek ilyen stílusokban. Ilyen például a hangszerezettség. Meg tudja mondani azt, hogy egy adott zenén belül milyen hosszan hall benne hangszeri aláfestést, illetve milyen hangszereket vél felfedezni a dallamok között. A zenétet nem csak a hangszerek tehetik kész zenévé, ami alól, noha van kivétel, de nem ez a jellemző tendencia. A hangszerek többnyire csak aláfestések az énekhangnak, avagy vokálnak, ami egy vagy több előadó, ritmusos, rímes, hanglejtéses, különböző hangmagasságon énekelt szövege. Az énekelt dalszövegek és

azok hangulatbeli jelentésétől elvonatkoztatva (mivel, hasonlóan az irodalmi költők költeményeihez, sokféle, különböző jelentéssel bírnak) szerepe lehet, hogy a teljes zeneszám hosszához viszonyítva milyen hosszsan lehet hallani énekhangot. Többek között, ami még jelzésértékkel bírhat egy zene stílusának meghatározásában, az nem más, mint a zenében fellelhető tempó, amelyet angolul és más környezetben BPM-nek, azaz beats-per-minute-nek (ütem-per-perc) hívnak. Szemléletesen a BPM úgy kapható meg, hogy hányszor vagyunk arra képesek, hogy kezünkkel vagy lábunkkal dobantsunk, koppintsunk, azaz hány „beat” kiadására vagyunk képesek azon zene meghallgatásának hatására. Persze nem elvárás, hogy valaki megmondja, hogy ő „kifejezetten a 120 és 140 BPM közötti zenét szereti”, de annyit elmondhat, hogy azok a zenék lassúnak tűnnek vagy épp gyorsnak.

Ezen információk megszerzését követően felállításra kerül egy neurális hálózat, amely a kapott információkból (többnyire számértékekből) képes meghatározni, hogy adott zene milyen stílusba tartozik bele. A tanításhoz a tanítás sikerességét elősegítve megadásra kerül, hogy adott zene milyen stílusba is tartozik bele. A tanítás eredménye szintúgy számérték lesz, ami azt fogja megmondani, hogy mekkora eséllyel tartozik az adott zene a betanított stílusokhoz. A cél az, hogy az elérhető legnagyobb pontossággal azt a stílust adja eredményül, amely ténylegesen hozzá tartozik. Természetesen előfordulhat, hogy egy zene több stílus jegyeit is magán hordozza, ekkor adja vissza ezek közül a legvalószínűbbeket, tehát amelyeknek a legnagyobbak az erre szolgáló mutatói.

2. Irodalomkutatás és módszertanvizsgálat

Mielőtt a tényleges alkalmazásfejlesztés elkezdődne, a hasonló megoldások áttekintése érdekében irodalomkutatás történik, amelyek eredményei az alábbiakban olvashatók.

2.1 Hasonló megoldások

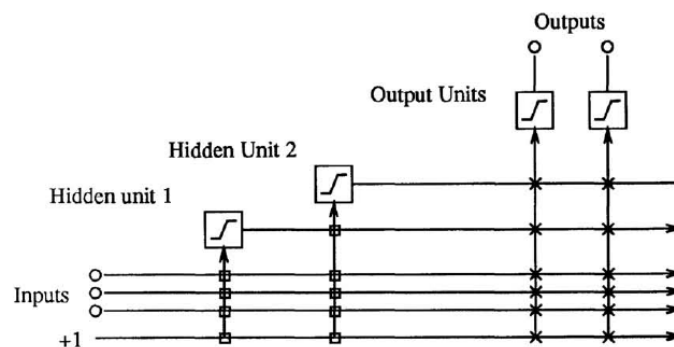
A zenei stílusok felismertetése egy igencsak régi problémának nevezhető, számos projekt készült ezzel kapcsolatban. Minden egyes projekt saját-saját utat jár be, amelynek során minél sikeresebben próbálja meg egy adott zene legvalószínűbb stílusát felismertetni.

Talán az első, konkrét zenei stílus felismertetésére vonatkozó irányulás 1997-ben történt, a Carnegie Mellon Egyetem munkatársai által. [1] Az ő problémafelfogásuk szerint akkoriban még igencsak bonyolultnak bizonyult még annak a felismerése is, hogy egy szöveget fel lehessen ismertetni gépi tanulás segítségével. Akkor négy stílus meghatározását próbálták meg elérni klasszifikáció segítségével. Ezen stílusok a dalszöveges („lyrical”), az őrjöngő („frantic”), a szinkópás („syncopated”) és pointillista („pointillistic”). Ezen stílusok a fejlesztés során kiegészültek a blues, az idézetes („quote”), a magas („high”) és alacsony („low”) hangszínű zenék felismertetésével. A cél az volt, hogy improvizálva egyet a fő 4 csoportból, és egy zenét bejátszva valamelyikből, amelyről maga az ember tudja a jellemző stílusát (pl. szinkópás), „mi annak a valószínűsége, hogy a gép is ugyanezt fogja stílusra mondani.” Minderre 5 másodperc volt megengedve. Akkoriban az mp3 tömörítés még nem volt igazán elterjedt, MIDI zenékre volt kitalálva mindez. Az adatok 15 másodperces mintákból lettek vételezve úgy, hogy ezen 15 másodperces részekből 5-5 másodperces részek lettek csúszóablakosan kiválasztva az ötödik másodperctől kezdve (hogy lehessen érzékelni emberi felfogással a stílusváltást). Mindösszesen 1200 darab 5 másodperces tanító részlet jött létre. Minden ilyen kis MIDI darabra „13 darab, alacsony szintű jellemző lett meghatározva, mint például MIDI kulcszám, hangerő, vagy hangerőváltozások”. A klasszifikációra háromfajta megközelítés volt elképzelve. Az első Bayes-klasszifikációt alkalmazott. A Bayes-klasszifikáció meghatározza a legvalószínűbb osztályt, amelybe beletartozhat az adott dallam. Ehhez a módszerhez viszont figyelembe kellett azt venni, hogy az „egyes tulajdonságok korrelálatlanok és normalizáltak legyenek”. A legvalószínűbb osztály meghatározása matematikai úton történik, miszerint melyik osztályhoz tartozó normalizált távolság a legkisebb. A tanulmányban az alábbiak szerint definiálták: „Normalizált távolság alatt Euklideszi-távolság van értve, amely meghatározható a

$$\Delta_C = \sqrt{\sum_{i=1}^n \left(\frac{F_i - \mu_{C,i}}{\sigma_{C,i}} \right)^2}$$

képlettel, ahol C a felismerni kívánt osztály, i jelenti az egyes jellemzőket a 13 közül, $\mu_{C,i}$ az egyes átlagokat, míg $\sigma_{C,i}$ az egyes szórásokat jelölik.” [1] A másik megközelítés lineáris klasszifikációt használ. Lineáris klasszifikáció alatt az egyes jellemzők súlyozott összegét veszi, ahol „az egyes osztályok esetén más-más súlyértékeket” állítottak be. Ezen esetben az algoritmus normál eloszlás szerint képzelel el a változókat. Már eleve kétséggel vették a módszert magát, a készítők megállapítása

szerint is „az elkülönítés lehet vagy épp nem lehet túl sikeres”. A harmadik, és egyben legerősebb próbálkozás neurális hálózatokra épült, ami nem vette figyelembe a „nemlineárisan összefüggő komponenseket és nem határozott meg erős feltételeket a jellemzőkkel, azok valószínűségeinek eloszlásával kapcsolatban”. Technikáját illetően úgynevezett „Cascade-Correlation” architektúrát használt, ami kezdetben két ismert adattal rendelkezett, egy bemenő és egy kimenő egységgel. Tanulás közben került kiegészítésre a hálózat rejtett rétegekkel, amik ezen bemenetekhez és kimenetekhez kapcsolódtak, majd ezen, belső rétegek is össze lettek egymással kapcsolva több szint mélységben. Minden réteg „nem-lineáris szigmoid függvényt használt, aminek segítségével súlyozott átlagot volt képes számolni a bemenetre”. [1] Kimenetként minden osztályra adott egyet-egyét: egy igaz-hamis értéket, hogy adott osztályhoz tartozhat-e azon minta vagy sem.



1. ábra „Cascade-Correlation” architektúra szemléletes ábrán, ahol két rejtett réteg kerül hozzáadásra a rendszerhez [2]

A mintákat 80%-20% arányban osztották el úgy, hogy a 80%-át különítették el tanításhoz, míg a fennmaradó hányadát illesztésre a kapott eredmény alapján. Minden részről pontos eredmény született, mind 4, mind 8 osztályra vonatkozóan. [1] Érdekes megfigyelés volt számukra, hogy míg a bayesi és a lineáris regressziós megoldás felépítése és kiszámítása csupán másodperceket vett igénybe, ugyanazon, ugyanannyi elemet tartalmazó tanítóhalmazra viszont a neurális háló több órán keresztül tanult. A pontosságot a nyolccsoportos klasszifikáció során a „quote” osztály rontotta, mivel ekkor ennek a stílusnak nem voltak még teljesen, 100%-osan az ismeretében, és nem is volt még sok, ahhoz hasonló dallam.

Number of Classes	Bayesian	Linear	Neural Network
4	98.1	99.4	98.5
8	90.0	84.3	77.0

1. táblázat Az 1997-es tanulmány és kutatás eredménye 4-, illetve 8-osztályos klasszifikáció esetén [1]

Egy egyévvvel későbbi írás különböző megoldást használ az előbb említettektől. Ezen, Karlsruhei Egyetemen végzett kutatás úgynevezett ETM-NN (Explicit Time Modelling with Neural Network, lefordítva neurális hálóval segített explicit időmodellezés)

technikára épít, aminek alapja, hogy „a különböző, hálóban rejlő rejtett rétegek a különböző absztrakciós szinten lévő akusztikai rájátszásokat képesek megkülönböztetni egymástól, ezáltal a zenei stílusok elkülönítésében is jelentős szerepet játszhat”. [3] Hasonlóan, itt is 4 zenei stílus felismertetése áll a középpontban. Itt a rock, a pop, a techno és a klasszikus zene kerül felismertetésre. Az említett ETM-NN modell egy régebbi modellezési típust, a HMM (Hidden Markov Model – rejtett Markov modell) módszernél bizonyították jobbnak. Alapvető elképzelés volt, hogy „a zenestílus felismerőnek akusztikus jelek időbeli felépítésével kell megbirkóznia”, a problémája a HMM rendszernek, amely ezen jeleket előállította, az volt, hogy „gyenge diszkriminatív erővel rendelkezett”. Ugyanakkor neurális hálózatokkal felépíthető ezen jelsorozat úgy, hogy legalább ugyanakkora hatékonysággal rendelkezzen, mint a korábbi megoldások. A neurális hálózatok pedig abban nyújtanak ebben segítséget, hogy az idő szerinti információkat, amelyek „kontextus egységben vannak eltárolva”, dolgozza fel, tanulja meg annak szerkezetét, hogy a bemeneti jelet hatékonyabban tudja feldolgozni. Az ETM-NN modell szerint az időbeli felépítésben rejlő „statisztikai vizsgálat új funkciókat biztosít a teljes hálózatnak”.

Ahogy már említve volt, a lényeg itt a különböző zeneakusztikus elemek felismerése, ami lényegében információkinyerést jelent az időbeli szerkezetből, azaz a zenéből. [3] Ezt a bemeneti „zenei jelet fel kell bontani különböző akusztikus elemek szekvenciájára, amelyből karakterisztikus vektor állítható elő”, és ezt egy háromrétegű neurális hálózatnak át lehet adni. Ez közvetetten azt eredményezte, hogy a megfelelő zenestílust megállapította, ugyanakkor „a különböző absztrakt akusztikus elemek megtanulása a rejtett rétegek által” tette ezt lehetővé, amely az elsődleges követelmény volt a rendszerrel szemben.

„A különböző beszédbeli elemek felismertetése fonémák formájában történik”, ami nem más, mint a nyelv egy kicsi, elvont egysége, több beszédhang összességét jelöli, ami zenék esetében az esetek nagy részében nehezebben működik, hiszen előfordulhatnak kiegészítő hangok vagy más, egyéb, effektként hangzó elemek is. A felismeréshez szükséges volt a zenéből egy karakterisztikus vektor felállítása, amely a hálózatot ellátta a szükséges információkkal. „Az információk megállapításához ebben a rendszerben egy tízegységes, rejtett rétegen keresztül történik az elemzés a kimenet segítségével, ami a tanítás után elhagyottá válik”. [3] Egy absztrakt elem, jelölve e_i -vel akkor jelenik meg, ha az i -edik rejtett egységben a legnagyobb aktivációs értékkel rendelkezik minden rejtett egységre vonatkozóan, és minimális hossza 0.4 másodperc. Egy ilyen zenében több absztrakt, kiemelkedő elem is létezhet, emiatt a vektoros felépítés. A versenyhelyzet elkerülése érdekében kikötés továbbá, hogy egyszerre csak egy rejtett réteg aktív. Az így kapott különböző fajta „eseményeket” lehet osztályozni olyan módon, ezáltal megkapni az időbeli felépítést, ami egymagában jelenik meg (e_i), másodmagával jelenik meg (e_i, e_j), harmadmagával fordul elő (e_i, e_j, e_k), milyen hosszan zajlik maga e_i esemény, illetve mi az összes elem átlaga, maximuma és varianciája, a megfelelő értékeket normalizált formában feltüntetve. A nagymennyiségű adat miatt redukciós folyamatot alkalmazva

csak azon adatok lettek meghagyva a vektorban, amely által a lehető legjobban meg lehet különböztetni a stílusokat, azaz „rendelkezik a lehető legnagyobb diszkriminatív erővel”.

A tanító adatbázis olyan módon épül fel, hogy mind a 4 stílushoz egyenként 3 órányi anyag lett feldolgozva, 360 minta lett vételezve, kb. 30 másodperces részekkel, minden típusból egyenlő arányban, sztereo (kétcsatornás) hang helyett monó (egycsatornás) hangformátumban. A tanító-validáló-tesztelő arány 67%-13%-20% formában került meghatározásra. Előfeldolgozásként a különböző hangszerekre készített egy szűrést, például zongora elsősorban klasszikus zenében fordul elő, techno stílusban kevésbé. Ehhez meghatározásra került két komponens, az adott hangszer alaphangja („fundamental tone”) és rezonancia karakterisztikája („timbre”), ez utóbbi „összefüggésbe hozható a felhangokra vonatkozó energiára”. [3] Mind a felváltani kívánt HMM rendszerre, mind a bevezetett ETM-NN rendszerre elkészült a tanítás, melynek eredményei a 2. táblázatban találhatók.

System	recognition rate
HMM with: left-to-right transitions	79.2%
ergodic transitions	76.4%
ETM-NN with: unigrams, duration, activation	81.9%
+ bigrams	84.7%
+ trigrams	86.1%

2. táblázat A különböző HMM és az ETM-NN módszerek felismerési arányai [3]

Az eredmények azt mutatják, hogy már önmagában, kevés információ megtartásával az ETM-NN módszer hatékonyságát tekintve jobbnak bizonyult, mint a HMM bármely változata. Bizonyításra került továbbá, hogy minél több adattal történt tanítás, figyelembe véve a „bigramokat” és „trigramokat”, úgy az eredmények még pontosabbnak bizonyultak.

s/r ¹	Rock	Pop	Techno	Classic
Rock	72.2%	27.3%	0%	0%
Pop	11.1%	83.3%	5.6%	0%
Techno	0%	0%	94.4%	5.6%
Classic	0%	5.6%	0%	94.4%

3. táblázat A felismerések közti konfúziós mátrix (azaz mekkora valószínűséggel következett be az, hogy adott stílus helyett egy másik stílus került felismerésre) [3]

A 3. táblázat szerinti konfúziós mátrix alapján megállapításra kerülhetett, melyek voltak azok a zenék, amelyeket a vektor előállításából pontosabban volt képes felismerni, és melyeket nem, továbbá amelyeket nem tudta felismerni, úgy mely más stílusra tett predikciót. Megállapítható, hogy a rockzene nehezebben felismerhető kategóriának bizonyult, gyakran tévesztve a popzenével. Ugyanakkor a techno és a klasszikus zene közel 95%-os pontossággal volt egyértelműen meghatározható.

Ugorva egy kicsit az időben, egy 2018-as, Waterloo Egyetemen történt elemzés más oldalról, mégis számértékekre hagyatkozva épített fel zenestílus elemzőt gépi tanulásra alapozva. [4] Kétfajta megközelítést használt erre a vizsgálatot végző. Az egyik szerint egy felügyelt tanítás formájában konvolúciós neurális hálót (CNN – Convolutional Neural Network) használ fel, amelynek tanítási alapja a zenéből készített úgynevezett spektrogramok voltak, míg egy másik megoldás értelmében a zenéből a rá jellemző idő- és frekvenciatartományt figyelembe véve számadatok szerint vette alaposabban vizsgálat alá. A kettő különböző megközelítés együttesen megközelítőlegesen 90%-os eredményt ért el. A kategóriákba, stílusokba sorolást a zene ritmikus felépítésére, harmóniájára és hangszerzettségére alapozza. Az elkészítéshez szükséges adatok egyenként 10 másodpercesek, nagyjából 40 ezer darab lett kiválasztva, és olyan zenei stílusok felismerése volt feladata, mint pop, rock, hip-hop, techno, blues, vokális és reggae, nem egyenlő felosztásban, wav formátumban.

Bárminemű tanítás első lépése az adatok előkészítése, feldolgozása. Első lépés az adatelőkészítés, mégpedig figyelembe véve a jel-zaj arányát (Signal-to-Noise Ratio - SNR), amely a hasznos és zavaró jel hányadosát adja meg, decibelben kifejezve. [5] Meghatározása a jel teljesítményének és a zaj teljesítményének hányadosával határozható meg, amplitúdóértékekkel kifejezve a jelhez tartozó amplitúdóérték és a zajhoz tartozó amplitúdóérték hányadosának négyzetével egyezik meg, és igen széles tartományba képes esni. A tanulmányban ezt ellensúlyozandó, ezen tulajdonság értéke az adott zenei jelre meghatározható az „ $y(t) = x(t) - \alpha * x(t - 1)$ ” összefüggéssel, ahol $x(t)$ az eredeti jel, míg $y(t)$ egy szűrt jelre utal, az „ α értéke 0.97-re van állítva”. [4] A zenestílus felismerése egyik esetben CNN segítségével valósul meg. CNN, azaz konvolúciós neurális hálózat kiválóan alkalmazható olyan tanítás esetén, amelynél a bemenet egy multimédiás fájl, amely jelen esetben képek formájában valósul meg. A hálózat a tanítás végére képes arra, hogy egy adott képről felismeri a témáját, ki/mi található rajta, tehát képes a klasszifikációra. Zenék esetén előállítható egy azt reprezentáló kép, mégpedig egy frekvencia-idő függvényében felírt spektrogram segítségével. Ezen tanulmány úgynevezett MEL spektrogramokat használ.

Azzal, hogy képek kerültek megalkotásra a zenéből, használható a konvolúciós neurális hálók adta előny, miszerint kép szerinti klasszifikációra tökéletesek. Megfelelő elemszámú minta esetén, mint képfelismerés esetén megadható, hogy egy beadott spektrogram melyik stílus spektrogramjához áll a legközelebb, vagy kép szerint melyik hasonlít jobban rá. Minden egyes konvolúciós rész 3 feladatot lát el:

- konvolúció lépésében egy, csúszóablakos mátrix szerint történik meg egy adott kép feldolgozása
- annak érdekében, hogy egy képre vonatkozó dimenziószámok csökkenjenek (mert egy kép nem 3x3-as felbontású, hanem annál lényegesen nagyobb), „down sampling” segítségével a minták száma csökkentésre kerül olyan módon, hogy véve például egy 2x2-es, csúszóablakosan haladó mátrixot, a 4 elemből csak a maximum kerül megtartásra, rögtön negyedakkora képet kapva.

- szükség van nem-linearításra is a neurális háló hatékonyságának növelésére, mivel a konvolúciós lépés maga lineáris művelet. Ellenben egy, a gépi tanulásban gyakran használt ReLU aktivációs függvény képes ezt megoldani, és nemlineáris eredményt produkál

A modell tanítására kétféle megközelítést használ. Az egyik szerint a tanításban részt vevő, konvolúció alapú súlyok fixek, konstansok, csak az ezen kívüli súlyértékek változtathatók. Egy másik szerint egy VGG-16 néven ismert modellben használt, előre rögzített súlyértékektől indul el. Felépítését illetően az utolsó réteg fogja megadni a sejtett stílust, amihez softmax aktivációs függvényt használ, amely az egyes stílusokhoz kapcsolódó valószínűségeket adja vissza. Gyakori jelenséggé vált megfigyelhető deep learning hálózatoknál általánosságban, hogy overfitting jelenik meg, azaz túlillesztés történik a mintára, túltanulta a modell a jellemzőket a tanító halmazzal kapcsolatban, ezáltal viszont kicsi túrértékű lesz az új adatokra vonatkozóan. Ezt elkerülve két stratégia kerül bevetésre, mivel a tanulmány készítője is overfittinget tapasztalt. Az egyik, L2-regularizáció névre hallgató megteszi, hogy a veszteségfüggvényt „az $\frac{1}{2} \cdot \lambda \sum_{i=1}^n w_i^2$ összefüggésből számítja, ahol w_i az adott rétegen használt súlyértéket reprezentálja, λ értéke ebben a tanulmányban fixen 0.001”. [4] A négyzetes összefüggésből következik, hogy a kis súlyértékek kevésbé fognak számíthatni, ellenben az 1-nél nagyobbak, jobban jellemezve így a hibát, és nagyobb „büntetést” adva a rendszernek. A másik pedig a „dropout” névre hallgat, és a regularizációt akarja elérni. Lényege, hogy a „tanítás során néhány neuron kikapcsolásra kerül. A kikapcsolás olyan módon történik, hogy van választva egy érték, itt 0.3, ami azt jelenti, hogy egy neuron 0.3 valószínűséggel fog kiesni adott iteráción belül”. Az adatok 90%-5%-5% arányban kerültek felosztásra tanító-validációs-tesztelő adatokra.

Ha nem képfeldolgozási módszerrel történik a meghatározás, akkor kézzel kell adatokat előállítani az adott zenéből. Ezen jellemzők előállításához a Python programozási nyelv egy beépített könyvtára, a *librosa* segített. Mind idő szerinti tulajdonságok, mind frekvencia szerinti tulajdonságok generálva lettek. Idő szerinti:

- középértékek („central moments”): „amplitúdó szerinti átlag, szórás, ferdeség és az eloszlásgörbe meredeksége” tartozik ide
- nulla-átlépési-érték („Zero Crossing Rate - ZCR”): megmutatja, hogy „a jel hányszor vált pozitív tartományból negatívba”. Ehhez minden 10 másodperces rész felbontásra kerül kisebb részekre, úgynevezett „frame-ekre”.
- „Root Mean Square Energy (RMSE)”, ahol minden időpillanatban vett energiák négyzetösszegének négyzetes közepét veszi
- BPM (beats-per-minute): hány ütem található egy percen belül átlagosan

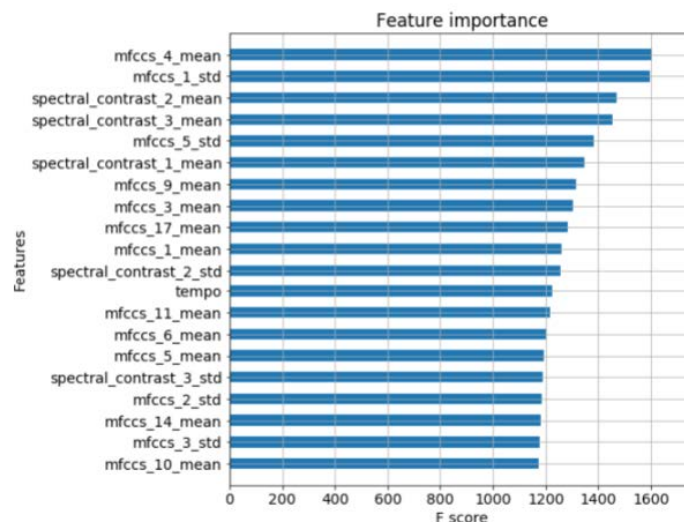
Az idő szerinti tulajdonságok mellett a frekvencia szerinti tulajdonságok:

- „Mel-Frequency Cepstral Coefficients (MFCC)”: a spektrogramok előállításában is szerepe volt, itt mint konkrét értéként szerepeltetve

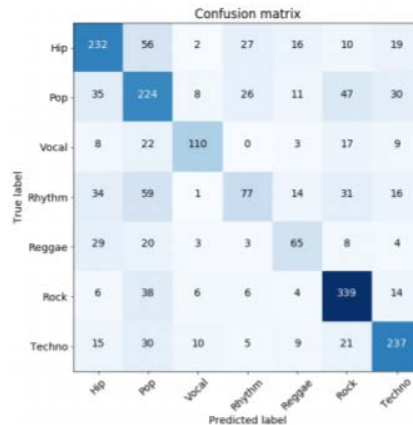
- kulcs („chroma features”): megadja, hogy adott zeneszámba mi a jellemző zenei kulcs, véges értékkeszlettel rendelkezik, 12 különböző kulcsú lehet egy adott zene („pitch classes” értékek szerint)
- spektrális centroid („spectral centroid”): „minden egyes frame-re meg van határozva az a frekvencia, ahol a legtöbb energia központosul”
- spektrális sávszélesség („spectral bandwidth”): „a p-edik időpillanatra vett spektrális sávszélesség a p-edik időpillanat és a spektrális centroid közti összefüggést írja le”
- spektrális kontraszt („spectral contrast”): „minden egyes frekvenciasávban a legkisebb és legnagyobb magnitúdójú érték közti különbség”
- spektrális legördülés („spectral roll-off”): megadja, hol található azon frekvenciaérték, ahol „a zenére jellemző összenergia értéke 85%-nál kisebb”

Klasszifikátorok, osztályozók szintjén szóba került a logisztikus regresszió, a „random forest” módszer, a „gradient boosting (XGB)” és a „support vector machines (SVM)”. A modellek helyességét és a kiértékelést olyan statisztikai mutatókkal támasztották alá, mint pontosság (a stílusmeghatározásnak), „F-score” („a konfúziós mátrix alapján készített harmonikus középérték a precision és recall értékeket figyelembe véve”) és az „AUC” érték (amely a „fals pozitív és igaz pozitív értékek alapján tesz összefüggést”).

Az eredmények pedig meggyőzőek, minden módszer tekintetében. Természetesen itt is előfordulnak helytelenül prediktált értékek, de 100%-os tanítást nem, vagy csak nagyon nehezen lehet megvalósítani. Önmagában már a zenei stílusok sem különülnek el egyes esetekben túlságosan élesen, erre a tanulmány szerzője is megerősítést adott.



2. ábra A 20 leginkább befolyásos tulajdonság megnevezése, F-score érték alapján [4]



3. ábra Egy VGG16 modellre történő tanítás legjobb eredményeit tartalmazó konfúziós mátrix – a főátlóban a helyes predikciók számával [4]

A 3. ábra szerinti konfúziós mátrixból leolvasható, hogy minden stílus esetén a legtöbb zenét a megfelelő stílus jegyében találta meg. Viszont, például a Rhythm stílus esetén igen magas arányban prediktált Pop vagy Hip stílusba.

2.2 A felismerni kívánt zenei stílusokról

Mielőtt említésre kerülnének azok a zenei stílusok, amelyek fel lesznek ismerve, szükséges pontosítani, illetve meghatározni azt, mit is jelent maga az a fogalom, hogy „zenei stílus”. Konkrét, egzakt definíciót nehezen lehet adni a tág felfogás miatt, de Aszalós Tünde megfogalmazása tűnik a legkézenfekvőbbnek. Szerinte a zenei stílus „Kortól, műfajtól és személyiségtől függő egységes jelleg, ami az eszközök: forma, harmónia, melodika, ritmika alkalmazásának módjában nyilvánul meg”, másképp megfogalmazva, kicsit általánosabb jelleggel: „azon kifejezési eszközök összessége, ami különböző művészeti ágak műfajaiban, műformáiban tárul fel.” [6] Értelmezése szerint egy stílust (többek között) olyan közös vonások, egybeesések jellemzik, mint „adott mű gondolati és érzelmi tartalma, a hangerő, a hangszín és a hangeffektusok, a tempó, a különleges ritmusok, a dallamformálás, dallamtagolás, az adott szövegek differenciálása, a zene programja, valamint a harmónia és formavilág”. A fejlesztési folyamat eredményeképpen az alkalmazás az alábbi stílusokat fogja felismerni.

Pop

A popzene, avagy teljes nevén „populáris zene” gyakori és igencsak kedvelt stílusként jelenik meg a zenék között. Nem véletlen az elnevezés, populáris, mint népszerű műfaj. [7]¹ Popularitását köszönheti, hogy tömegek számára szól, benne könnyedén megjegyezhető és felidézhető dallamok találhatók. Nagyon sok alfaja van (például stílusként megemlíthető a pop-rap és a jazz-pop, ami a határok elmosódását erősíti), talán emiatt is népszerű sokak számára. Ütemezése szempontjából gyakran figyelhető meg ismétlődő ritmus. Ahogyan a verseknél a rím, itt is megfigyelhető egyfajta ritmusos

¹ „A Zenei Enciklopédia a >>Szórakoztató Zenei Lexikon<<(Hölzer Tamás, Enciklopédia Kiadó, 2003) digitális utóda.” (forrás: <https://www.zenci.hu/oldal/impresszum>)

dalforma. Előadhatja egyvalaki, de csoportban előadott alkotás sem ritka. Jellemző hangszerei közé sokféle tartozhat.

Rock

Ha csak két zenekart kellene említeni a sok, rock stílusban előadók közül, akkor a Queen és a The Beatles zenekar sokak számára ismert lehet. Ugyanakkor egy rendkívül tagolt stílusról van szó, némely teljesen eltérő hangszerezettséggel, vokális előadással, ezáltal kifejezőerővel bír. [8] Ezen műfajnak társműfajaként megemlítendő a metál (vagy heavy metal), amely az egyik, ha nem a legextrémebb zenei műfaj, minthogy ezt a stílus magyar neve is jelképezheti, jelentése nehézfém. A rockzene népszerűsége ma is töretlen, története egészen a második világháború befejezéséig nyúl vissza. Hangzásában kiemelt szerepe az elektromos gitárnak, a basszusgitárnak és a doboknak van, de előfordul billentyűs tag is. Leginkább csoportokban adnak elő, úgynevezett rockbandákban, ahol 4-5 ember alkot legalább egy ilyen csoportot.

Klasszikus

Pontos meghatározását elnevezésbeli ellentmondások nehezítik. Bizonyos szempontból klasszikus zenének csak azok a zenék tekinthetők, amelyek a XVIII. században a klasszikus zenei korhoz kapcsolódnak (amilyen irányzat továbbá például a barokk kor, a reneszánsz kor vagy klasszicista kor). [9] A tágabb, és a programban is használt megközelítés szerint klasszikus zenének tekinthető azon komoly zenék mindegyike, amely letisztult, művészeti korszakbeli megkülönböztetés nélkül. Percenkénti ütemszámáról nehezen lehet itt beszélni, ugyanakkor tény, hogy legtöbbször a vokál-teljes zeneszám arány itt a legkisebb, további kiindulási lehet a hangszereket illetően a hegedű, a zongora vagy a különféle fúvós hangszerek.

Rap

A monoton és olykor szegényes ütemezés mellett az előadó mondanivalója található az előtérben, így a klasszikus értelemben vett zenékkal ellentétben itt jóval magasabb, ha nem is a legmagasabb vokál-zeneszám arány. Hangszereket illetően a mélyebb hangokat megszólaltató hangszerek a dominánsak, a dob és a mély gitárok, továbbá szerepe a különböző hangeffekteknél van, amit olykor az előadók, saját maguk szólaltatnak meg. [10] Az előadó, avagy rapper ritmikusan adja elő mondanivalóját. Számos hasonlóságot lehet felfedezni a hip-hop zenei stílussal.

House

Ezen stílus zenéiben fellelhető hangok a magas frekvenciatartományban mozognak, mély hanggal alig találkozni. Kifejezetten magasabb hangot kiadó hangszereket, illetve elektronikusan előállított dallamot használ az előadója. Több zenei stílus együtteseként jelenik meg, a „diszkó, electro, funk, synthpop és az R&B leszármazottja” [11]. Egy-egy

tipikus hangszerei közé pedig a „szintetizátor, a dobgép, a sequencer és a sampler” tartoznak.

Trance

Viszonylag újnak számító zenei stílus, története csupán az 1990-es évekig nyúl vissza, azóta elsősorban a fiatalok körében nyújt sikert. [12] Az elektronikus zene egyik ága, ugyanakkor sok különbséggel rendelkezik. Többféle társ műfaja van (és az idő előrehaladásával csak egyre több és több), a legnépszerűbbnek a „felemelő” uplifting trance stílus mondható. Pörgős, leginkább mély hangok, és sokféle dallam jellemzi, amelyek fülbemászó jellegükkel nyerik meg a hallgatójukat.

Jazz

Az 1900-as évektől egészen napjainkig népszerű műfajnak mondható könnyűzenei stílusirányzat. [13] Eleinte szájhagyomány és hallás útján terjedt, emiatt terjedhetett gyorsan, nem volt szükséges a kottaolvasás tehetsége. Fontos eleme továbbra is az improvizáció, amihez azonban szabályok és kötöttségek vonatkoznak. Előadásmódjára jellemző, hogy nem rendelkezik túl magas ütemszámmal, sok lassú zeneszám kötődik ehhez a stílushoz, jellemző hangszerei a fúvós trombita, klarinét és szaxofon, de a zongora is kulcshangszerként jelenik meg egy-egy jazzelőadó esetén.

Elektronikus

Ahol a hangokat valamilyen számítógép hozza létre, szövegezteti meg, az generálja, ha csak egy mondattal kellene jellemezni ezt, a 70-es évektől erőteljesen jelen lévő zenei stílust. Persze előfordulhat valódi hangszerek által kiadott zenei kiegészítés, de aránya igen elenyésző. A stílus eleinte inkább való volt hallgatásra, mintsem táncolásra, viszont a 90-es évektől kezdődően a különféle diszkók alaphangulatát szolgáltatták. [14] A hangokat a stílusbeli előadók, másnéven DJ-k VST-k, Virtual Studio Technology segítségével hozzák létre, az ilyen hangokat ebből adódóan VSTi-nek (VST Instrument) is emlegetik. [15] A felismerni kívánt stílusok közül ez mondható a legszerteágazóbbnak.

Latin

Elsősorban gyűjtőfogalomként ismert stílus, spanyolajkú területekről származik, mint Spanyolország, Portugália, illetve Latin-Amerika bizonyos országaiból. [16] Jellemzője, hogy vokálisan elsősorban anyanyelven történik az előadás, jellegzetes hangszerei közé pedig odasorolható a különböző vonós hangszerek, gitárok, az úgynevezett rázó hangszerek (mint például a csörgő) és a dobok. Sokszor mint tánczene jelenik meg a köztudatban, ugyanakkor a csacsacs, a rumba vagy épp a salsa ugyanolyan latin zenei stílus, mint amilyen táncstílus is.

Country

Nagyjából a XX. század elejére tehető a zene stílusának megjelenési, amely az Amerikai Egyesült Államokból, annak is déli részéről „származik”, sokaknak mai is a legnépszerűbb zenestílusának számít. Igyekezetek történtek annak elérésére, hogy széleskörűen elterjedhessen, de mindez kudarcba fulladt, megmaradt leginkább „az amerikaiak népzenejének”. [17] Érdekes összefüggése mutatkozik a rock stílussal, úgy tartják, hogy „a country rock a popzene egy al műfaja, a rock és a country összeállításából alakult ki” [18], manapság is ez a leginkább ismert részműfaja. Érdeemes lesz figyelni összefüggéseit más, említett stílusokkal kapcsolatosan.

Hiphop

Szintén az Amerikai Egyesült Államokban megjelent stílusról van szó, melynek eredete pontosan ismert, Bronx. Nem is egy konkrét stílusról, hanem egy „szubkultúráról” van szó, amely rendelkezik „saját zenei stílussal, ami az MCing/emszízés” [19], ami nem jelent mást mint „ütemre szövegelést”, de megfigyelhető benne rappelés, továbbá „DJing/Scratch = lemez-karcolás” elemei. Tematikáját illetően az ilyen stílusú zenék, legyen szó bármely alműfajáról, „laza, határozott, agresszív, néha bolondos”. Zenei stílusaira odafigyelendő, hogy a rap elemei megjelenhetnek benne, az kiegészülve egy-egy hiphopra jellemző elemmel, mint az említett lemezkarcolás is.

New Age

A popzene egy-egy, egymással ütköző alműfajából, keletkezett, amely nyugodtságot áraszt előadásmódjában. Fontos róla megjegyezni, hogy „egy szinten az elektronikus zenéből származik a kései 1960-as évekből, néha >>űrzenének<< is nevezték” [20]. Egyes rockzenekarok, mint például a Pink Floyd is alkalmazták műveikben. Maga a New Age zene „az elégedetlenség műve”, hiszen a kortárs alkotók elégedetlenek voltak az újfajta stílus és zenekészítési mód terjedésével, és „elkezdtek békés, zavaró dalokat írni és rögzíteni, amelyek főleg gitáron és zongorán játszódtak”. Az újfajta, elektronikus stílus végül vegyült a hagyományossal, így „készült” el. Megemlítendő, hogy később a jazz és klasszikus zenék is elkezdtek megjeleníttetni a stílus jellemzőit.

Dance

Pontosabb lenne a megfogalmazás, ha mint EDM, azaz „Electronic Dance Music” [21] szerepelne, ennek egy rövidítéseként létező stílus. Ha még hozzákerül az, hogy „röpke néhány év alatt popzene lett”, akkor szembeötlik, hogy csupán két félmondatból 2 főbb stílus jegyeit magán hordozza. Előzmény stílusának a diszkót tartják, ugyanakkor a diszkó mint stílus sosem vált népszerűvé. Érdekesség, hogy Amerikában rendre egy stílusként említették régebben az „eurodance-szel, és a tempósabb, táncorientált R&B-vel”, illetve a hiphoppal is. Az idő előrehaladtával a trance, a dutch house, electro-house stílusok hordták magukon, újonnan egyes előadók, mint Avicii által populáris zene lett.

R'n'b

Inkább „könnyűzenei megnevezés, ami nem egy egyértelmű műfajra utal”. [22] Az Amerikából származó r'n'b stíluseredetét tekintve, amikor „létrejött”, megemlíthető a jazz, a blues, a gospel valamint a pop, a későbbiekben, alműfajaként magában vehető a hiphop-ot, valamint a dance-t is, így magán hordozva ezen stílusok jegyeit, összefügghet más említett stílusokkal. Zenéiben megjelenő hangszerek közé tartozik többek között a szaxofon, a zongora, az elektromos gitár, de énekhang, vokál is található alatta.

Techno

Az EDM, azaz az „elektronikus tánczene egyik fajtája” [23], származási helyét tekintve ez is Amerikába tehető. Stílusának előzményeként megemlíthető az electro, az electric jazz, valamint a Chicago house is. Rendkívül alap, minimalista hang- és eszközkészlettel rendelkezik készülésekor, leginkább elektronikus jegyek övezik. Egyesek szerint, nem összekeverendő más stílusokkal, meg lehet különböztetni például a trance-től, valamint a tech house-tól.

2.3 Egy hasonló, népszerű alkalmazás – Shazam

„Ez a zene annyira fülbemászó! Annyira tetszik! De mi is lehet valójában?” Ezt a kérdést általában egy Shazam nevű alkalmazás megnyitása szokta követni. A Shazam ezt követően készít az eszköz mikrofonján keresztül egy néhány másodperc hosszú felvételt, ami után, egy kis idő elteltével visszaküldi az általa hallott zeneszám előadójának nevét, a zeneszám címét, valamint egy-két, az adott zenével kapcsolatos linket, ajánlást, letöltési oldalt. Felmerülhet a kérdés, hogy mégis hogyan működhet egy ilyen, több millió zeneszámmra alapozó alkalmazás, holott mindössze egy rövid részletet kap a felhasználótól, ami részlet lehet, hogy tele is van a környezetből származó, a detektálást vélhetően nehezítő zajjal.

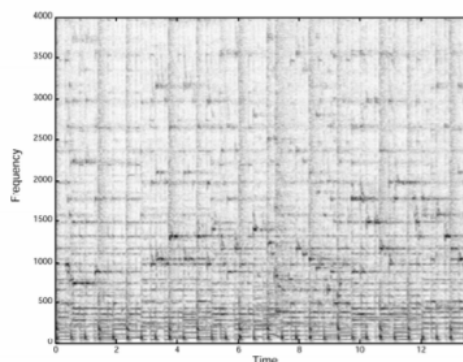
Nemcsak hosszú, hanem elég összetett útvonalat jár be az a hangfájl, ami felvételre kerül, és a végén lesz belőle egy előadó-zeneszám páros. [24] Szükséges egyfajta illesztés, aminek hatására össze lehet párosítani a mintát az azonosított zenével. Miután a felvétel elkészült, és elküldésre került az adat, ami adat már, mint „audio fingerprint”, azaz lenyomat kerül továbbításra. Így ezen „fingerprint” számára kell illesztést találni az adatbázis milliányi, ugyanolyan formájú „fingerprint” bejegyzése között.

Felmerülhet a kérdés, hogy egy zenének hogyan lehet kialakítani ezeket a lenyomatait, amelyeket el lehet tárolni, és lehet összehasonlítani a beérkező információval. Szükséges ilyenkor a hang, mint jel feldolgozása. Egy megközelítés szerint a legelső lépés nem más, mint az adott zeneszámmra elvégzett diszkrét Fourier transzformáció. [25] Fourier transzformáció során a digitális jelből matematikai egyenletet és összefüggést lehet készíteni. A jelfeldolgozás ilyen formája során egy hangfájl nem egy, hanem többdimenziós formában jelenik meg. Egy adott időpillanathoz mindig egy frekvenciaértéket rendel hozzá, ebből a sokaságból alakul ki a hangsáv. Ez egy átlagos,

nagyjából 3 perc hosszú zene esetén több millió, akár tízmillió mintát fog tartalmazni. Ez a mennyiség egyenesen arányos a zene hosszának másodpercben kifejezett értékével, az egy másodpercben lévő minták számával (gyakorlatban kifejezve, milyen frekvencián kerül rögzítésre a zene), illetve hogy hány csatornás az adott zene, sztereó, kétcsatornás rögzítés esetén egy kettes szorzót jelent, míg egycsatornás, monó esetén pedig egyes értéként szerepel. [24] Diszkrét transzformáció esetén a futási ideje $O(N^2)$ -es, emiatt igencsak hosszú időt venne igénybe. [26] A gyorsítás érdekében nem ilyen, hanem gyors Fourier-transzformációra van szükség, amely $O(N * \log_2 N)$ idő alatt végez, így hatékonyabbnak mondható.

Sokszor, kis időközönként elvégezve a gyors Fourier-transzformációt előállítható a zenének a spektrogramja, ami egy adott jelnek mutatja a frekvencia és az idő függvényében a hang amplitúdóját, ami nem más, mint „időben szinuszosan változó mennyiségek legnagyobb értéke”. [27]

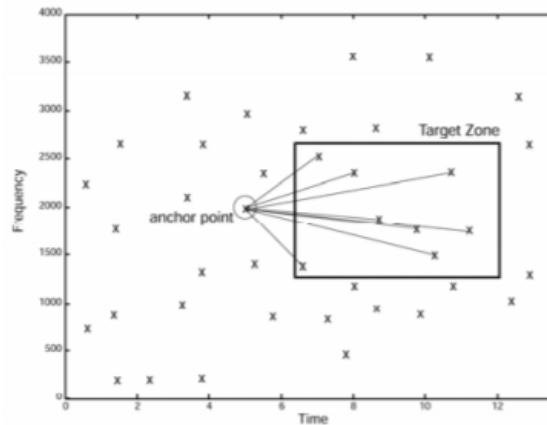
A „fingerprint” előállításának következő lépéseként ebből a diagramból szükséges megállapítani csúcserőértékeket, amelyek egyediségére vonatkozóan be lehet egy zeneszámot azonosítani. [24] Ezeket a csúcserőértékeket úgy lehet megállapítani, hogy a spektrogramon meg kell keresni a kiugróan magas amplitúdóértékeket, és úgy igazítani, hogy az adott környezetben lévő értékek ne legyenek ennél magasabbak. Ezen csúcserőértékek meghatározása a zajok zavaró hatásának kiszűrésére szolgálnak. Ezen pontok megállapításával elengedhető az összetett képek használata, elegendő megtartani a különböző idő-frekvenciapárosokat, amelyek a kiemelkedő amplitúdóértékekkel rendelkeznek. A folyamat itt még nem ért véget, hiszen még így is előfordulhat, hogy nagy mennyiség pontsokaság áll rendelkezésre, szükséges további egyszerűsítés ezen a halmazon.



4. ábra Egy adott zene spektrogramja az idő-frekvencia függvényben [28]

A következő lépésként szükséges a „kombinatorikus hash” érték meghatározása. [28] Ezen értéket meg lehet határozni úgy, hogy kijelölve, úgynevezett „anchor” pontként egy adott frekvencia-idő pontot (ahol azonosítva lett a kiugróan magas amplitúdóérték) mint kezdőpont, ettől keresni a következő néhány, imént meghatározott frekvencia-idő pontokat. A két pont között bizonyos időtávolság található, és bizonyos frekvenciabeli különbség figyelhető meg, így számítható egyfajta távolság a két pont között. Nem csak egy ilyen pont meghatározása szükséges, hanem egy adott területen belül több pont

számára szükséges, mindegyik rendelkezni fog egy-egy ilyen, hash értékkel. Minden ilyen, „anchorpont” - célpont páros el van tárolva a hozzájuk tartozó frekvenciakülönbséggel, a hozzájuk kapcsolódó időbeli különbség értékével, ami lesz maga a hash érték.

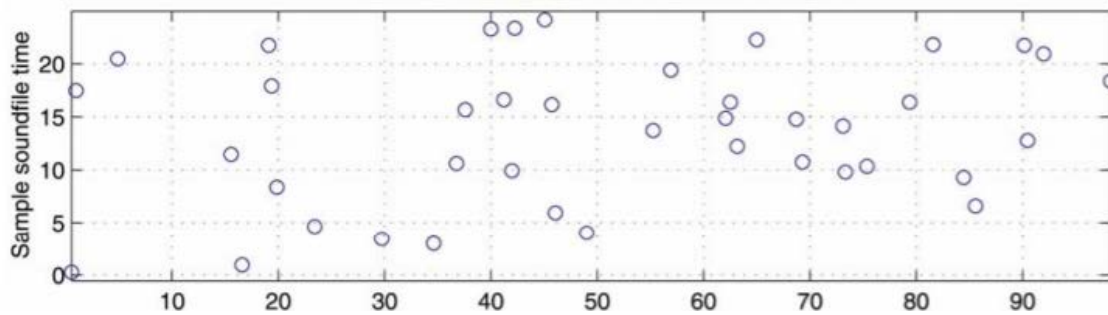


5. ábra A kombinatorikus hash generálása egy kiinduló, "anchor" pontból [28]

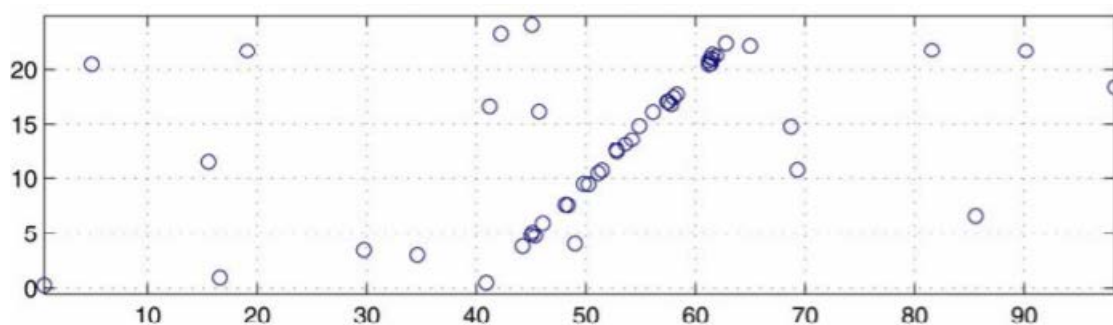
Amint megtörtént a felhasználótól beérkező zene „fingerprint” elkészítése, meg kell határozni azt a legvalószínűbb zenét, amely ezen pontok illeszkedéséből adódik. Lehet, hogy a felhasználó nem pont az „anchor” pont kezdetén kezd bele a zene felvételébe, így meghatározni szükséges azt az offset időt, ami időkülönbség megjelenik a felhasználó által felvett zenéből készített „fingerprint”, egy adott kijelölt „anchor” pontjának és a felvétel kezdete között eltelik, hogy ezzel az idővel már ne kelljen számolni, és lehessen illeszteni a felhasználó „anchor” pontjának kezdetét az adatbázisban lévő, lehetséges találatként vizsgálható zenék „fingerprintjeinek anchor” pontjához. Azaz keresendő a felvétel megkezdésétől számított első „anchor” pont, minek utána ki kell vonni az elindítás és a megtalálás közötti idő mennyiségét.

Illesztés akkor lesz megfigyelhető, ha a tárolt hash értékek és a felhasználó zenéjéből kiolvasható hash értékek egyeznek. Szemléletesen, ha illesztésre kerül „anchor” pont az „anchor” ponthoz, akkor lényegében olyan, mintha a bevitt zenét pont abban a pillanatban indítanánk el, és ugyanazt lehet hallani egyik és másik oldalon. Ha egy geometriai alakzatból lehet kiindulni, akkor kell venni egy olyan négyzetet, aminek oldalhossza a felhasználó által rögzített hangfájl offset-tel csökkentett, használható értékével. Egyezés akkor figyelhető meg, ha a hash értékek egy átló mentén illeszkednek egymásra, függvényyszerűen, ha behúzható egy origóból indított, $f(x) = y$ függvény, és a pontok erre illeszkednek. Például adott egy 15 másodperces, felhasználó által rögzített zene. Ebből, mint offset, megjelenik 2 másodperc, tehát az első „anchor” pontnak nevezhető pont a „fingerprint” esetén 2 másodpercnél jelentkezik, így a használható rész 13 másodperc marad. Szükséges keresni ezen a 13 másodpercen belül olyan számértékeket, melyek hash értékei hasonlóak, illetve egyeznek. Amennyiben egyezés figyelhető meg, azaz egyes hash-hash értékek egybevágó pontok, és egy átlóban jelentkezik, akkor kijelenthető, hogy megvan a zeneszám. Azaz, ha ez teljesül, akkor a

megállapításra, hogy arról az adott zenéről van szó, és ezt az információt visszajuttatja a felhasználó számára.



6. ábra Sikertelen illesztés, a hash értékek szétszóródva helyezkednek el. [28] X tengely mentén szerepelnek az adatbázisban lévő adatok, míg Y tengely mentén a mintaként beadott zenének megfelelő értékek



7. ábra Sikeres illesztés - jól látható a 40. és 60. másodperc közt az átlóban sűrűsödő pontok halmaza, az illesztés a zene 40. másodpercétől kezdődik meg. [28]

2.4 Gépi tanulás, neurális hálózatok, a konvolúciós neurális hálózatokról

Az összeszedett adatok feldolgozásához a gépi tanulás eszközére van szükség. Gépi tanulás esetén az egyes összefüggések, jellemző tulajdonságok megfigyelésére neurális hálózatok kerülnek meghatározásra. A kétfajta különböző modellhez két különböző fajta modellre is van szükség az adatok jellemzőinek különbsége miatt, hiszen ahogy az majd látható lesz, az egyik esetben képeket kell feldolgozni, míg másik esetben „egyszerű” adatok feldolgozásával kerül megállapításra egy-egy stílus. Ami ezekben közös lesz, az maga a neurális hálózat, melyekkel az adatok feldolgozásra kerülnek.

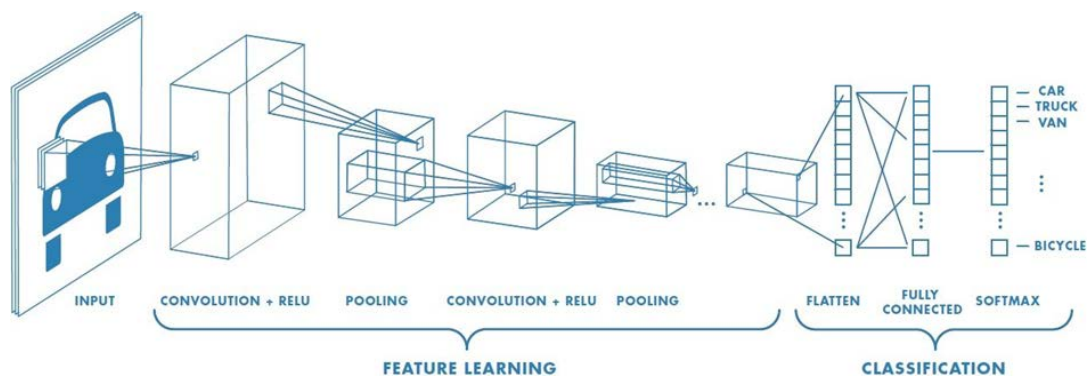
A neurális hálózat biológiailag inspirált koncepció, amelynek kiindulási alapja az agyi működésre utal. Működési alapegységei az úgynevezett neuronok, amelyek egymáshoz kapcsolva képesek egyre összetettebb információk feldolgozására, tehát tulajdonképpen utasítások sorozatát foglalja magában. [29] Segítségével olyan feladatok végezhetőek el, mint a csoportokba rendezés, azaz klaszterezés vagy pedig osztályozás, a klasszifikálás. Ehhez szükségeltetik a megfelelő osztályokba tartozó adathalmaz felcímkézése, amely által a megerősítéses tanítás folyamatához lehet jutni. Ezáltal lehet a különböző osztályokba tartozó elemeket felismertetni a hálózat által. Klasszikus példa a kutya-macska megkülönböztetés, adott számú kutya, illetve ugyanannyi számú macska képének feldolgozását követően figyelendő, hogy az alkotott modell mekkora magabiztossággal határozza meg egy számára megadott, de új képről,

hogy az kutya vagy éppen macska. Lényegében a gépi tanulás által adott megoldás ad erre egy olyan „univerzális, általánosított, mindenre kiterjedő képletet” [29], amely által meghatározható egy új bevitt érték eredménye, azaz annak kimeneti értéke. Jelen esetben a kimeneti érték egy-egy stílushármas lesz, a leginkább valószínűsített három stílus megnevezésével az azoknak szánt valószínűségekkel.

A feladat egyik felének megoldásához, amikor spektrogramokból kerül majd megállapításra egy-egy zene stílusa, azokhoz úgynevezett konvolúciós neurális hálózatok alaposabb ismeretére van szükség.

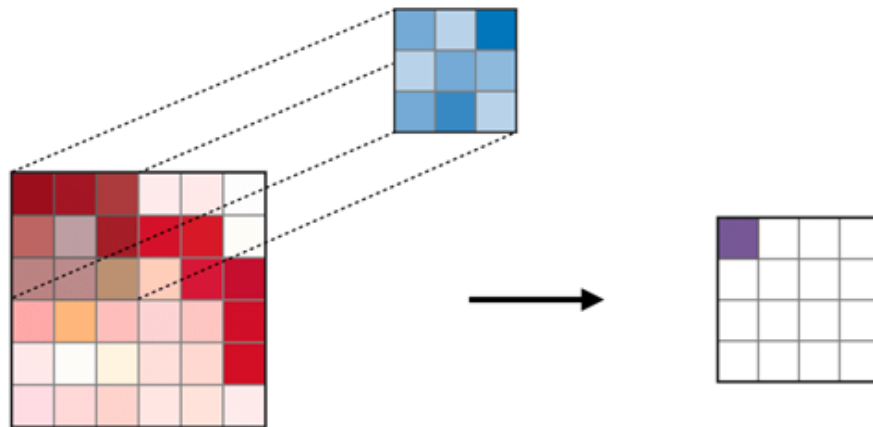
Az offline zenestílus-felismerés képek segítségével történik. Ahhoz, hogy a képek formája, jellegzetességei felismerésre kerüljenek, és lehessen klasszifikálni különböző osztályokba, egy olyan gépi tanulási formára van szükség, amely képes képek ilyen jellemzőit felismerni, azok alapján tanulni. A képekre való összetettségre vonatkozóan szükséges valahogy az egyes jellemzők, paraméterek egyszerűsítése, hogy azok tanulható formában legyenek.

A gépi tanulás szempontjából a konvolúciós neurális hálózatok „igen hasznosnak számítanak, amennyiben a megoldandó probléma multimédiás anyagok, mint kép, videó, zene, illetve szövegek közti klasszifikáció, osztályozás”. [30] Az osztályokba sorolás alapja a különböző részletek felismerésén alapul anélkül, hogy „manuális, kézi tulajdonság-megállapításra, kivonásra kerülne sor”, tehát ez adott szempontból automatikusan történik. Ugyanakkor szükségeltetik a tanulás megindításához számos paraméter, tulajdonság is, amely paraméterek a tanulás előtt már ismertek, és a rendszer számára átadhatóak. A hagyományos neurális hálózatokkal ellentétben előnye, hogy az nem veszi figyelembe a különböző a függőségeket a szomszédos, képet alkotó pixelek között, míg a konvolúciós igen. Azon megoldás ugyanis a különböző, képet reprezentáló RGB értékeket egydimenziós adatstruktúrába rendezi. Így viszont elvesznek a szomszédsággal járó kiegészítő információk, amelyek a különböző tulajdonságok felismerésében nagy szerepet játszanak (pl. detektálni egy nagyobb, RGB színskála szerint ugyanolyan színű pixeleket alkotó nagyobb alakzatot, amelyek egy adott csoport esetén előfordulhatnak). Konvolúciós neurális hálózatok esetén viszont ilyen jellegű adategyszerűsítés-, illetve manipuláció a modell felépítése során csak a végén esedékes. Így az eredmény pontosabb marad, és értékes információvesztés sem következik be.



8. ábra Példa egy konvolúciós neurális hálózatra – klasszifikáció különböző járművekre [30]

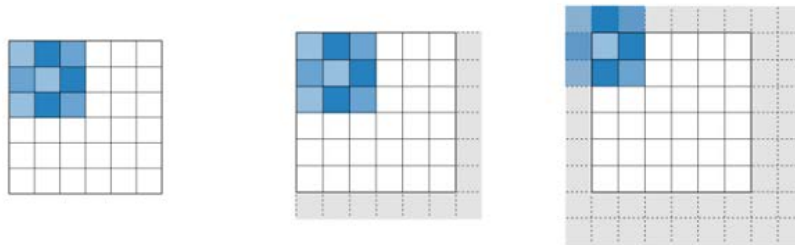
Ahogy azt a konvolúciós neurális hálózatokat reprezentáló, 8. ábra szerint megjelenő példakép is mutatja, a képfeldolgozás több fázison keresztül történik meg, amelyből végül megállapítható annak valamekkora pontossággal annak osztálya. A bemenetet adja, adják azok a multimédiás anyagok, jelen helyzetben képek, amelyekről eldöntendőek a különféle osztályok. Egy kép számos matematikai műveleten, fázison megy keresztül, minek végére előállítható a kért osztály. A hálózat első hányadában kerülnek megállapításra a különböző tulajdonságok adott képeken. Tulajdonságnak számít például egy autó esetén, hogy hol és merre található rajta egy kerék, vagy visszapillantó tükör, ezek viszonylag kötött jellemzők. Ahhoz, hogy egy ilyen megállapításra kerüljön, matematikai úton felbontásra kerül színcsatornák alapján a kép, amely szürkeárnyaltos kép esetén mindössze egy, ugyanakkor egy RGB színkóddal rendelkező színes kép esetén már három csatornát jelent. [31] Minden ilyen pixelt egy-egy színkód reprezentál, ezen értékek kerülnek eltárolásra annyi mátrixban (kétdimenziós kép esetén), ahány csatorna rendelkezésre áll. Miután ez megtörtént, számos, a módszer nevét alkotó, konvolúciós réteg kerül hozzáadásra a rendszerhez. A konvolúciós réteg lesz az a réteg, amely képes a különböző tulajdonságok felismerésére a képeken. [32] Ezt mátrixműveleteken keresztül éri el olyan módon, hogy programozott módon, egy adott méretű kernel, filter, szűrő végighalad a képkockák által reprezentált számértékeken, elvégzi a szükséges műveleteket, és adott eredményt az így létrejövő mátrixban tárolja. Egy $N \times N$ méretű és filterszámú konvolúciós réteg, mátrix alapesetben egy $A \times A$ méretű képből $((A-N) + 1)$ -szer $((A-N) + 1)$ nagyságú konvolúciós képet fog megalkotni.



9. ábra Egy 6x6-os képen 3x3-as méretű kernellel végzett konvolúciós műveletek által előálló 4x4-es mátrix [32]

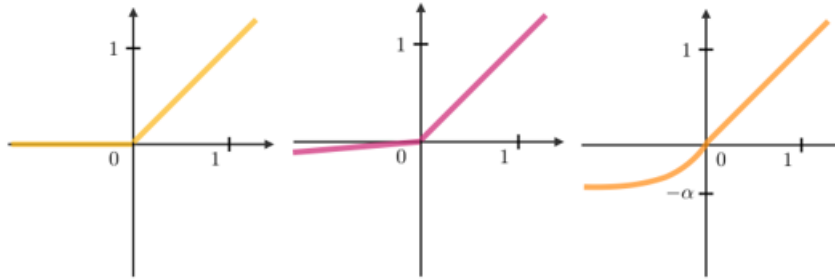
Ez a kimeneti érték és mátrix több paraméterrel befolyásolható. Megadható az, hogy egyszerre mennyit, hány „lépést” mozduljon a megfelelő irányba, jobbra, illetve a kép mátrix jobb szélén, lefelé. Ez a pixelértékeket tároló mátrix fizikai határáig mozgítható, azaz addig, amíg még található érték a szűrő által érintett részmatrixon (teljesen üres részmatrixon nem végez mátrixműveleteket). Ez a tulajdonság a *stride* tulajdonság. [32] Figyelembe vehető tulajdonság, hogy a szűrő mozgó ablakában mennyi olyan érték található, amelynek egyes részei a tényleges dimenziókon kívüliek, mégis található

benne legalább egy darab, nullától különböző érték. Ez a *zero-padding* tulajdonság, amely nullás értékekkel tölti fel a bemenet határait, és háromfajta különböző módot lehet elsődlegesen megkülönböztetni. Mindhárom a konvolúciós ablak haladása szempontjából fontos, mennyi értéket tud érinteni, figyelembe véve az esetleges *stride* értéket, ami alapeseten 1, minden dimenzióban. A *valid* megoldás esetén a mátrix nem egészül ki extra nullás értékekkel, ha a kernel dimenziói már nem illeszthetők az adatokra, akkor azokra nem végez konvolúciós műveleteket. A *same* megoldást szokás „half-padding” néven is illetni, annyi nullás értékkel egészül ki a jobb-, illetve lenti határok mentén, amennyi szükséges az összes érték eléréséhez, figyelembe véve a *stride* tulajdonság értékét. Ezen típus esetén egy $A \times A$ bemeneti képre használt $N \times N$ méretű szűrő kimenete egy ugyanakkora, $N \times N$ -es konvolúciós mátrix lesz. Ezeken felül megemlíthető még a *full zero-padding*, amelynek jellemzője, hogy „a végkonvolúciókat a bemeneti határértékekre alkalmazzák” [32].



10. ábra Példaképek a *zero-padding* tulajdonság típusaira és lehetséges dimenzió-kiterjesztéseire: *valid* (bal), *same* (közép) és *full* (jobb) [32]

Az így kapott mátrixon egy aktiváció kerül végrehajtásra annak érdekében, hogy a hálózatban jelen legyen a nem-linearitás azáltal, hogy a mátrixműveletek során eredményül kapott értékeket figyelembe véve eliminálja. A leginkább használt aktivációs függvény a *ReLU*, avagy *Rectified Linear Unit*, amely „gyorsabb és hatékonyabb tanítást biztosít azáltal, hogy a mátrixban lévő negatív értékeket kinullázza, és csak a pozitív értékeket tartja meg benne”. [32] Ekkor viszont észrevehető lehet, hogy a modell pontossága nem akar javulni, látszólag „nem akar tanulni”. Ez a „dying ReLU” jelenség, amely során a „ReLU neuronok inaktívvá válnak, és kimeneten csak nulla értéket szerepeltetnek. Ekkor opcióként felmerülhet, hogy egyes, nullától kisebb értékek is kerüljenek átadásra aktivációként a következő fázisba. Ehhez kétfajta megközelítés a *LeakyReLU* és az *ELU* nevű aktivációk.



11. ábra A konvolúciós mátrixon alkalmazható aktivációk függvényei, általuk adott kimeneti értékeik az Y tengely mentén szerepeltetve: ReLU (bal), LeakyReLU (közép) és ELU (jobb) [32]

A konvolúciós mátrixon elvégzett aktiváció után az adatok további szűkítése érdekében, ezáltal a számítási komplexitás csökkentése érdekében pooling rétegek kerülnek a hálózatba bekötésre. Alapvetően kétfajta módja van az adatok megtartása szempontjából. Bármely módszer is kerül felhasználásra, lényeges szempont, hogy mekkora dimenzióból kerül kiválasztásra a megfelelő érték. Az egyik ilyen a *max pooling*, mely során a poolingnak választott dimenzióból a legjobb érték kerül kiválasztásra, a másik pedig az *average pooling*, amikor a poolingnak választott dimenzióban átlag kerül kiszámításra, és az az érték fog szerepelni a pooling kimeneteként. Az így kapott kimeneti kép dimenziója a választott pooling dimenzióval lesz egyenlő. Leginkább a *max pooling* típus használatos, az tartja meg a lényeges információt, míg az *average pooling* átlagjellegéből adódóan zajos lehet. [32]

A konvolúciós rétegekből és pooling rétegekből a hálózatban több is szerepelhet. A konvolúciós rétegek esetében az első réteg(ek) az alacsony szintű tulajdonságok felismerésére alkalmasak. Ebből kiindulva, a megfelelő értékeket pooling alkalmával megtartva, egyre előrébb haladva egyre összetettebb, magasabb szintű tulajdonságok kerülnek megállapításra, ami valójában a tényleges cél, az összes, magas szintű, fontos tulajdonság megállapítása, amely végül előállítja a megfelelő stílust.

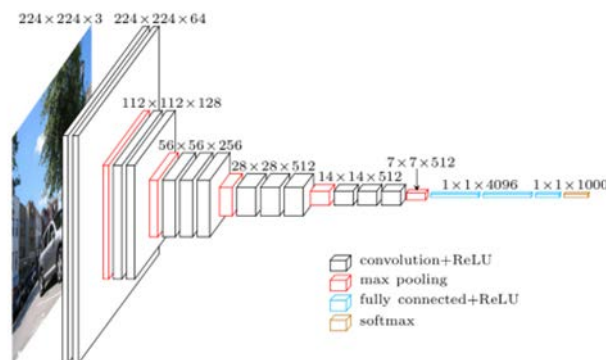
Amennyiben a műveletek megvannak, meglehetnek ezen magas szintű tulajdonságok, úgy a legutolsó pooling réteget követően lehet végrehajtani magát a tényleges klasszifikációt. Ehhez a gyűjtött információkat, ahogy a 8. ábra mutatta, egydimenzióba szükséges rendezni, ekkor már nem szükséges fenntartani a felismerés szempontjából fontos szomszédsági tulajdonságokat, a szükséges információk össze lettek szedve, ki lettek számítva, meg lettek határozva. Ezen egydimenziós adatok egy sűrűn hálózott hálózatba kerülnek bekötésre, melynek végén egy annyi neuronból álló réteg kötendő, amelyben a neuronegységek mennyisége megegyezik a kimenetként szánt osztályok mennyiségével, azaz ha tíz kategória felismerendő, úgy tíz egységet fog tartalmazni. Az utolsó réteg aktivációja *softmax* aktiváció lesz és nem a *ReLU* vagy annak valamely változata. *Softmax* aktiváció megteszi, hogy kimenetére egy olyan vektort fog adni, amely az egyes kategóriákra vonatkozó valószínűségek lesznek. [32] Ha összegezve vannak ezek a valószínűségek, akkor eredményül 1-et ad. Adott kép esetén a legvalószínűbb osztályhoz tartozó érték valószínűsége lesz a legmagasabb mind közül, amely nem jelenti azt, hogy más értékek nem lehetnek ezen érték közelében.

Előfordulhat ugyanis, hogy egy adott bemenet nem határozható meg 100%-os egyértelműséggel, hogy mely kategória tagja, mivel tartalmazhat olyan elemeket, amelyek által több kategóriába is besorolható.

2.5 Kész, működő, bizonyítottan pontos konvolúciós hálózatok

Mielőtt a tanítás megkezdésre kerülhet, felmerül a kérdés, hogy korábbi feladatokhoz készült-e már korábban olyan modellarchitektúra, amely képes volt adott feladat megoldására? Illetve ezzel kapcsolatosan, ez képes-e arra, hogy ezen feladat megoldásában is segítséget nyújtson? Vagy esetleg saját architektúrával kellene próbálkozni? Az egyik döntés szerint lehetne egy olyan modellt állítani, amely teljesen az elejéről kerül kialakításra a szükséges ismeretek tudatában. A másik döntés értelmében egy már bevált architektúrát felhasználva történne a tanítás.

2010-től 2017-ig évente került meghirdetésre az ImageNet Large Scale Visual Recognition Challenge nevű kihívás, amelynek keretében több kategóriában, gépi tanulás alapú megoldásokat vártak egy 1000 kategórián alapuló adatbázishoz, amelynek elsődleges témája az objektumfelismerés, -lokalizálás, illetve kategorizálás. [33] 2014-ben készült el ezen kihívás keretében a VGG16 nevezetű modell, amely amellett, hogy konvolúciós neurális hálózat, meg is nyerte adott évben ezt a versenyt. [34] Azóta továbbra is helytálló modellnek tekintendő, amelynek szerkezete jó kiindulási alap lehet a hasonló feladatok megoldására. Megnevezésében a 16-os szám arra utal, hogy összesen 16 olyan réteggel rendelkezik, amelyeknél tanítható paraméterek szerepelnek. Meglehetősen nagy, összetett hálózatról van szó, mintegy 138 millió paraméterrel rendelkezik alapesetben.



12. ábra A VGG16 architektúrája szemléletes ábrán – itt az utolsó softmax aktiváció az ImageNet kihívásnak megfelelően 1000 kategóriára vonatkozik [34]

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 224, 224, 64)	1792
conv2d_2 (Conv2D)	(None, 224, 224, 64)	36928
max_pooling2d_1 (MaxPooling2)	(None, 112, 112, 64)	0
conv2d_3 (Conv2D)	(None, 112, 112, 128)	73856

conv2d_4 (Conv2D)	(None, 112, 112, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 56, 56, 128)	0
conv2d_5 (Conv2D)	(None, 56, 56, 256)	295168
conv2d_6 (Conv2D)	(None, 56, 56, 256)	590080
conv2d_7 (Conv2D)	(None, 56, 56, 256)	590080
max_pooling2d_3 (MaxPooling2D)	(None, 28, 28, 256)	0
conv2d_8 (Conv2D)	(None, 28, 28, 512)	1180160
conv2d_9 (Conv2D)	(None, 28, 28, 512)	2359808
conv2d_10 (Conv2D)	(None, 28, 28, 512)	2359808
max_pooling2d_4 (MaxPooling2D)	(None, 14, 14, 512)	0
conv2d_11 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_12 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_13 (Conv2D)	(None, 14, 14, 512)	2359808
max_pooling2d_5 (MaxPooling2D)	(None, 7, 7, 512)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_1 (Dense)	(None, 4096)	102764544
dense_2 (Dense)	(None, 4096)	16781312
dense_3 (Dense)	(None, 15)	61455
=====		
Total params: 134,321,999		
Trainable params: 134,321,999		
Non-trainable params: 0		

13. ábra A VGG16 architektúrájának összefoglalója a benne szereplő rétegek szerint, 15 kategóriára vonatkozóan

A modell összefoglalójából (13. ábra) kiolvasható, hogy az alacsony filterszámú konvolúciós rétegekből 2 darab szerepel egymást követően, míg később, a magasabb filterszámú rétegekből 3 darabot köt egymásba. Ezen konvolúciós duó-, illetve triórégeket minden esetben egy pooling réteg követ, típusát tekintve minden esetben *MaxPooling* fordul elő, amelynek dimenziója 2x2-es, elérve azt, hogy minden pooling réteg negyedeli a következő konvolúciós rétegek számára átadott paraméterek számát. Az utolsó ilyen réteg után a konvolúciókat felváltandó sűrűn kötött hálózatra vált azután, hogy a fennmaradó értékeket egydimenzióssá alakította a *Flatten* réteggel. Az utolsó ilyen *Dense* réteg kimeneti paraméterszáma megegyezik azon értékkel, amennyi stílusra történik prediktálás.

Komplexitását, valamint tanító paramétereinek számát tekintve a modell összetettebbnek számít a megvalósítandó feladatnál. A VGG16 modellje ugyanis egy 1000 kategóriára vonatkozó képkategorizáló, jelen esetben pedig 15 kategória áll rendelkezésre. Így felmerül a modell egyes rétegeinek „lefagyasztása”, valamint lekapcsolása a rendszerről. Ezzel csökken a tanító paraméterek száma, ezzel viszont az

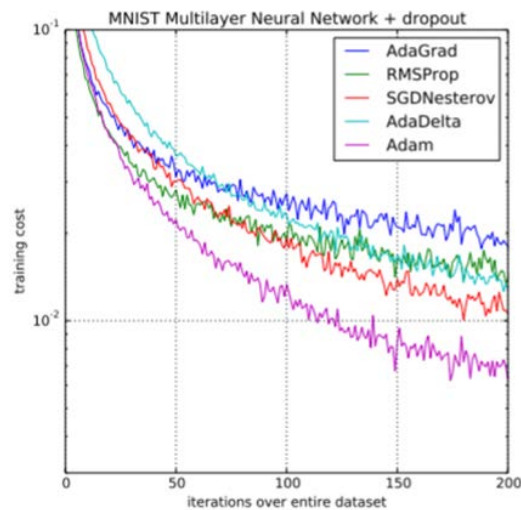
adatok taníthatóvá válnak videokártyán a feldolgozás gyorsítása érdekében. Gyakran szembesülő probléma lehet, hogy egy modell elkészítése után annak tanító paraméterei nem elhelyezhetőek teljes egészében GPU-n, így szükséges annak zsugorítása, hogy a tanítás maga elindulhasson.

A tanítás lefuttatásához a felállított modellen felül szükséges egy megfelelő optimalizáló kiválasztása. Ezekből többféle áll rendelkezésre, számos ezek közül a tanítások során kipróbálásra kerülnek. Kipróbálandó optimalizálók közé tartoznak többek között az *Adam*, *Adamax*, *SGD*, illetve az *RMSProp* nevűek.

Az optimalizálók közül a későbbiekben elsők között került kipróbálásra az *SGD*, azaz a *Stochastic Gradient Descent* nevű. Kiindulva *Gradient Descent* voltából, olyan függvényt értünk alatta, amelynél a modell veszteségértékeit a lehető legalacsonyabb értékre csökkentjük, ezáltal elérve az optimális állapotot. [35] [36] Figyelembe veszi a modell különböző paramétereit, valamint a modell tanításához felhasznált adatokat. „Iteratívan, minden egyes lépésben választja paramétereit, miközben megtalálja a súlyok és az úgynevezett *bias* értékek legjobb kombinációját, a már említett veszteség minimalizálása érdekében.” [37] Stohasztikus mivolta pedig onnan ered, hogy ezen optimalizáló a tanítómintából egyetlen példát választ ki, amelyre támasztva számítja ki a következő lépések gradiensek egy becslését. Megfelelő számú iteráció mellett az *SGD* működőképes, ugyanakkor eléggé zajos, többek között a véletlen választás mivoltából. Paraméterként megadható számára, milyen legyen tanulási rátája (*learning rate*), valamint állítható *momentum* érték is, amellyel a helyes irányba terelhető a veszteségérték csökkentése, elkerülve, tompítva az ellenkező, nem-helyes irányba történő oszcillációt.

Egy újabb megközelítésként 2015-ben publikálták az *Adam* elnevezésű, stohasztikus optimalizációra alkalmas metódust, melyet „könnyű végrehajtásúnak, számítási szempontból hatékornak, alacsony memóriaigényűnek” [38] állítottak be, amely „jól alkalmazható a nagyobb problémák megoldására az adatok, illetve paraméterek szempontjából”. „Hiperparaméterei nem igényelnek nagy hangolást, illetve számos további algoritmus alapját is adja”. Két korábban bevált, gyakran használt *SGD* kiterjesztést használták fel megalkotásához. Az *AdaGrad* megoldásból a „paraméterenkénti tanulási sebességet tartja meg, amely a ritka gradiensekkel kapcsolatban felmerülő teljesítmény-problémákat küszöböli ki”. A másik előny pedig az (úgyszintén felhasznált) *RMSProp*, azaz *Root Mean Square Propagation* megoldásból származik, amely szintúgy „fenntartja a paraméterenkénti tanulási sebességeket, amelyek a legutóbbi magnitúdó értékekből átlagolással kerül meghatározásra, amely adja a súly gradiensét, milyen gyorsan változzon az, zajos, nem feltétlen egyértelmű problémák esetén hatékony a megoldása”. Az *Adam* nem csak az átlagos első momentumból, azaz magából az átlagból számítja ki a *learning rate* értékét, hanem a „második momentumok átlagértékeit is felhasználja, mégpedig teszi ezt úgy, hogy a gradiens exponenciális mozgó átlagát és a négyzetes gradienst is meghatározza, ahol

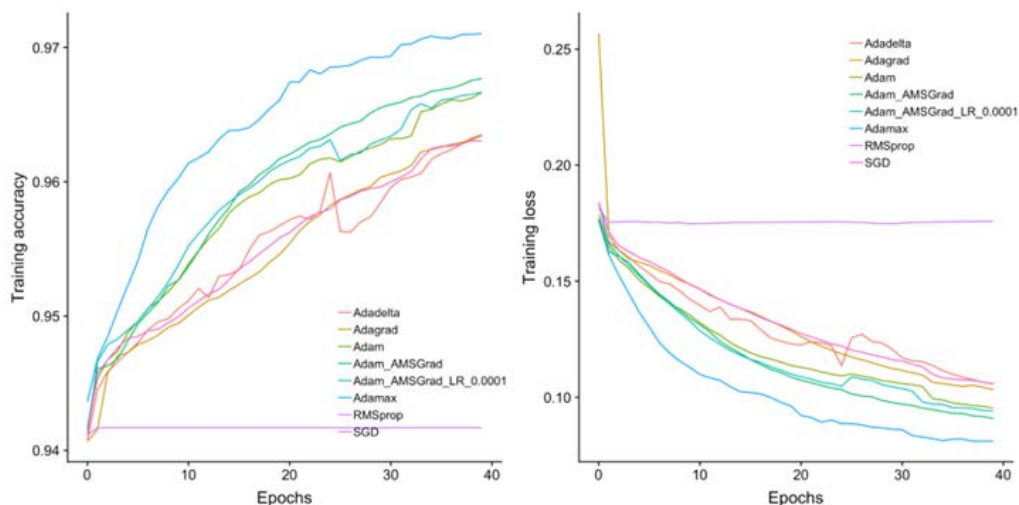
további paraméterei, a β_1 és a β_2 súlyozza ellen az esetleges *decay rate*, azaz „hanyatló” értékeket ezen mozgó átlagok szempontjából.” [38]



14. ábra MNIST képeken többrétegű neurális hálózattal végrehajtott tanítás eredményei a különböző optimalizálók eredményeivel a tanítási veszteségértékeken. Lila színnel jelölve az Adam optimalizáló [38]

Adamax variánsa a végtelen normára alapozik, mégpedig a „második szintű gradiensérték számításában”. „Meglehetősen nagy p értékek esetén az *Adam* esetében definiált L_2 norma kiváltható ugyan L_p normával, de meglehetősen instabilnak bizonyul”. Amennyiben „ p nem egy konkrét értékhez, hanem a végtelenbe tart, úgy nemcsak egyszerű, de egy stabil algoritmust eredményez.” [38]

Az egyes optimalizálók más feladatokban megnyilvánuló hatékonyságának ellenőrzésére számos diagram rendelkezésre áll. Néhány ilyen a Kaggle oldalon található, amelynek felhasználói az oldalon található feladatokkal kapcsolatban fórumon osztják meg más kompetitorokkal megoldásaikat. Egy ilyen képcsoportosítással kapcsolatos feladat kapcsán ugyanolyan modell felhasználásával, ugyanazon a feladaton az alábbi eredményeket hozták a különböző optimalizálók.

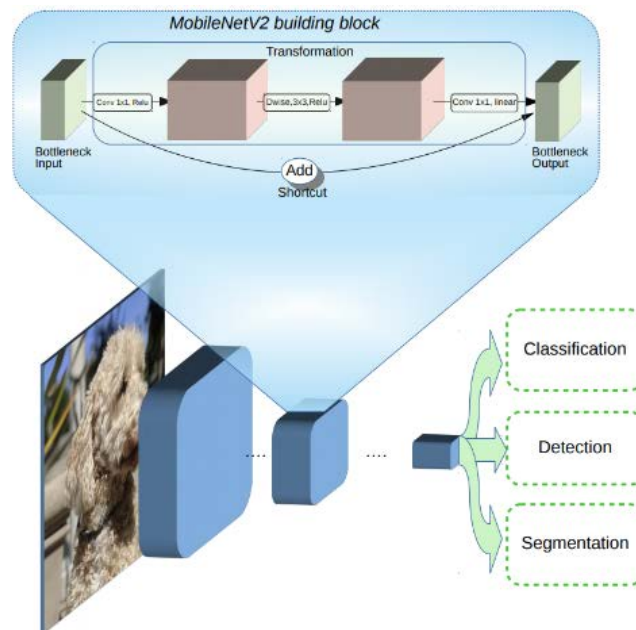


15. ábra ResNet50 modellen alapuló tanítás eredményei 512x512 nagyságú képeken, tanítási pontosságértékek (bal oldalon) és tanítási veszteségértékek tekintetében. Ezen modellen a legjobb optimalizálóként az Adamax (kék) került ki [39]

A *VGG16* mellett figyelembe került egy jóval alacsonyabb paraméterszámmal rendelkező modell is. A hatékony, state-of-the-art-nak tekinthető modellek a *keras* nevű Python library-ben kerültek implementálásra, amely szerint a *VGG16* modellével egyező pontosságú, de nagyságrendekkel kevesebb paraméterrel, konkrétan az elérhető modellek közül legkevesebbel rendelkező *MobileNetV2* egy másik megközelítése lehet a feladatnak. [40] Jelentős különbségként említendő meg, hogy a konvolúciós rétegek burkoltan szerepelnek a modellen belül, hiszen úgynevezett *bottleneck* rétegekből kerül felépítésre a hálózat jelentős része. Ahogy az a *VGG16* esetében is volt, ezen architektúránál is, bemeneti paraméterként 224×224 -es képeket vár el. További különbségek mutatkoznak az architektúra szerkezetében, valamint a *pooling* típusában.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

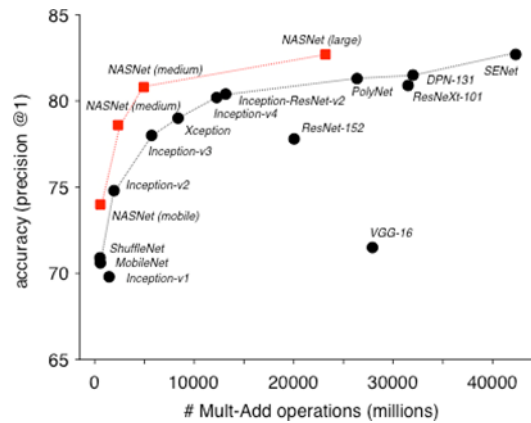
16. ábra *MobileNetV2* architektúrája, ahol „ c ”: a kimeneti csatornák száma, t : az úgynevezett kiterjesztési faktor, n : az ismétlés darabszáma, és s : a stride mérete az első ismétlés alkalmazásával, ahol $n > 1$, az ezt követő ismétlésekben pedig 1” [41]



17. ábra A bottleneck réteg felépítése, a transzformációs ablakban konvolúciós - mélységi („dwise”) konvolúciós - konvolúciós rétegek váltakozásával alakul ki a kimenet [42]

Szintúgy, ahogy a *VGG16* és a *MobileNetV2* modellek is az *ImageNet* kihívására készültek, a *NASNet* nevű is e célból került elkészítésre, amelyet ugyanúgy képkategorizálásra specializáltak. A *keras* dokumentációja szerint többfajta változata van. [40] Ezek közül talán a *NASNetMobile* emelhető ki a paraméterekhez képest viszonyított pontosságok szempontjából. Előzménye az *AutoML* projekt volt, ennek egy

továbbfejlesztéseként állt elő a Google ezzel a megoldással. [43] Használatát amiatt is indokoltnak tartották, hogy egyrészt mutat-e javulást az eddigi megoldásokhoz képest, másrészt pedig, hogy a már említett *ImageNet* képkategorizálásán magasabb, jobb eredményeket mutatott, mint az addigi, 2017. végi legjobbnak vélt, state-of-the-art módszerek mellett, hogy kevesebb műveletet végez el.



18. ábra NASNet modell pontossága pirossal kiemelve az összeadás-szorzás (multiply-add) műveletek szerint, látható módon a Mobile változata is közel 75%-os eredményt adva [43]

3. Feladat és specifikáció

3.1 A tanuláshoz vett adatok. Mi használható, mi nem használható?

A tanításban rész vevő minden egyes stílushoz legalább 1000-1500 darabos adathalmazra van szükség. Amennyiben ez a mennyiség 1000 darabban van meghatározva, úgy a tanítás alkalmával 90-10 arányban kerülnek szétosztásra a minták, miszerint az 1000 darab zene 90%-a, azaz 900 darab kerül a tanítóhalmazhoz, míg a fennmaradó 10%, 100 darab zene pedig a validációs halmazt fogja adni.

A tanítási és fejlesztési folyamat két fő részre oszlik el. Az egyik rész, amikor az internetről szedett adatok alapján kerülnek meghatározásra a lehetséges stílusok. Ezen kívül a másik, amikor nem áll rendelkezésre internet, offline módon kerül meghatározásra, képi adatok felhasználásával, spektrogramok feldolgozásán keresztül. A tanítás folyamata során a tanítás előtt használható az egyes zenékhez metaadatként hozzáadott stílusmegnevezés, ennek alapján kerülhetnek stílusok szerint mappákba szétosztásra a zenék, de a tényleges tanulás során nem vehető igénybe. Az online alapú zenemeghatározáshoz pedig szükséges a megadott, összegyűjtött zenékhez tartozó, hivatalos forrásból összeszedett adatok, valamint az ezekkel kapcsolatos statisztikai elemzés is, amelyből kimutatásra kerülhet, hogy az egyes stílusok mennyire különíthetők el egymástól, illetve mennyire egyeznek meg egymással, így meghatározva azt, hogy a tanítás ilyen módon működőképes állapotú lehet vagy sem.

3.2 Zenékből kiolvasható és megszerezhető tulajdonságok, azok elemzése

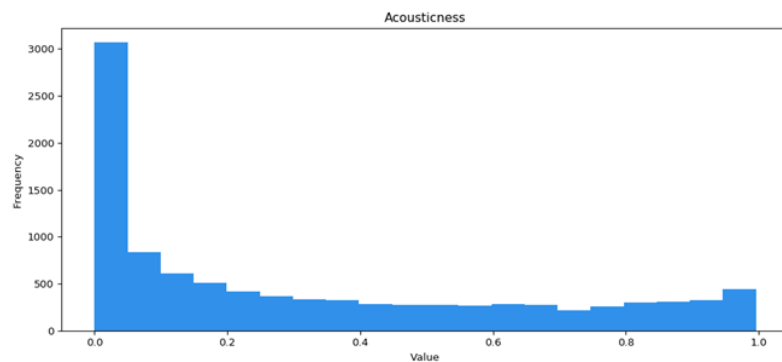
Minden zenei stílus neurális hálóval történő tanítása előtt adatokra lesz szükség, amihez az adott stílusú zenét különböző eszközökkel meg kell vizsgálni. Az egyik ilyen tulajdonság, ami mérhető és figyelembe vehető, az az egy percben lévő ütemek száma, másnéven BPM érték. A zenei világban temponak szokás nevezni az adott zeneszám részletének sebességét. Sebesség alatt az egyes hangszerek által megszólaltatott hangok számát lehet érteni, aminek jelzésére sokkal inkább használt a BPM, avagy beats-per-minute, ütem-per-perc jelzés. A BPM megmutatja, hogy egy adott percben hányszor szólal meg „erőteljesebben” egy hangszer által kiadott hang. Szemléletesebben kifejezve, hányszor lehet dobbantani lábbal egy adott percben úgy, hogy a dobbantás és a hang megszólaltatása szinkronban vannak egymással. [44] Például, ha egy zeneszámra jellemző BPM értéke 60, akkor az azt jelenti, hogy egy percben átlagosan 60 alkalommal lehet ezt a dobbantást elvégezni, 60 ilyen „beat” kerül elhangzásra. Azért átlagosan, mivel gyakran előfordul az az eset, hogy egy zeneszám lassú résszel indul el, ahol az ütem még meglehetősen lassú, ugyanakkor később magasabb ütemszámra vált, míg a végén pedig szintén lassabb levezetés zárja a számot.

A BPM értékek megadott zenék szerint a Spotify WebAPI szolgáltatásán keresztül kerülnek összegyűjtésre, valamint feldolgozásra. A fejlesztés során, a zenék összeszedését követően az először kitűzött 9 felismerendő stílus mellé további 6 kategória került, így a tanítás kezdetekor először 15 stílusra történt tanítás, minek utána,

annak nem túl hatékony működésének következményeként újból 9 stílus volt a meghatározott cél. Az adatok elemzése során 15 kategóriára vonatkozó adatok olvashatóak ki, ugyanakkor a stíluszám csökkenés után a megmaradt stílusokra ugyanazon tanítóadatok vonatkoztak, mint előtte is voltak.

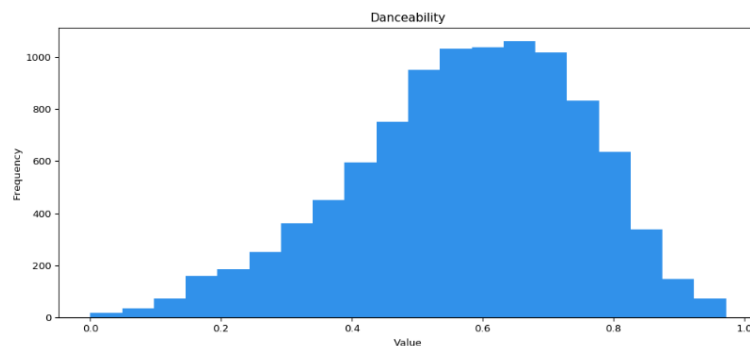
A BPM elemzése mellett számos más, a zenei stílusok felismerésében segítő adatok olvashatóak ki a WebAPI segítségével. [45] Ezen adatok az alábbiak:

- **acousticness:** Egy megadott zene akusztikusságát határozza meg. Akusztikus zenék az olyan zenék, amelyek akusztikus hangszerek által kiadott hangokat tartalmaz, azaz nincsenek bennük elektronikus formában létrehozott komponensek. 0 és 1 közötti értékeket képes felvenni, ahol az 1-es érték azt jelenti, hogy a választott zene nagy határozottsággal akusztikus.



19. ábra Az acousticness tulajdonság jellemző értékeinek eloszlása [45]

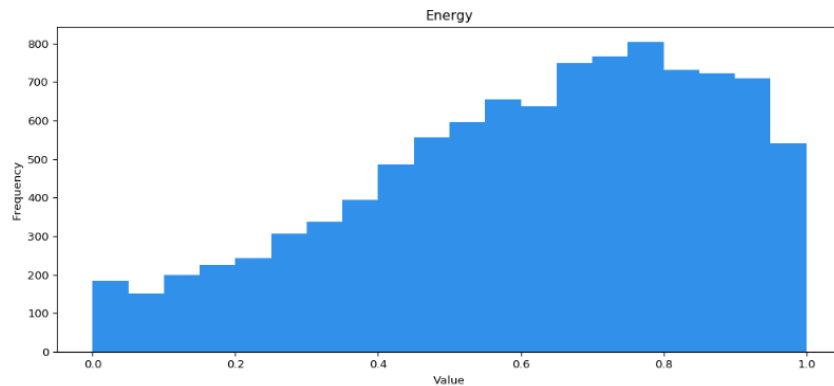
- **danceability:** magyar fordításban „táncolhatóság” jellemzőként hangzik, amely megmutatja, hogy egy adott zeneszám mennyire alkalmas arra, hogy táncolni lehessen rá. Több összetevőből lehet felépíteni, többek között szerepe van az imént említett BPM-nek is, de szerepel mellette a „ritmusstabilitás (azaz mennyire tart az adott zeneszám egy bizonyos ritmust), az ütem erőssége és a zenére jellemző szabályszerűsége.” [45] Egy 0 és 1 közötti, nem egész értéket ad eredményül, ahol a 0-s érték a legkevésbé táncolható, míg az 1-es érték a leginkább táncra alkalmas zenét jelenti.



20. ábra A danceability tulajdonság jellemző értékeinek eloszlása [45]

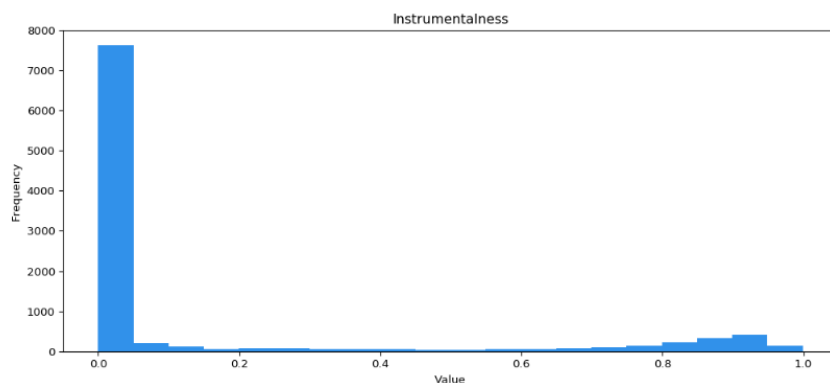
- **energy:** „egy adott zenére jellemző intenzitást valamint aktivitást figyelembe vevő érték”, amely 0 és 1 közötti lehet. Jellemzően azok a zenék, amelyek

energikusak, azok „hangosak, zajosak és gyorsnak hangoznak”, például a rockzene is ilyen. Az intenzitáshoz pedig olyan elemek tartoznak, mint „a zenére jellemző dinamikus tartomány, az észlelt hangosság, az onset ráta és az általános entrópia.” [45]



21. ábra Az energy tulajdonság jellemző értékeinek eloszlása [45]

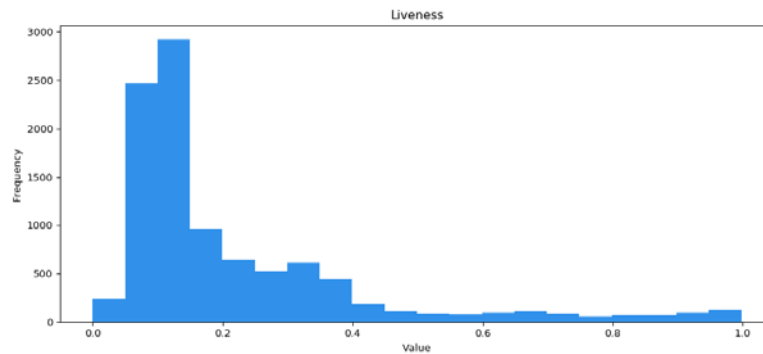
- **instrumentalness**: meghatározza, hogy egy adott zeneszám tartalmaz-e vokált avagy nem. 0 és 1 közötti értéket vehet fel lebegőpontosan, amennyiben 0 az értéke, az azt fogja jelenteni, hogy tartalmaz vokált, és minél közelebb van az 1-hez az értéke, úgy annál nagyobb a valószínűsége, hogy nincs benne vokál, azaz énekhang.



22. ábra Az instrumentalness tulajdonság jellemző értékeinek eloszlása [45]

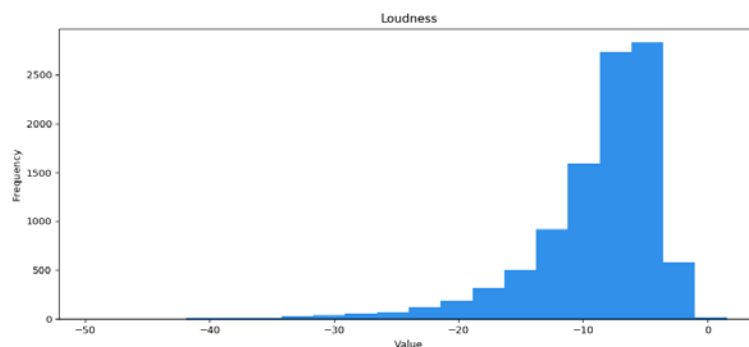
- **key**: egy adott zenére számított, várható zenei kulcs értéke. Maga a zenei kulcs jelen esetben összesen 12-féle értéket vehet fel a *pitch classok* értelmében, 0-tól kezdve olyan módon, hogy az egymást követő kulcsok közti különbség egy félhangot jelent. A fő hangok a C-D-E-F-G-A hangokat jelentik, a félhangok kapnak egy # jelölést, így épül fel a C-C#-D-D#-E-E#-F-F#-G-G#-A-A# zenei kulcsok létrája. [46] Amennyiben a WebAPI által szolgáltatott adat a -1, úgy azt jelenti, hogy nem lehetett egyértelműen meghatározni ezt a kulcsértéket.
- **liveness**: olyan tulajdonság, amely megadja, hogy található-e közönség, hallgatóság egy adott zenében vagy sem. Minél magasabb ez az érték, annál nagyobb annak valószínűsége, hogy „élőben került előadásra a zene”. [45] 0 és 1

közötti értékeket tud felvenni, „0.8-as érték felett erősen valószínűsíthető az élő előadásmód”.



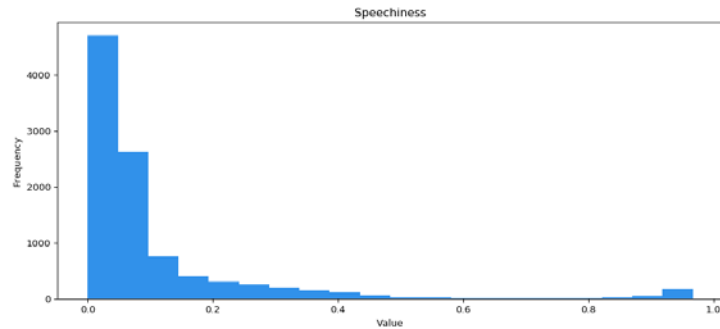
23. ábra A liveness tulajdonság jellemző értékeinek eloszlása [45]

- loudness: egy adott zenére vonatkozó „összes hangosság (loudness) decibelben kifejezve”. Az egész zenére vonatkozóan átlagolásra kerülnek értékei, ami mérőszámként az egyes zenék hangosságának összehasonlítására szolgál. A loudness „a zenének egy olyan jellemzője, ami az elsődleges pszichológiai korrelálása a fizikai erősségnek, azaz az amplitúdó értékének”. Értékei -60 és 0 decibel között mozognak.



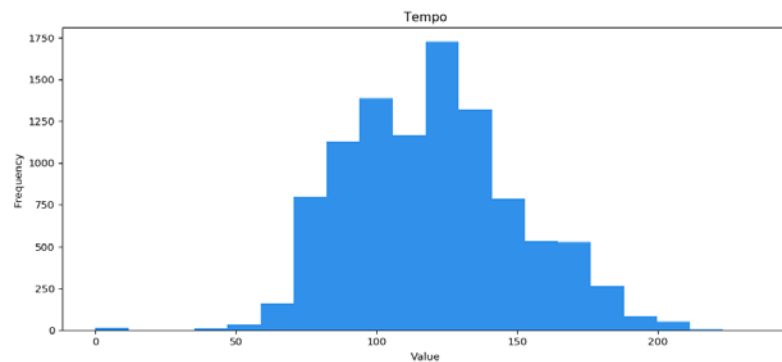
24. ábra A loudness tulajdonság jellemző értékeinek eloszlása [45]

- speechiness: Az egy adott zenében a „kiejtett, beszélt szavak jelenlétét” azonosító tulajdonság. 0 és 1 közötti értékeket képes felvenni, ahol minél inkább közelít az 1-es értékhez, annál több, annál magasabb a zenében megfigyelhető beszélt szavak aránya. „1-es értékek elsődlegesen talk-show-kban, verselésben figyelhető meg, ugyanakkor 1/3 és 2/3 közötti értékek esetén egyaránt jelen lehet énekhang és hangszerek által játszott zene is, amely a rap kategóriára lehet jellemző”. [45]



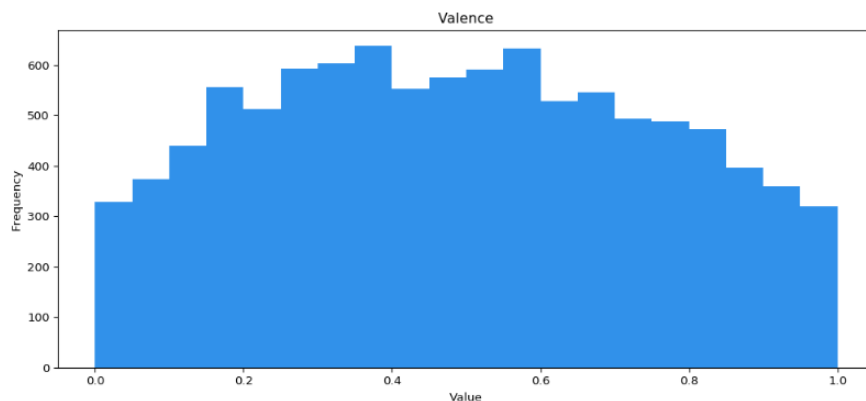
25. ábra A speechiness tulajdonság jellemző értékeinek eloszlása [45]

- tempo (BPM): A korábban említett fogalom meghatározás szerint az egy perc alatt megfigyelhető ütemek (beat-ek) egész zenére vonatkoztatott számértéke.



26. ábra A tempo (BPM) tulajdonság jellemző értékeinek eloszlása [45]

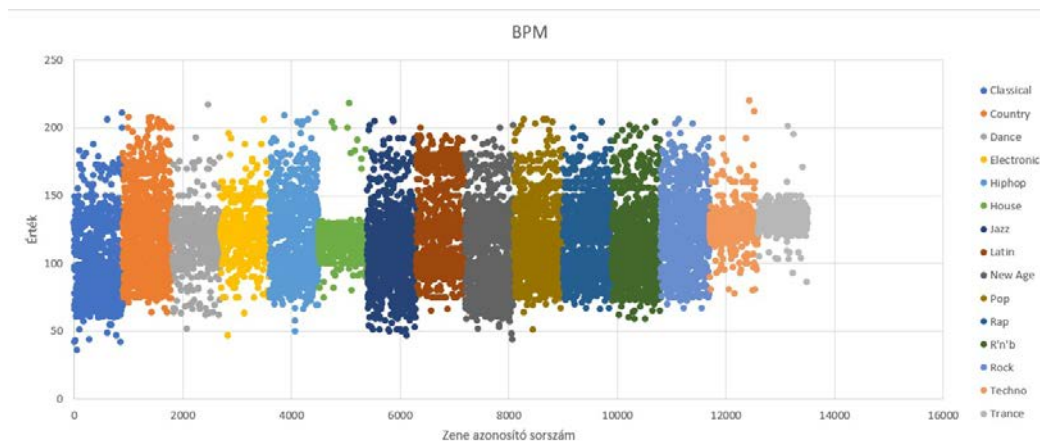
- valence: egy zene által közvetített, sugárzott zenei hangulat, pozitivitás leírására szolgáló jellemző, amely 0 és 1 között vesz fel értékeket. Minél magasabb ez az érték, annál jellemzőbb a pozitivitás, hangulati szinten például „öröm, boldogság, felszabadult érzet” társul mellé, az alacsonyabb értékek pedig minél inkább negatívakat mutatnak, így „szomorúságot, dühöt és feldúlt lelkiállapotot” jelezhet.



27. ábra A valence tulajdonság jellemző értékeinek eloszlása [45]

A fejlesztés folyamata során felmerült, hogy ezen, online alapú zenestílus-meghatározáshoz kerüljön saját programkód által meghatározva külön az ütem-perc, avagy BPM érték, vagy megfelelő a program szempontjából az, ha azon adatok is

a Spotify adatbázisából kerülnek meghatározásra. Itt ez utóbbi került alkalmazásra. Egyszerű lett volna ugyanis annak implementálása, hogy egy újabb oszloppal bővüljön az adatok halmaza minden zenére vonatkozóan. A fejlesztés folyamata során azonban szembetűnő volt, hogy jónéhány kategória esetén nem került meghatározásra annyi zene, valamint azokhoz tartozó egyéb tulajdonság, hogy ezt lehessen alkalmazni. Szükséges volt manuálisan lejátszási listákat keresni, majd az azokat alkotó zenék szerint az adatokat megkeresni és elmenteni. Ezen zenék fizikailag nem voltak megtalálhatóak tárhelyen, le kellett volna ugyanazon zenéket tölteni, a megfelelő beírt metaadatokkal együttesen, ami meglehetősen időigényes feladat lett volna. Továbbá számolni kellett volna azzal, hogy ezen zenékből, hogy BPM érték kerüljön kiolvasásra, azokat wav formátumban kellett volna beszerezni, vagy ha nem, akkor a megszerzett zenét kellett volna átkonvertálni ilyen formátumba, ami növelte volna tovább az amúgy sem alacsony tárhelyigényt. Annak érdekében, hogy ne fordulhassanak elő kiugró értékek, mind a BPM tulajdonságra, mind pedig a többi megállapított tulajdonság esetén pontdiagramon került ábrázolásra, ezt ellenőrizendő, korrigálandó, továbbá megfigyelendő, hogy az egyes stílusok mennyire tudnak egymástól elkülönülni adatok szintjén.



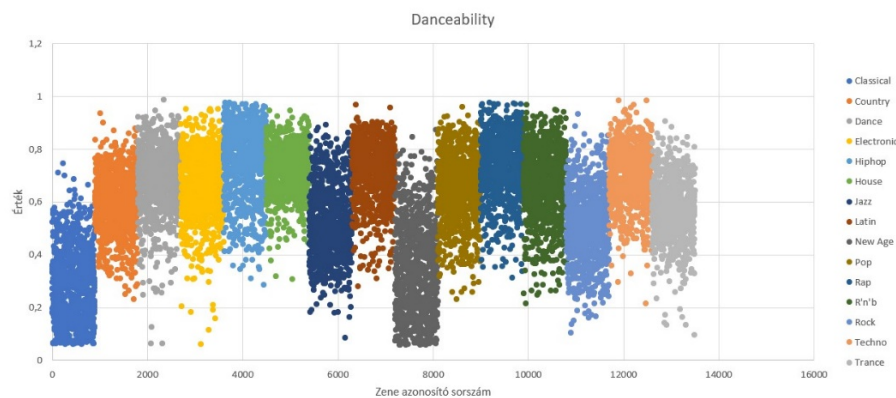
28. ábra A tanító adathalmazra vonatkozó BPM értékek pontdiagramon ábrázolva, egy-egy zenestílust körvonalazva, a külön stílusok külön színnel jelölve

Készült a BPM értékekre vonatkozóan egy statisztikai vizsgálat is, amelyben az egyes stílusokra jellemző átlag-, módusz-, medián-, minimum-, maximum-, valamint szórásértékek kerültek meghatározásra. Ezen értékekből, amennyiben már jelentkezik valamilyen adatok vonatkozásában különbség, úgy a stílusok között is lehet megkülönböztetést alkalmazni. Annak érdekében, hogy a móduszértékek valóságosak legyenek, úgy ebben a vonatkozásban egészen kerekítve lettek az értékek. Ennek oka, hogy az értékek három tizedesjegy pontossággal kerültek alapesetben eltárolásra. Így a leggyakoribb érték akár egy véletlenszerű érték is lehetett volna.

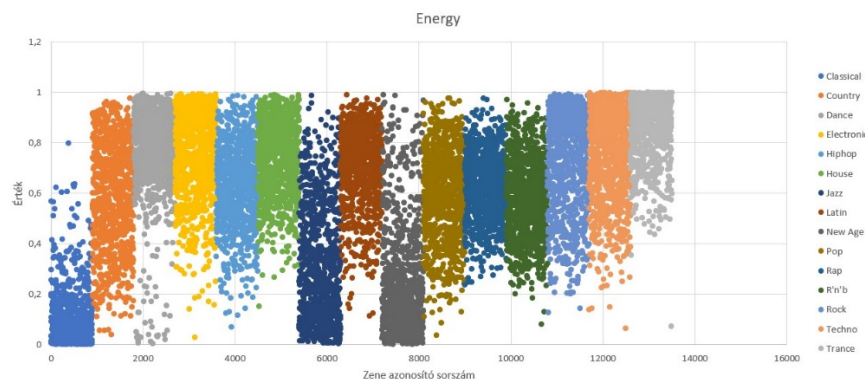
stílus	átlag	módusz(*)	medián	minimum	maximum	szórás
Classical	98,541	100	90,779	35,903	205,971	29,614
Country	123,756	120	121,854	63,620	208,426	30,714
Dance	121,176	125	124,039	52,021	217,333	15,842
Electronic	124,594	128	124,996	47,388	205,986	15,947
Hiphop	119,930	120	119,966	49,656	211,357	30,087
House	120,539	124	122,079	74,885	218,088	11,004
Jazz	110,114	110	109,392	46,950	205,539	29,423
Latin	122,899	120	109,507	64,921	199,706	32,891
New Age	109,116	120	107,143	43,513	201,535	32,039
Pop	117,388	105	113,860	51,414	205,846	27,363
Rap	120,362	120	121,946	66,005	204,185	27,302
Rock	113,381	100	107,901	59,041	203,862	28,697
Trance	125,345	142	123,804	66,853	205,544	28,616
R'n'b	127,094	128	126,009	78,001	220,017	11,751
Techno	133,615	138	133,997	85,743	200,991	7,872
átlag:	119,190	120	117,151	59,061	207,626	23,944

4. táblázat Statisztikai vizsgálat eredményei a 15 stílusra vonatkozóan

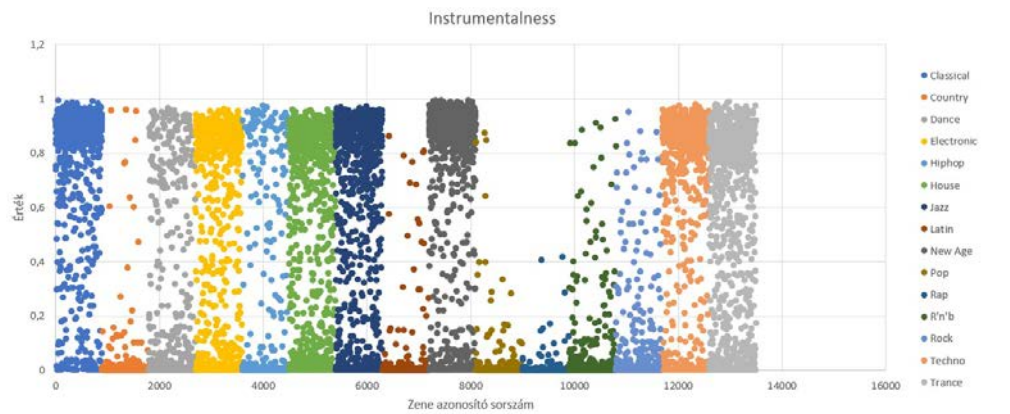
Észrevehető a 4. táblázat alapján, hogy egyes stílusok, mint *techno*, *r'n'b* és *house* esetén a szórás viszonylagos alacsony értékeket mutatnak a többi kategóriához képest, emellett, például a *techno* kategória esetén, ezen átlag is kiemelkedően magas a többi értékhez képest. Átlagot tekintve, amennyiben a BPM érték alacsony, úgy előzetesen utalhat *klasszikus*, *new age* vagy akár *jazz* kategóriára. Természetesen ezek csak feltételezések, hiszen egy-egy tulajdonság önmagában nem elegendő ahhoz, hogy egy zene stílusát meg lehessen határozni. A feladat lesz az, hogy az összes adatra vonatkozóan a jellegzetességek felismerésre kerüljenek.



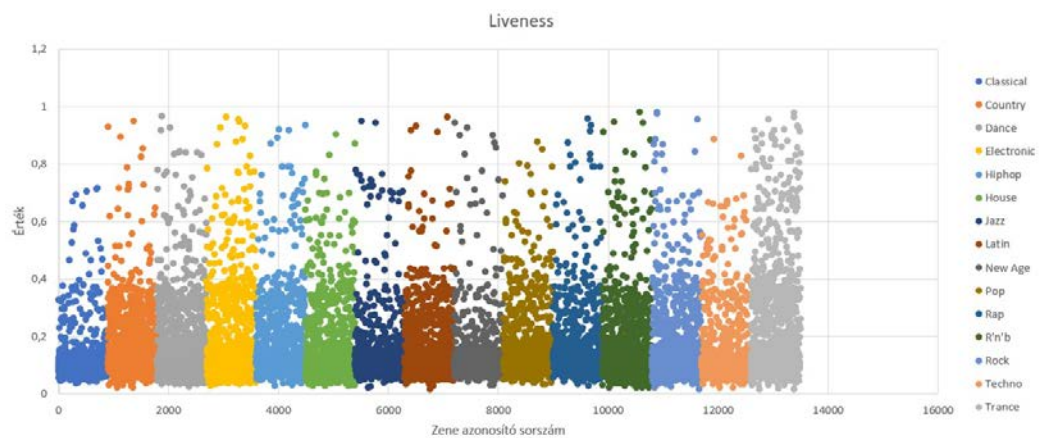
29. ábra Danceability értékek a 15 kategóriának tanítóhalmazára vonatkozóan pontdiagramon ábrázolva



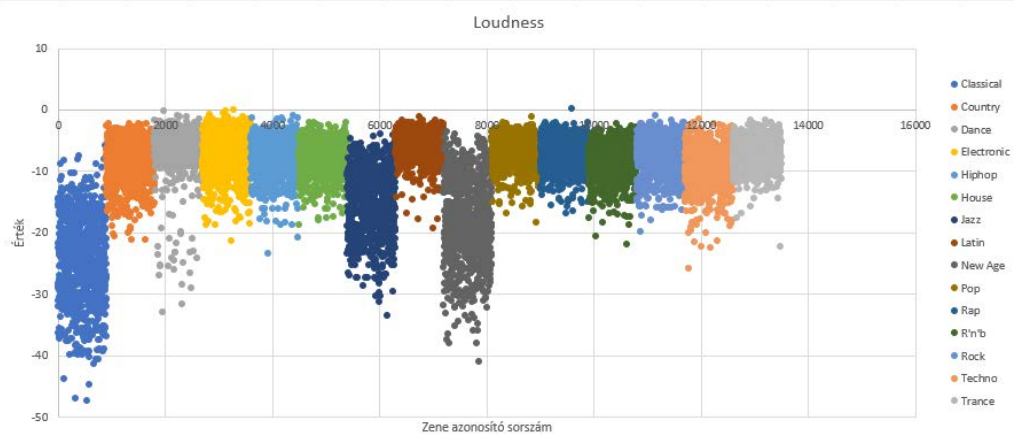
30. ábra Energy értékek a 15 kategóriának tanítóhalmazára vonatkozóan pontdiagramon ábrázolva



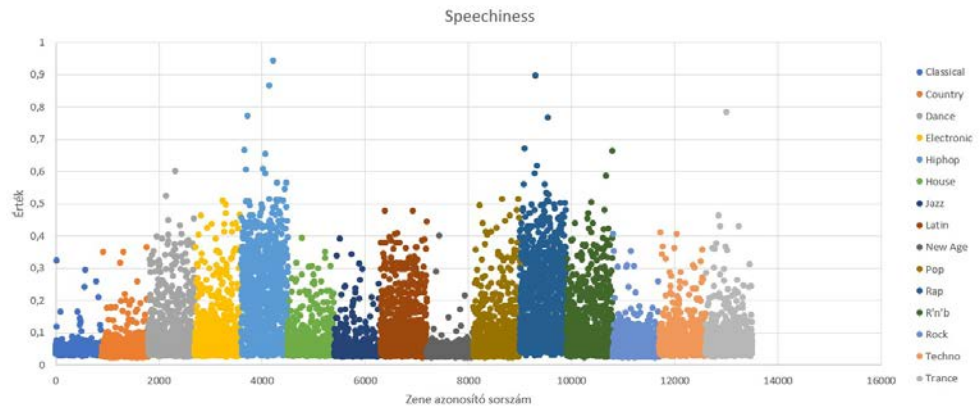
31. ábra Instrumentalness értékek a 15 kategóriának tanítóhalmazára vonatkozóan pontdiagramon ábrázolva



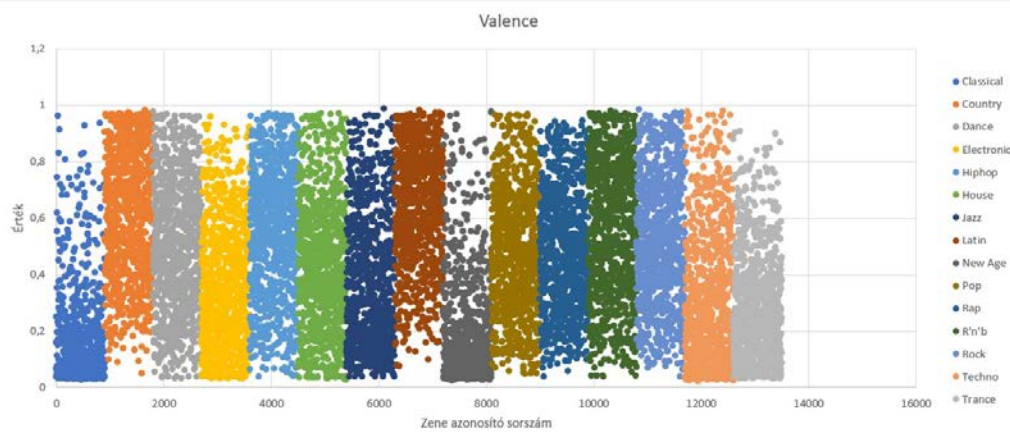
32. ábra Liveness értékek a 15 kategóriának tanítóhalmazára vonatkozóan pontdiagramon ábrázolva



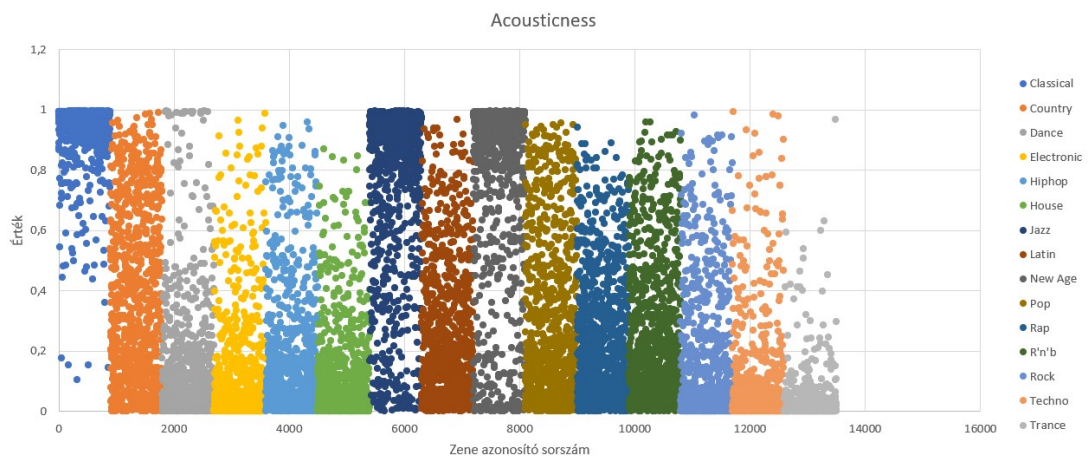
33. ábra Loudness értékek a 15 kategóriának tanítóhalmazára vonatkozóan pontdiagramon ábrázolva



34. ábra Speechiness értékek a 15 kategóriának tanítóhalmazára vonatkozóan pontdiagramon ábrázolva



35. ábra Valence értékek a 15 kategóriának tanítóhalmazára vonatkozóan pontdiagramon ábrázolva



36. ábra Acousticness értékek a 15 kategóriának tanítóhalmazára vonatkozóan pontdiagramon ábrázolva

Fontos lehet az adatok előkészítése szempontjából, illetve a tanulás hatékonyságának növelése érdekében, hogy összefüggő adatok ne legyenek jelen az adathalmaz tulajdonságai között. Ez hozzájárulhat a tanítási költségek csökkentéséhez is, kevesebb

adat feldolgozásából relatív gyorsabban lehet a tanítást véghez vinni. A tulajdonságok leírásából kapcsolatosan merült fel ennek többek között a lehetősége, amikor a különböző tulajdonságok egymásra hivatkoztak, például a *danceability* tulajdonság értékeinek kialakításában szerepet játszik a BPM érték is. Ennek felderítésére az összes, összegyűjtött adatra vonatkozóan készült egy korrelációanalízis. Korrelációanalízis egy olyan eszköz, amely esetén három vagy annál több változó esetén a változókat páronként vizsgálva feltételezve van, hogy a változók eloszlása a normáeloszlást követi, a változópárok közti összefüggés mértékét pedig a korrelációs együttható határozza meg. [47] A korrelációs együttható értékkészlete -1 és +1 közé tehető. Minél jobban közelít az együttható abszolút értéke az 1-hez, úgy annál nagyobb összefüggés figyelhető meg a változók között. Így célként fogalmazódik meg, hogy a változók korrelációs együtthatója lehetőleg minél kisebb legyen. A már említett zenékre az alábbi eredmények jöttek ki korrelációanalízis során:

	<i>acousticness</i>	<i>danceability</i>	<i>energy</i>	<i>instrumentalness</i>	<i>key</i>	<i>liveness</i>	<i>loudness</i>	<i>speechiness</i>	<i>tempo</i>	<i>valence</i>
<i>acousticness</i>	1,000									
<i>danceability</i>	-0,536	1,000								
<i>energy</i>	-0,818	0,480	1,000							
<i>instrumentalness</i>	0,320	-0,330	-0,324	1,000						
<i>key</i>	-0,063	0,042	0,070	-0,026	1,000					
<i>liveness</i>	-0,157	0,000	0,212	-0,104	0,027	1,000				
<i>loudness</i>	-0,719	0,562	0,822	-0,554	0,053	0,172	1,000			
<i>speechiness</i>	-0,165	0,283	0,133	-0,265	0,014	0,053	0,199	1,000		
<i>tempo</i>	0,014	-0,001	-0,010	-0,025	0,001	0,006	0,005	0,017	1,000	
<i>valence</i>	-0,294	0,485	0,380	-0,460	0,047	0,039	0,421	0,178	0,014	1,000

5. táblázat Korrelációanalízis eredménye a zenék adataira vonatkozóan

Az 5. táblázat szerint a főatlóban csupa egyes található, de ez természetes, hiszen egy adott változó önmagától 100%-osan függ, sokkal inkább fontosak a különböző változók egymáshoz viszonyított értékei. Viszonylag erős összefüggés figyelhető meg számos helyen, ezt jelzik az erősebb színek, a főatlót leszámítva. Legtöbb összefüggés az *acousticness*, *loudness* és *valence* kategóriák esetén volt megfigyelhető. Amennyiben ezen adatok eltávolításra kerültek a korreláció miatt, úgy a javított korrelációanalízis a 6. táblázatban látható eredményeket hozta.

	<i>danceability</i>	<i>energy</i>	<i>key</i>	<i>liveness</i>	<i>speechiness</i>	<i>tempo</i>	<i>instrumentalness</i>
<i>danceability</i>	1,000						
<i>energy</i>	0,480	1,000					
<i>key</i>	0,042	0,070	1,000				
<i>liveness</i>	0,000	0,212	0,027	1,000			
<i>speechiness</i>	0,283	0,133	0,014	0,053	1,000		
<i>tempo</i>	-0,001	-0,010	0,001	0,006	0,017	1,000	
<i>instrumentalness</i>	-0,330	-0,324	-0,026	-0,104	-0,265	-0,025	1,000

6. táblázat Javított korrelációanalízis eredménye a zenék adataira vonatkozóan

Az adatok eltávolítását követően a legnagyobb korrelációs együttható abszolút értéke nem érte el a 0.5-es értéket, ami a közepes korreláció egy mérőszáma. [48] A javított korrelációanalízisben nem szereplő értékek ugyanakkor nem kerültek permanens törlésre az adathalmazból, azok a tanítás megkezdése előtt kerültek a tanítóadatból kivételre az adatelőkészítés során.

3.3 Specifikáció

A fejleszteni kívánt rendszerrel kapcsolatban számos funkcionális és nem funkcionális tulajdonságot meg lehet említeni. Az alkalmazás funkcionális fő célja, hogy adja vissza felhasználója számára az általa bevenni kívánt zene lehetséges stílusainak valószínűségeit, ugyanis, említett módon, lehet, hogy egy adott zene több stílus jegyeit is magán hordozhatja. A felhasználó képes arra, hogy betallózzon egy, a számítógépén tárolt hangfájlt. Hangfájlok közül támogatásra kerülnek a wav, mp3, m4a, illetve flac formátumban megadottak. Ezek közül a spektrogram alapú meghatározás esetén támogatottak a wav, mp3 és flac formátumok, míg adat alapú predikció esetén az mp3, m4a és flac formátum az elfogadott.

Az illesztés és a predikció szempontjából elengedhetetlen, hogy a fejlesztés során figyelembe vett konfigurációs fájl módosításáról gondoskodjon a felhasználó, rendelkeznie kell egy megfelelő verziójú Python interpreterrel, valamint azon belül is adott csomagokat kell telepíteni a helyes működéshez.

A felhasználó dönthet arról, hogy milyen formában akarja választott zenéjének stílusát meghatározni. Rendelkezésre áll egy offline alapú meghatározási mód, amelynek alapja zenei képekből kialakított spektrogramok formájában fog megtörténni. Ebben az esetben a felhasználónak rendelkeznie kell egy említett formátumú zenei fájljal, amelyet szükséges betallóznia a programon belül. Ennek validálása megtörténik, egyrésztől megállapításra kerül, hogy ténylegesen a támogatott formátumú zenék valamelyikét adta át, másrésztől pedig, mivel a spektrogramok kialakítása a zeneszám 45. másodpercétől a 90. másodpercéig fog tartani, így azon zenének legalább 90 másodpercesnek, másfél percesnek kell lennie.

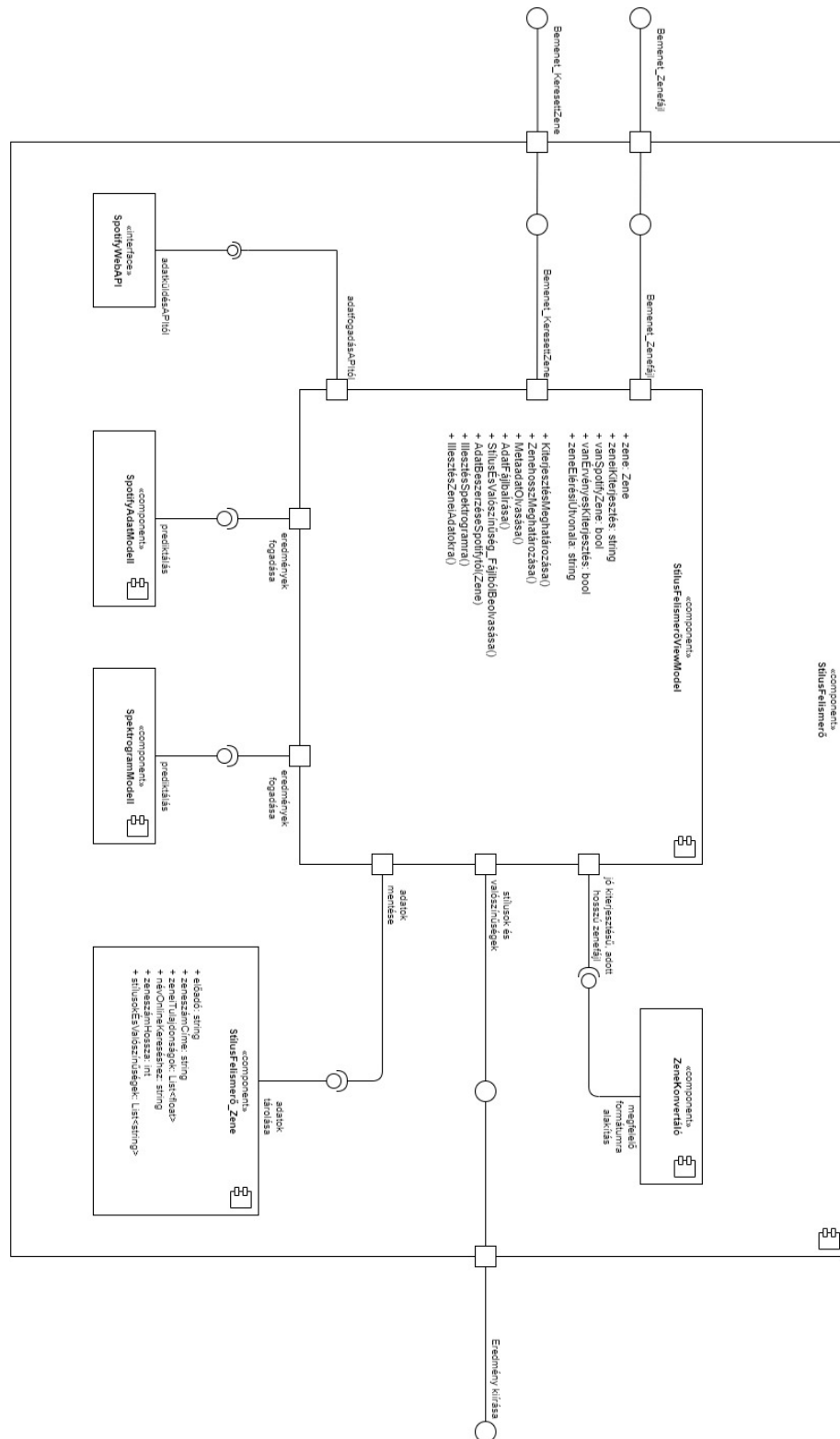
Amennyiben online alapú stílusmeghatározás történik, úgy mindenféleképp szükséges internetkapcsolat a helyes működéshez. Az adatok beszerzése a Spotify internetes adatbázisán keresztül történik, ezen adatok kerülnek illesztésre a tanított modellre. Zene bevitelét illetően kétfajta opció közül lehet választani. Amennyiben nincs meg adott számítógépen azon zene fájlja, amelyről online alapon kellene eldönteni a stílusát, úgy lehet keresni adott zenére, mintha Spotify-on történne meg az. Ez esetben az elsőnek egyező dal kerül azonosításra és feldolgozásra, amennyiben az megtalálható. Ha viszont megtalálható a számítógépen az említett zenefájl, úgy egyrészt megint szükséges a fájl validálása megfelelő formátum szerint, ami után nem hosszúság, hanem metaadat validálás történik. Metaadat ellenőrzés során kiolvasásra kerül a betallózott zene szerzője, amelyből ha több található, úgy az első helyen szereplő előadó kerül mentésre, valamint az adott zeneszám címe. Innen a háttérben történik meg a keresés, úgy, mintha a keresőmezőbe került volna. Fontos, hogy ez az eljárás sem garantálja azt, hogy arra helyes keresést lehet ejteni. A metaadat, mint olyan, kézzel módosítható tulajdonság, így nem megfelelő beírás esetén a megfelelő működés hibára juthat. Ha ez történik, de lehet tudni ez előadó-zeneszám neve párost, úgy keresést továbbra is lehet végrehajtani ugyanezen az ablakon. Sikeres keresést követően megadott zene stílusmeghatározáshoz

szükséges adatai kiírásra kerülnek tájékoztató jelleggel, majd megtörténik annak illesztése a megfelelő modellre, amely a gépi tanulás eredményeképp jött létre.

A gépi tanulás, mint a fejlesztési kulcsfolyamat egyike, rendelkezik egy-egy követelménnyel, amelynek meg kell felelni. A tanítás adatokon keresztül történik, amely olykor rendkívül komplex számításokat igényel, főleg, amikor spektrogram alapon, képek segítségével kerül meghatározásra a tanítás. Ehhez elengedhetetlen olyan videokártya megléte, amely lehetővé teszi a folyamat gyorsítását. Ez szükséges továbbá a spotifyos adatokra alapuló modell megalkotásához is, az adatok összetettsége noha lényegesen kisebb, ugyanakkor a tanítás itt is felgyorsítható ezáltal. Ez a felhasználó számára is elvárt követelmény lesz a predikciók elvégzéséhez.

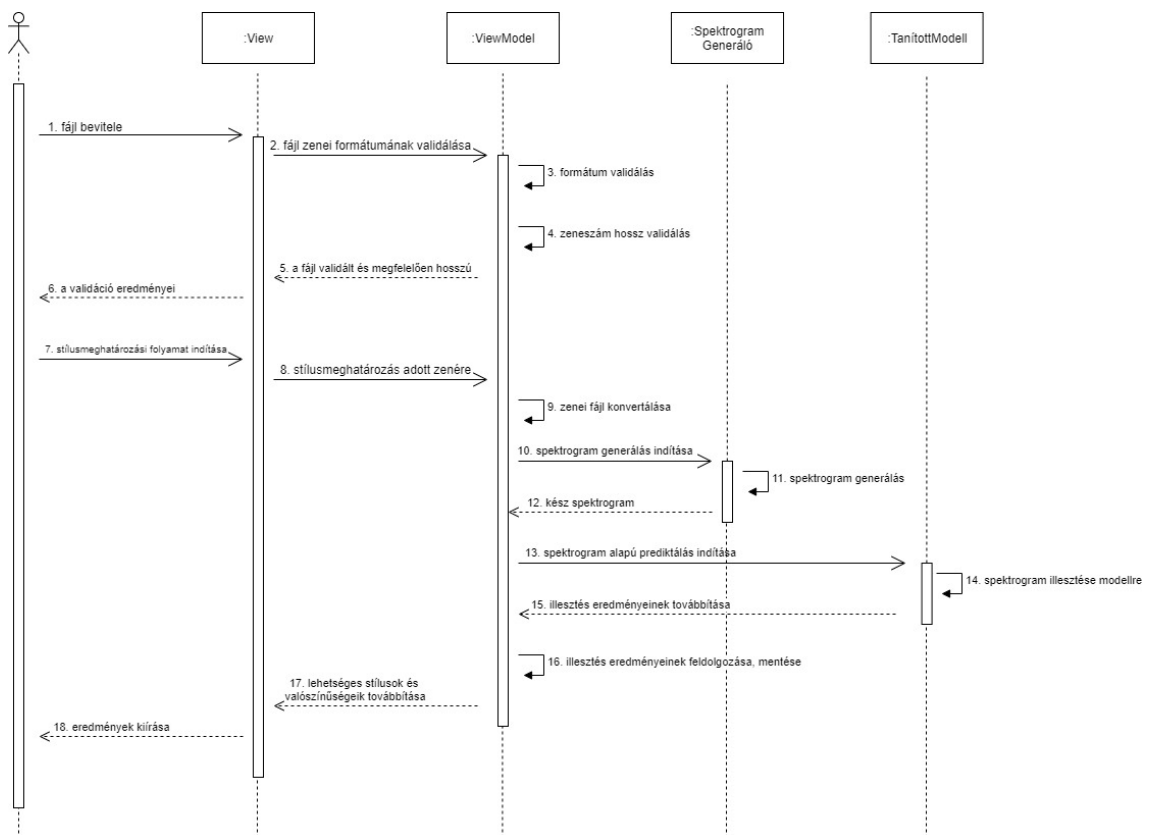
4. Tervezés

A rendszer alapját adó, illetve a különböző funkcionális és nem funkcionális követelmények grafikus szemléltetésére készült egy, a rendszert komponenseiből álló komponensdiagram, két időbeli szemléltetésre megfelelő szekvenciadiagram, és egy Gantt-diagram, amely a fejlesztés különböző fázisait időbeosztás alapon mutatja be.



37. ábra A rendszer komponens-diagramja

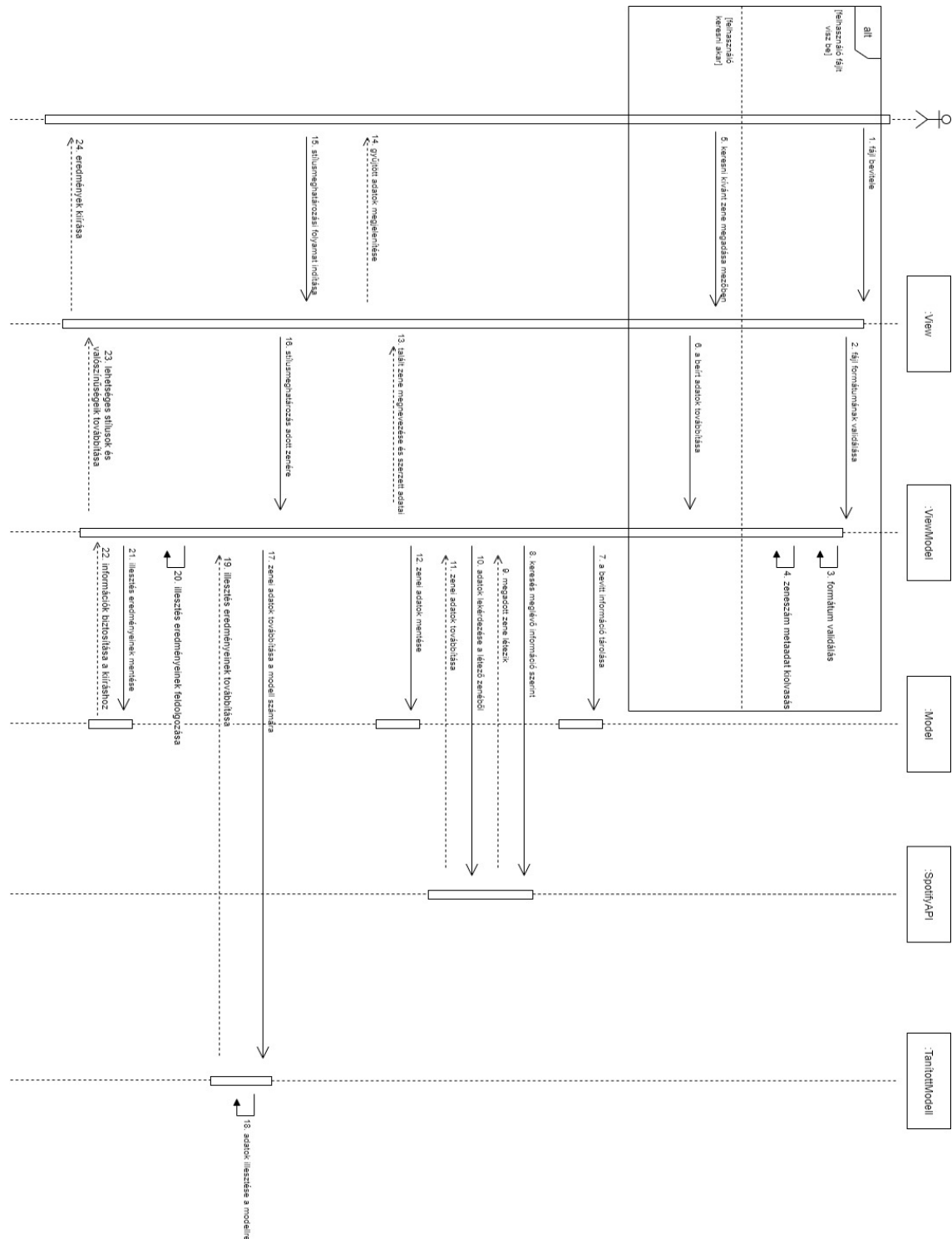
A rendszer komponens-diagramja szerint a felhasználó kétféle bemenetet tud biztosítani a rendszer számára. Amennyiben rendelkezik konkrét zenei fájljal, akkor igénybe tudja venni az online alapú zenestílus-felismerő részét a rendszernek, amennyiben az megfelelő metaadat beírással rendelkezik, illetve az offline, spektrogram alapú felét is. Ha nem rendelkezik a felhasználó zenei fájljal, hanem csak ismeretében van annak az előadónak, és ezen előadó zeneszámának címének, amire predikciót akar végrehajtani, úgy az is bemenetként szolgálhat. A bemenetet követően a megfelelő adatok feldolgozásra kerülnek, amelyből, ha már adottak megfelelő információk, úgy mint egy zene objektum létre is jön a memóriában. Spektrogram alapú felismeréshez szükséges a megadott zene átkonvertálása, amire már a bevitel pillanatában biztosítani kell, hogy annak hossza elérje, illetve meghaladja a szükséges 90 másodpercet. Ezen készített spektrogram illesztődhet a spektrogram modellre, aminek eredményeként kialakított stílusokat, valamint az azokhoz tartozó valószínűségeket, kiíratásra kerül a felhasználó számára. Ez, amennyiben Spotify adatokra hagyatkozik a felhasználó, úgy a Spotify WebAPI-ján keresztül a szükséges információ beszerzésre kerül, mire egy külön, adatokra felépített modell fogja a megfelelő valószínűségeket és stílusokat meghatározni.



38. ábra A rendszer szekvencia-diagramja egy offline alapú, spektrogram szerinti predikció esetén

A rendszer szekvencia-diagramja hasonlóképpen építhető fel és magyarázhatóak az egyes lépései, mint komponens-diagram esetén történtek. Amennyiben spektrogram szerint történik illesztés, úgy egy megfelelő fájl esetén (amely mind formátum, mind

időtartamát tekintve megfelelő) annak tömörítetlen, wav formátumra alakított, 45 másodperces részletéről kép készül. Az elkészült spektrogram illesztésre kerül a megfelelő modellre, aminek eredményei eltárolásra, valamint kiírásra kerülnek a felhasználó számára. A felhasználó egymás után több zenét is betallózhat, és hajthat végre rajta prediktálást.



39. ábra A rendszer szekvencia-diagramja egy online alapú, Spotify adatok szerinti predikció esetén

A rendszer szekvencia-diagramja a Spotify szerinti stílusdetektáláshoz egy elágazással nyit, hiszen az is megfelelő információforrás, ha a felhasználó maga üti be, kézzel az általa keresendő zenét, illetve az is, ha egy megfelelő formátumú zenefájlt tallóz be, amely rendelkezik többlet információ hordozására szolgáló metaadattal, amelyben benne van legalább az adott zene szerzője, és magának a zenének a címe. Bármelyik is adott, utána szükséges ezen adatokra egy keresés. Ha megtalálható a zene Spotify-on, akkor az adatbázisban is megtalálható 99,9%-os valószínűséggel. Az újonnan megjelenő, lejátszható zenék hamarabb vannak jelen, mint az azokhoz tartozó adatok az adatbázisban, így néhány óra késéssel, de megtalálhatóak lesznek az adatok. A meglévő információk tudatában megtörténhet az illesztés a modellre. Ahogy az a spektrogramos megoldás esetén is létezik, úgy itt is egymás után több zenére végezhet keresést és prediktálást a felhasználó.

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
Irodalomkutatás, információgyűjtés										
Technikák összevetése, döntés a fejlesztési módszerekről										
Tanításhoz vett zenék összegyűjtése										
A zenék szétosztását segítő automatizáló készítése										
Zenék elemzése, adatgyűjtése, szűrése										
A tanuló algoritmusok felépítése, elkészítése, pontosítása, eredmények elemzése										
A grafikus program elkészítése, kapcsolat kialakítása a felépített modellekkel										
Eredmények ellenőrzése, modellek tesztelése										
Értékelés, döntés továbbfejlesztési lehetőségekről										
Dokumentáció										

40. ábra A rendszer fejlesztésére vonatkozó Gantt-diagram

A rendszer fejlesztése során ezen szakaszok számítanak fontos pontnak, mondhatni mérföldkönek. Leghamarabb az információgyűjtés és az irodalomkutatás szakasza zárul le, ezt követően mindenképp döntés történik arról, milyen technikák kerülnek felhasználásra a fejlesztés során. Szükséges a zenék összeszedése, majd ezt követően az azokra vonatkozó adatok kigyűjtése, amelyek a rendszer tanító- és validációs mintáit szolgáltatják. Ezt követi maga a tanítás, aminek a végére elkészül a két modell, amelyekre a megfelelő bevitt adatokat illeszteni, összehasonlítani lehet. Ezen két, fő komponenst szükséges egymással összekötni, valamint összeköttetést teremteni így a felhasználóval is, a zenei adatokon keresztül. A fejlesztés végén kiértékelésre kerülnek a különböző modellek által adott eredmények, amelyek a tesztek során jöttek létre, valamint megfogalmazásra kerülnek továbbfejlesztési lehetőségek, amelyek a fejlesztés közben felmerülnek. A projekt egészére vonatkozóan nem maradhat el a teljes körű dokumentáció.

5. A fejlesztés folyamata

Az eredeti terveknek megfelelően magának a program elkészítésének nulladik lépése a zenestílusok felismertetéséhez szükséges zenék beszerzése különböző forrásokból. Fontos megemlíteni, hogy ezen zenék felcímkézése más, ugyanazon letöltőoldalt használó, illetve hallgató felhasználók tették meg, aminek ellenőrzésére sor került. Így ezen lépés volt, hogy a keresendő stílusokra keresés történt, hogy minél több zene kerüljön beszerzésre az adott stílusokkal kapcsolatosan. Ez elsődlegesen albumok formájában valósult meg először, amely albumot alkotó zenék metaadatában megtalálható az adott zenére jellemző stílus. Viszont ide kapcsolódóan még megemlítendő, hogy az oldalon megjelenített zenestílus még így is eltérhetett a metaadatba írt adatától. Természetesen feldolgozás során ez utóbbit vettem figyelembe.

Melodic House	Singer & Songwriter
Minimal	Soul And R&B
New Wave	Soundtrack
Reggaeton	SpaceSynth
Salsa & Tropical	Swing
Salsa y Tropical	World Fusion
Sconosciuto	Xmas

7. táblázat A zenék szétosztása a metaadatban szereplő stílus megnevezés alapján történt – néhány példa létrejött mappa elnevezésekből

Ezen metaadatokba írt stílusadatok alapján számos mappába szortírozásra kerültek a letöltött hanganyagok. Ezt követően, mivel több, relatív kevés, nem nagy és nem is keresett stílushoz kapcsolódóan kerültek ki zenék, úgyhogy egy újabb, de már manuális rendezésre volt szükség. Előfordulhatott olyan is, hogy egy adott zenének, zenéknek nem volt beírt metaadata stílusra vonatkozóan. Itt újbóli megkeresést követően program segítségével kerültek beírásra ezen zenékbe a stílusok, és azok úgy kerültek szétosztásra. Maga a metaadat, illetve annak kiolvasása és tárolása ID3 címkék, tag-ek formájában valósul meg. Alapvetően ID3v2 típusú címkék formájában történik meg a megfelelő információ hordozása, amely hangok, zenék esetén legnagyobb részt működőképes megoldást ad. Nemcsak a jelenleg ismert, szükséges adatok tárolására szolgál, hanem fel van készítve esetleges, további plusz információ mentésére is, ugyanis „információs blokkokban, úgynevezett frame-ekben van az információ eltárolva” [49], aminek „formátumát a feldolgozó szoftvernek nem szükséges ismernie, csupán azt, hogy mennyi az az adatmennyiség, amennyit át kell ugrania”, amire nincs szüksége lejátszás szempontjából, ezt „jelölő flag-ek, illetve az átugrandó információ mennyiségével adatik meg”. Programszinten ezt a *mutagen* nevű library *EasyID3* komponensével fel is lehet dolgozni. [50]

```
import glob
from mutagen.easyid3 import EasyID3

folder = 'E:\\Szakdolgozat\\CATEGORIZED\\Chanson'
files = glob.glob(folder + '/*.mp3')
print(files)

for i in files:
    audio = EasyID3(i)
    audio['genre'] = 'Jazz, Pop'
    audio.save()

print('Done')
```

41. ábra Példa egy beírt stílussal nem rendelkező mappán belüli zenék felcímkezésére. Itt minden, Chanson mappabeli zene ellátásra kerül Jazz, illetve Pop stílussal

A zenék szétosztását követően készült annyi új, nem tervezett stílusról is gyűjtemény, ami egyrészt egy újabb keresést eredményezett, kiegészítve a szükséges darabszámig az értékeket, másrészt pedig szó lehetett arról, hogy a 9 kategória helyett 15 kategóriára történjen tanítás.

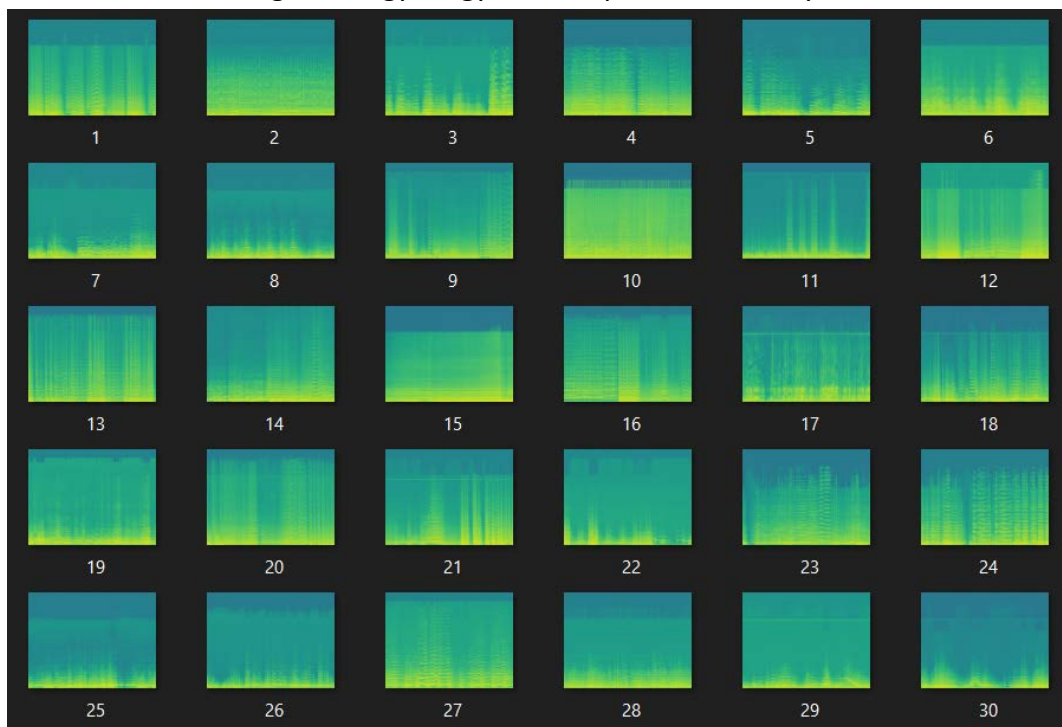
Zene stílusa	Darabszám	Hiány 1000-ig
Classical	879	-121
Country	200	-800
Dance	844	-156
Electronic	1505	505
Hiphop	271	-729
House	494	-506
Jazz	944	-56
Latin	583	-417
New Age	471	-529
Pop	2840	1840
Rap	971	-29
Rock	1184	184
Trance	686	-314
R'n'b	216	-784
Techno	360	-640

8. táblázat Az első letöltésekkel beszerzett zenék darabszáma, és a szükséges 1000 darabhoz lévő hátralévő mennyiség. A negatív értékek még hiányra utalnak, a piros színnel kiemelt stílusok pedig az újonnan felvett stílusok

Újabb kereséseket követően sikerült a kellő mennyiségű zenét összeszedni, mind a 15 kategóriához előállt a kategóriánkénti 1000 darab zene. Megkezdődhetett a zenék feldolgozása, mind az offline, képfelismerésen alapuló megoldáshoz, mind pedig a Spotify-on keresztül érkező adatok feldolgozásához.

Az offline stílusprediktálás spektrogramok segítségével történik. Kérdés volt, hogy ezen spektrogramok egy adott zene mely részét tartalmazza egybefüggően. A zenék elején sok esetben még csak egy felvezető rész található, amely nem sok énekhangot tartalmaz, amennyiben a zene maga ténylegesen tartalmaz, továbbá felmerülhet problémaként, hogy nincs megszólaltatva a hangszerek mindegyike. Ezzel párhuzamban a legvégéről sem érdemes, egyrésztől ugyanezen jellegzetességek miatt, a levezetés jelenlétével, valamint a zenék hosszára sincs egyértelmű meghatározás, nem minden

zene ugyanolyan hosszú. Így megközelítésemben azt határoztam el, hogy valahonnan a zene közepéről történjen mintavételezés. Ezen a részen nagyobb valószínűséggel már található vokál, és nagyobb valószínűséggel található refrénrész is, amely egy adott zene legintenzívebb, legjellegzetesebb szakaszának tekinthető. Felmerült, hogy a zene milyen hosszú legyen. A túl hosszú zene tartalmazhat ismétlődést, ha ugyanaz a refrén abban az időtartamban kétszer is előfordul, de túl rövid sem lehet, mert akkor vagy a vokál, vagy pedig a refrénrész rövidül valamennyire. Így lett meghatározva egy 45 másodperces időtartam, mégpedig a zene 45. másodpercétől a 90. másodpercéig. Fontos, hogy spektrogram csak wav formátumú fájlról készülhet tömörítetlen adottsága miatt, így ilyen formátumba kellett alakítani az amúgy nem ilyen formátumú zenefájlokat. Emiatt számolni kellett egyfajta veszteséggel, főleg, ha eleve már tömörített formátumból (például mp3-ból) került átalakításra. Az átalakítást követően jöhetett a spektrogramok elkészítése, majd pedig az elkészült képek átellenőrzése. A létrejött képek egyenként 3840x2880 felbontásúak lettek. Néhány kép a feldolgozás közben „megsérült”, aminek oka egyrészt a nem megfelelő hosszúságú zenéből adódhatott, utalva újból a zenéből vett mintára, a zenének magának legalább másfél percesnek kellett lennie. Továbbá bizonyos formátumok esetén, például az adatalapú predikció esetén elfogadott m4a formátumú zene feldolgozása ugyanúgy hibás képeket eredményezett.



42. ábra Példa elkészült spektrogramokról

Zenestílus	Összes spektrogram	Sérült spektrogram	Jó spektrogram
Classical	1041	10	1031
Country	1230	0	1230
Dance	1084	2	1082
Electronic	1478	2	1476
Hiphop	1067	12	1055
House	1072	0	1072
Jazz	1018	3	1015
Latin	1170	0	1170
New Age	1097	5	1092
Pop	2840	3	2837
Rap	1009	6	1003
Rock	1181	3	1178
Trance	1284	24	1260
R'n'b	1019	2	1017
Techno	1020	2	1018

9. táblázat Adott zenestílusokhoz készült összes spektrogram, hibás spektrogramok, és végleges spektrogram mennyiség, amely tanításhoz és validációhoz felhasználható

Az összegyűjtött zenékkel kapcsolatosan, mivel minden zenéhez összegyűlt a legalább 1000 spektrogram, így összehozhatóvá vált a pontos 90-10-es felbontás, miszerint 90% a tanítóhalmazhoz került, azaz 900 minta, míg 10% a validációs halmazhoz került, ami a fennmaradó 100 mintának felel meg. Lehetett volna egyes stílusoknál több zenét felvenni, ugyanakkor követelmény a tanítóhalmazra vonatkozóan, hogy egyforma arányban szerepeljenek bennük a tanító minták, azaz keresendő a legkisebb érték, és ahhoz igazítani a többi stílus mintáinak számát.

Egy külön C# program segítségével valósult meg a Spotify felől érkező adatok összegyűjtése, kiírása, megfelelő szétosztása. Mielőtt ez megtörtént volna, egy külön fájlba kerültek azon „előadó_neve zeneszám_neve” párosok, amelyekre keresés volt ejtendő. A nevek azonban nem voltak kellően megfelelőek ahhoz, hogy előálljon kellő mennyiségű adat. Minden zene saját metaadatából került beolvasásra, viszont előfordulhatott, hogy ezen adat alapján olyan összetett keresés jött létre, hogy a keresés maga Spotify-on nem vezetett eredményre, nem adott vissza semmilyen értéket. Előfordult olyan stílus is, hogy hiába volt hozzá több, mint 1100 zene, mégis adatként körülbelül 400 került összeszedésre. Az ezzel kapcsolatos adatok a 10. táblázat sorai között olvashatók.

Zene stílusa	Összegyűjtött adat	Összes zene	Elvesztett adat	Darabszám 1000-ig
Classical	609	1041	432	391
Country	929	1230	301	71
Dance	714	1084	370	286
Electronic	407	1478	1071	593
Hiphop	813	1067	254	187
House	596	1072	476	404
Jazz	217	1018	801	783
Latin	464	1170	706	536
New Age	730	1097	367	270
Pop	1309	2840	1531	-309
Rap	311	1009	698	689
Rock	438	1181	743	562
Trance	819	1284	465	181
R'n'b	763	1019	256	237
Techno	775	1020	245	225

10. táblázat Metaadatból talált zenék mennyisége stílusok szerint, összes összegyűjtött zene mennyisége, hibás zenék metaadat alapján, valamint a hiány az 1000 darab szükséges, stílusonkénti zenéhez

Ekkor összegyűjtési módot váltottam. Ezen keresések ugyanis többé-kevésbé spotifyos kereséseknek minősültek a letöltött zenékkel kapcsolatosan. Egy olyan megoldást véltem kitalálni, amelynek segítségével kész lejátszási listákból kigyűjtött zenéken keresztül kerülnek kikeresésre a különböző zenék adatai. Ez érvényes, sőt, kézenfekvőbb megoldásnak minősült, mivel ha egy lejátszási listában benne van, és az nyilvánosan kereshető, akkor az a zene meg is hallgatható, és meg is található az adatbázisban. Két dologra kellett leginkább odafigyelni. Az egyik, hogy az így keresendő zenék között ne legyen ismétlődő, ez viszont könnyen kezelhető volt, hiszen azonosító, ID alapján kerültek keresésre, így, ha egyezést talált, úgy még egyszer nem keresett rá a program. A másik dolog pedig az összegyűjtés előtti lejátszási listák összeszedése volt. Mivel a Spotify felkínál néhány zenei stílust, és azokhoz tartozó zenéket, így az egyszerűbb volt, illetve ha nem, akkor magára a stílusra kellett ejteni egy keresést. Ekkor ugyanúgy, mint letöltött zenék esetén, megvan a lehetősége a felhasználók általi hibás kategorizálásnak.

```
Paging<PlaylistTrack> playlistTracks = new Paging<PlaylistTrack>();
for (int i = 0; i < listOfAlbums.Count; i++)
{
    playlistTracks = apiForGet.GetPlaylistTracks(playlistId: listOfAlbums[i]);
    for (int j = 0; j < playlistTracks.Items.Count; j++)
    {
        if (playlistTracks.Items[j].Track != null && playlistTracks.Items[j].IsLocal == false)
        {
            listOfSongLinks.Add(playlistTracks.Items[j].Track.Id);
        }
    }
}
```

43. ábra Az összegyűjtött albumlinkeken végig iterálva összeszedendő azok zenéinek azonosítója, amennyiben azok Spotify-on és nem a felhasználók helyi tárhelyén található meg

```

for (int i = 0; i < listOfSongLinks.Count; i++)
{
    string dataRow = "";
    AudioFeatures songData = apiForGet.GetAudioFeatures(listOfSongLinks[i]);

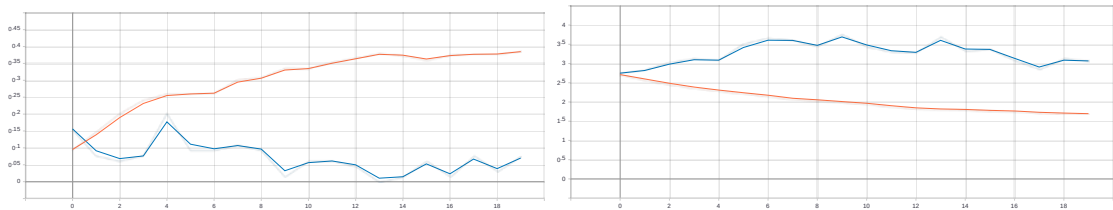
    dataRow += songData.Acousticness + ";" + songData.Danceability + ";" +
        songData.Energy + ";" + songData.Instrumentalness + ";" + songData.Key + ";" +
        songData.Liveness + ";" + songData.Loudness + ";" + songData.Speechiness + ";" +
        songData.Tempo + ";" + songData.Valence + ";" + style;

    if (songData.DurationMs > 60000)
    {
        Console.WriteLine("Datarow found for " + (i + 1) + "/" + listOfSongLinks.Count + " with length (ms): " + songData.DurationMs);
        dataMustWritten.Add(dataRow);
    }
}

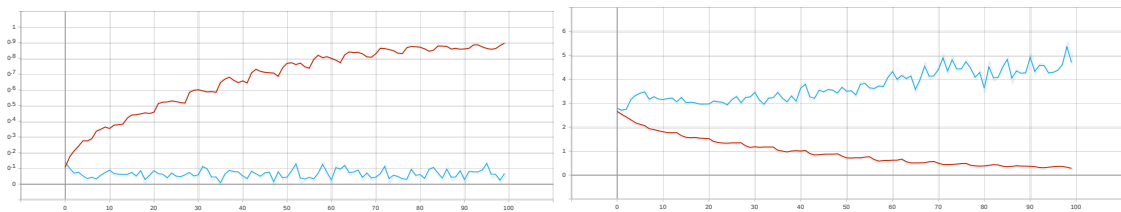
```

44. ábra Az összegyűjtött azonosítók alapján a megfelelő információk összeszedése, kigyűjtése, amelyek egy fájlba kerülnek kiírásra, ha annak hossza egy percnél nagyobb

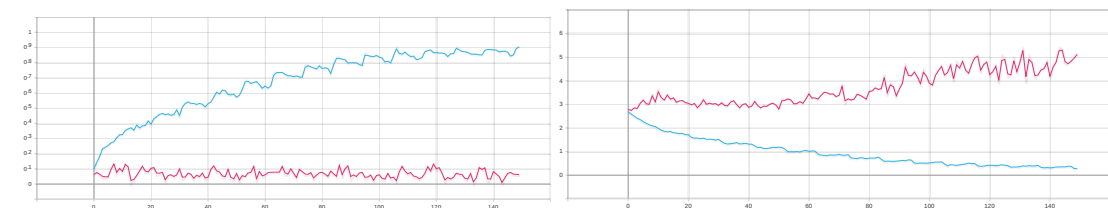
A szükséges információk megszerzése, rendszerezését követően megkezdődhetett az érdemleges tanítás, amelyet először az offline, spektrogram alapú részét érintette. Elsőnek megpróbálkoztam a legkevesebb paraméterrel rendelkező *MobileNetV2* modellel, többféle próbálkozáson keresztül. Ami fix volt ezeknél a tanításoknál, az úgynevezett *batchsize* értéke volt, ami alapvetően azt határozza meg, hogy egy időben a hálózaton hány minta kerül áthaladásra. Ez az érték ennél a modelltípusnál 64-es volt, 64 darab 224x224-es kép került eltárolásra, illetve a hálózaton keresztül tanításra bocsátva egyszerre egy adott iteráción belül. A tanítás először 20 *epochon* (azaz 20-szor érintve minden egyes képet, és kielemezve azok jellemző értékeit) keresztül történt, majd pedig, növelve ezen, egyszer 100-as, majd pedig 150-as érték lett kijelölve.



45. ábra Tanítás 20 epochon keresztül *MobileNetV2* architektúrával, a pontosság- (balra) és a veszteségértékek tekintetében. A narancs színű értékek a tanítóhalmazhoz, míg a kék színű értékek a validációs halmazhoz tartoznak.



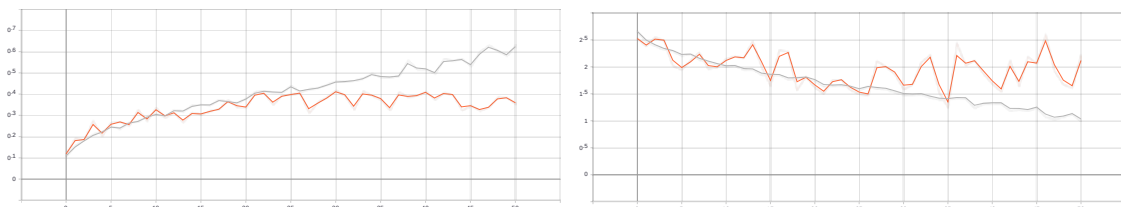
46. ábra Tanítás 100 epochon keresztül *MobileNetV2* architektúrával, a pontosság- (balra) és a veszteségértékek tekintetében. A bordó színű értékek a tanítóhalmazhoz, míg a kék színű értékek a validációs halmazhoz tartoznak.



47. ábra Tanítás 150 epochon keresztül *MobileNetV2* architektúrával, a pontosság- (balra) és a veszteségértékek tekintetében. A kék színű értékek a tanítóhalmazhoz, míg a lila színű értékek a validációs halmazhoz tartoznak.

A 15 kategóriára vonatkozó véletlen klasszifikátor érték megmutatja, hogy a klasszikus valószínűség szerint mekkora annak a valószínűsége, hogy egy zenét véletlenszerűen kiválasztva sikeresen, helyesen megállapítható annak stílusa. Ez jelen esetben 1/15-öd, azaz körülbelül 6,67%. A *MobileNetV2*-vel történt tanítások számos alapkövetkeztetést engednek. Egyrészt a tanítási halmazzal sikeresen lehet tanítani, a tanító pontosságok a 100, illetve 150 epochon keresztüli tanítások alkalmával 90% felettiek amellet, hogy veszteségértékei folyamatos csökkenést mutatnak. A probléma a validációs halmazzal adódik. A validációs halmaz pontossága ugyanis nem halad együtt, még egy darabig sem a tanító pontossággal, az beáll a klasszifikátor értékéig, akörül oszcillál. Ez a tanító halmazra vonatkozó *overfitting*, túlillesztés jelenségét mutatja, a tanító halmaz példáit jól megtanulja, ugyanakkor a validációs mintákra nem működik.

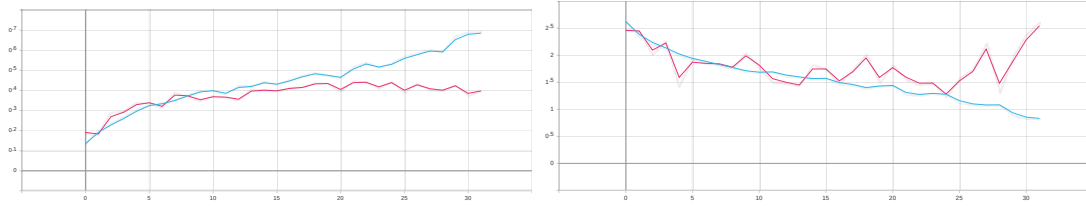
Folytatva, és rátérve a *VGG16* modellhez többfajta változata ki lett próbálva a stílusokra vonatkozóan. Számos közülük még a tanítás megkezdéséig sem jutott el. Ennek oka az esetek jelentős részében *OOM*, azaz *Out Of Memory* kivétel volt, normalizált adatok esetén is. Nem fért rá ugyanis a videókártyára a tanításhoz szükséges értékek mindegyike, annak ellenére sem, hogy a spektrogram alapú tanításra egy *Geforce GTX1080* videókártya lett felhasználva, amely 8 GB dedikált memóriával rendelkezik. Egy kivételes eset volt, amikor *SGD* optimalizálóval egyfajta tanulás elindult, viszont ennél már változtatva volt az alap architektúra. Minden esetben a *filterszám* változatlanul maradt a *VGG16* értékeihez képest, ugyanakkor egyre kevesebb konvolúciós réteget tartalmazott, de mindegyik *pooling* réteg előtt legalább egyet megtartva.



48. ábra *VGG16* architektúrájának egy egyszerűsítésével megkezdett tanítás eredményei a pontosság (balra) és a veszteségértékek tekintetében. Szürke színnel jelölve a tanító halmazhoz, míg narancsszínnel jelölve a validációs halmazhoz tartozó értékeket.

Ahogy az a *MobileNetV2* esetén is volt, itt is a 20. epoch környékétől egyfajta *overfit* figyelhető meg. Az *overfitting*, mint túlillesztés szerint a modell a tanítóhalmaz elemeit egyre jobban „megismeri”, ugyanakkor a mutatott validációs halmazra a fejlődés nem folytatódik akkora ütemben. [51] A validációs érték kb. 40%-on áll, ahogy azt a 49. ábra mutatja, ami már javulás az előzőekhez képest, sőt, a véletlen klasszifikátorhoz képest is nagyjából 6-szoros javulást eredményezve. Ugyanakkor az 50%-tól, ami kezdetben meg lett határozva mint első nagyobb cél, még elmarad tőle.

További egyszerűsítést megfogalmazva, az utolsó *pooling* réteget követő *fully-connected* neurális hálózatban, *dense* rétegekben került csökkentésre a benne szereplő egységek számának csökkentése, az eredeti 4096-os értéket alacsonyabbra, de olyan értékeket választva, amelyek 2 hatványai.



49. ábra VGG16 architektúrájának egy egyszerűsítésével megkezdett tanítás eredményei a pontosság (balra) és a vesztésgértékek tekintetében, Adam optimalizálással. Kék színnel jelölve a tanító halmazhoz, míg rózsaszínnel a validációs halmazhoz tartozó értékeket.

Az eredmények ismét javulást mutattak, közel 45%-osak. Annak érdekében, hogy a jelenleg meglévő adatok ellenőrzésre kerüljenek, végrehajtásra került függetlenül egy, a validációs halmazra vonatkozó illesztés, amelyben meghatározásra kerülhetett, hogy milyen stílusok kerülnek már döntő többségben felismerésre, és melyek azok, amelyek háttérbe szorulnak.

```
def decoding_Predictions(toPredict):
    listForPreds = ["Classical", "Country", "Dance", "Electronic", "HipHop",
                    "House", "Jazz", "Latin", "New Age", "Pop",
                    "Rap", "R'n'b", "Rock", "Techno", "Trance"]

    toPredict = toPredict.ravel()
    toPredict = toPredict.tolist()

    for count in range(3):
        MAX = max(toPredict)
        maxIndex = toPredict.index(MAX)
        gotStyle = listForPreds[maxIndex]
        print(str(count + 1) + ". legvaloszinubb mufaj: " + gotStyle + ",
              ekkora esellyel: " + str(round(MAX * 100)) + "%")
        toPredict.remove(MAX)
        listForPreds.remove(gotStyle)
```

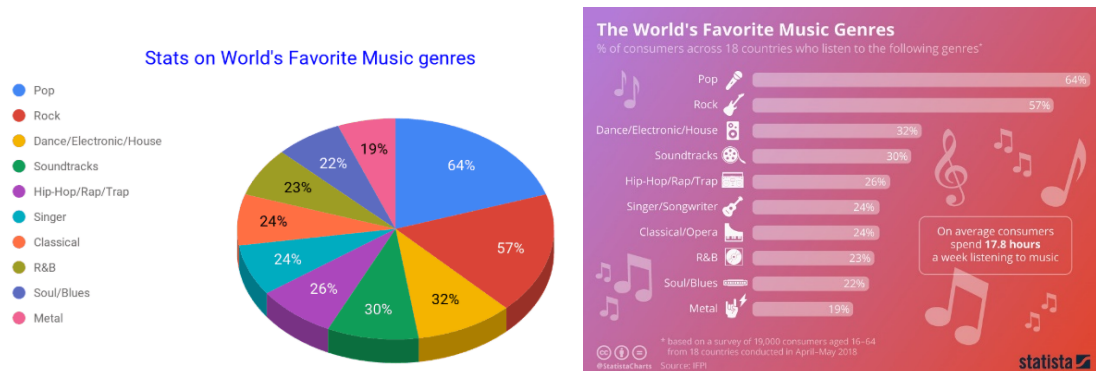
50. ábra Stílusok meghatározásához dekódoló kódrészlet 15 kategória esetén, amelyben paraméterül az utolsó, softmax réteg által adott eredmények kerülnek megadásra, abból a legjobb 3 kerül a felhasználó számára kijelzésre

Az így kapott eredményekből a legutolsó modellt felhasználva, az alábbiak kerültek kiolvasásra:

- Nagyon gyakran fordult elő az első 3 kategóriában a *trance*, *techno*, *hiphop*, *dance*, *house* stílusok
- Többé-kevésbé jelentek meg a *new age*, *electronic*, *rock* stílusok
- Ugyanakkor a többi, nem megnevezett kategóriák esetén nagyon ritkán, vagy abszolút nem kerültek kiadásra, holott olyan stílusú zenékre is történt prediktálás

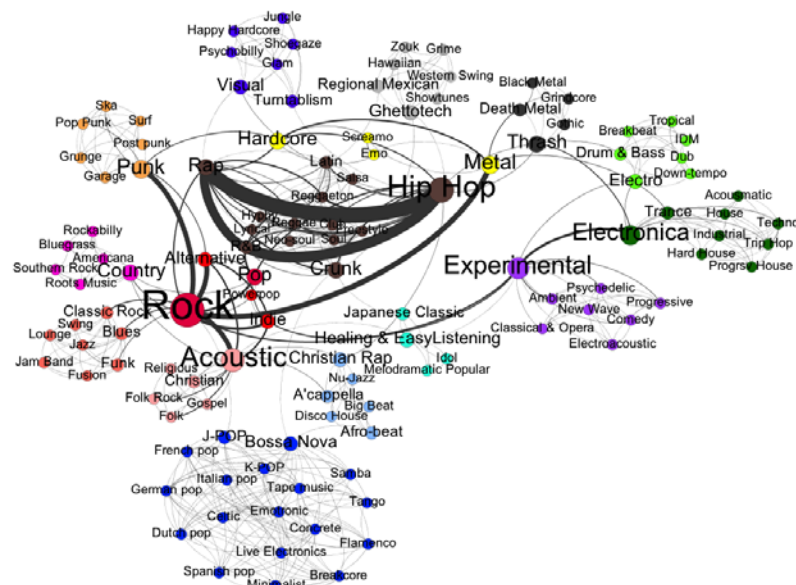
Amikor a felismerendő zenestílusok listája kibővült, úgy számolni kellett esetleges pontosság csökkenésre. Ez már a véletlen klasszifikátorból is várható volt, ugyanis 15 kategória esetén 6,67%-kal lehet egy zene stílusát meghatározni, ez az eredeti 9 darab kategória esetén 11,11%, ami több, mint másfélszer annyi. Felmerült következő lépésként, hogy a modell jelenlegi architektúráját tovább nem egyszerűsítve, más megoldást keresni.

Egyfajta megoldásnak tűnt a hasonló kategóriák eliminálása mindkét halmazból. A zene stílusok meghatározása továbbra is meglehetősen szubjektív, más-más vélemények és csoportosítások alakulnak ki azokról.



51. ábra Különböző statisztikai meghatározások az egyes stílusokra vonatkozóan, leggyakrabban, illetve legtöbbet hallgatott zenék stílusai szerint [51] [52]

Egy tanulmány során, amelyben az egyes stílusokkal kapcsolatos háttérinformációkat, keresettségeket, illetve szociológiai adatokat dolgozták fel. [53] Kutatásuk során azt figyelték meg, hogy a különböző előadók a MySpace nevű közösségi oldalon milyen stílusú zenét játszanak a felhasználók, előadók saját állításuk szerint, illetve ha már használnak profiljukon egy ilyen stílusmegnevezést, azt milyenek mellett teszik meg.



52. ábra Stíluscsoportosulások a MySpace közösségi oldalon aktív zenészek, előadók között, csomóponti színnel jelölve a nagyobb csoportokat, továbbá az élék vastagsága a stílusok együttes szereplésének gyakoriságával arányos [53]

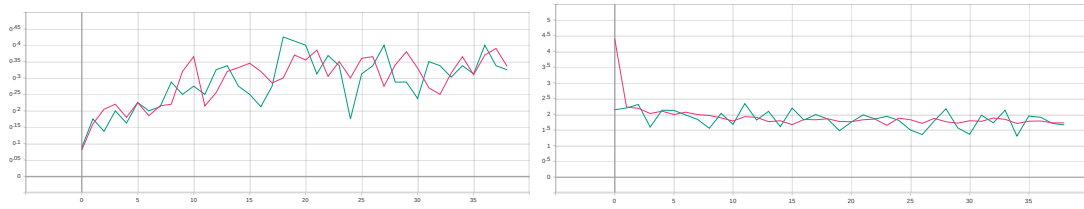
A statisztikai kimutatásokból, valamint az egyes zenestílusok közötti kapcsolatokról adódóan az alábbi döntések kerültek meghatározásra a felismerni kívánt stílusok 9 kategóriára történő visszacsökkentésére:

- Az *elektronikus* kategória megtartása, amellyel eliminálásra kerül a zenék felépítéséből adódóan a *trance*, *techno*, *dance*, *house*, *new age* stílusok, mivel azok is elektronikus zenének tekintendők készítésükből adódóan
- Az *r'n'b* stílus eliminálása, a *hiphop* kategória megtartása. A *hiphop* több értelemben hasonlít a *rap* kategóriára, figyelembe kell venni, mekkora zavart okoz ez majd a rendszerben, amennyiben okoz

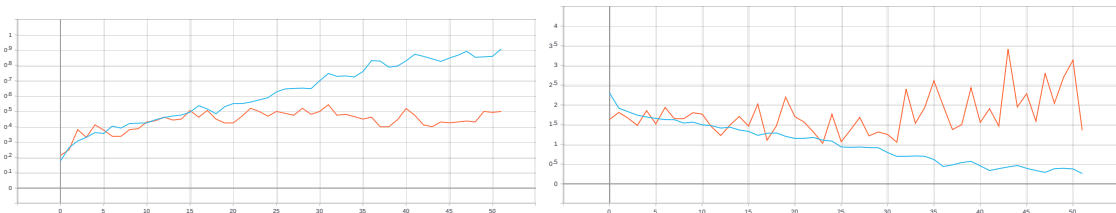
A stílusok kivételét követően megmaradt kategóriaként a *classical*, *country*, *electronic*, *hiphop*, *jazz*, *latin*, *pop*, *rap* és *rock* stílusok.

Visszatérve a modellhez, elindítva a tanítást, a tapasztalat azt mutatta, hogy a tanítás ezen adathalmazra nem indult meg, folyamatosan az immár 11% körüli pontosság körül ingadozott mind a tanító, mind a validációs halmazra vonatkozóan. Az optimalizálók közötti váltogatás, valamint az egyes optimalizálók paramétereinek változtatása nem vezetett eredményre. Veszteségértékekre vonatkozóan a tanító minták esetén, ahogy a pontosságnál is, egy adott értékre beállt, míg a validációs veszteség ingadozó értékeket mutatott. Ez utalhatott egyrészt arra, hogy nemhiába adott eredményül a modell nagyon gyakran csak adott típusú zenéket, merthogy azok hasonlóak voltak egymáshoz, illetve nagyon sok más stílus felett domináns stílusnak voltak tekinthetők. Másrészt, ennél az adathalmaznál utalhatott a *dying ReLU* jelenségére. [54] Az alap modell architektúra értelmében ugyanis minden egyes konvolúciós réteg aktivációja *ReLU* aktiváció, amelyek a tanulás során „elhalnak”, nem lesznek képesek az összefüggések megfigyelésére. Ennek következtében az egyes rétegek aktivációja le lett cserélve *LeakyReLU* aktivációra, törekedve arra, hogy annak *alpha* paramétere ne legyen túl magas, ne fogadjon el túlságosan nagy negatív értéket a tovább haladáshoz. Eleinte csak néhány rétegnél lett kiváltva ez, viszont ez sem vezetett eredményre, továbbra is az előbbiekhöz hasonló értékeket produkált, viszont ha minden helyen a *ReLU* ki lett váltva *LeakyReLU* aktivációval, úgy már megtörtént a tanulás megindulása. Emellett, az *overfitting* jelenségének elkerülése érdekében regularizáció került a hálózatba, az egyes konvolúciós-maxpool rétegek között *Dropout* lett alkalmazva. A *Dropout* pont az *overfitting* jelenségét próbálja meg elkerülni. Maga a dropout jelenti, hogy „az egyes egységek kimaradásra, kikerülésre kerülnek, mind a látható, mind a rejtett rétegeknél a neurális hálózatban. Azáltal, hogy egy egység kiejtésre kerül, még nem jelenti azt, hogy végleg eltávolításra kerül, csak ideiglenesen. A kiejtéssel a bemenő és kimenő kapcsolatok is kiejtésre kerülnek.” [55] A kiejtendő egységek meghatározása véletlen szerint történik, paraméterül át kell adni egy $[0,1]$ közötti számot, ami egy „ p valószínűség lesz, amely megadja, hogy egy adott egység mekkora valószínűséggel kerülhet kiejtésre”.

Regularizációval, illetve *LeakyReLU* aktivációk segítségével az alábbi eredmények születtek, feltüntetve a leghatékonyabb megoldásokat. Mindegyik esetben az optimalizáló az *Adamax* nevű volt, az volt mind közül a leghatékonyabb.



53. ábra 9 kategóriára vonatkozó tanítás eredményei pontosság- (balra) és veszteségértékek szerint, rózsaszínnel jelölve a tanítóhalmazra, zöld színnel jelölve a validációs halmazra vonatkozó értékek, ahol a batchsize 8-as értékűre választott



54. ábra 9 kategóriára vonatkozó tanítás eredményei pontosság- (balra) és veszteségértékek szerint, kék színnel jelölve a tanítóhalmazra, narancs színnel jelölve a validációs halmazra vonatkozó értékek, a batchsize 16-os értékű

Utóbbi esetben, amikor nagyobb *batchsize* lett választva, látható, hogy sokkal jobb a tanítás eredményei, ahogy a lépésekkel előrébb halad a tanítás. Ahol ez az érték kisebb, ott viszont az látható, hogy az értékek egymás mellett haladnak, ugyanolyan ütemben növekszik, csökken, illetve maga a validációs veszteség a tanító veszteségértékei körül oszcillál. A második tanítás eredménye a validációs halmazra a legjobb esetben elérte az 50%-ot, nagyjából 55%, amely pontnál a tanító pontosság 75%-os. Ez is ugyanakkor egyfajta overfittinget mutat, de magasabb validációs pontosság mellett.

Magában a programban nemcsak a legjobbnak vélt kategória kerül kiíratásra, hanem a legjobb 3, mégpedig a kapott valószínűségek szerinti legjobb 3. Így tovább vizsgálándó, mekkora ez a pontosság abban az esetben, ha nem az a cél, hogy csak legjobbként van elfogadva a megoldás, hanem akkor is, ha az első 3 stílus közé sorolandó. Továbbra is fontos megemlíteni való, hogy egy adott zene nem csak egy stílus jegyeit hordozhatja magán. Előfordulhat ugyanis, hogy több stílusjegyeket köszönhetően nem feltétlen az első helyre pozicionálja sem a modell, sem pedig a felhasználó az elvárt stílust, hanem csak a második vagy harmadik helyre. Feltételezve azt, hogy ha már kiírásra kerül az elvárt stílus, magasabb pontosságot lehet eredményként kapni.

```
counter = 1
for i in files:
    pngPath = i
    img = image.load_img(pngPath, target_size=(224, 224))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)

    for iw, ix, iy, iz in np.ndindex(x.shape):
        x[iw, ix, iy, iz] = x[iw, ix, iy, iz] / 255

    preds = model.predict(x)
    print(str(counter) + ". zene stílusai:")
    decoding_Predictions_9categories(preds)
    counter += 1
```

55. ábra Zenestílusok, valamint annak valószínűségeik kiíratása egy adott zene halmazra, amelyek elérési útvonala a files-ban található, normalizált értékeként vizsgálva

A validációs adathalmazon történő átfuttatás során az 57. ábra által mutatott eredmények jöttek ki.



56. ábra Első stílusként, második stílusként, illetve harmadik stílusként megtalált zenestílusok darabszáma, valamint százalékos aránya VGG16 architektúrán végzett tanítás alapján

Ahogy az eredmények mutatják, a validációs adathalmaz 900 darab zenéjéből első helyen 412 darabot talál el helyesen, második helyen 200-at, míg harmadikként 123-at, azaz összesen 735 darab zenét, amely arányait tekintve közel 82%-os eredménynek felel meg top3 pontosság szerint. Ez azt jelenti, hogy a „várt” stílus 82%-os valószínűséggel fog megjelenni, és mindössze 18% annak a valószínűsége, hogy a fennmaradó 6 kategória egy tetszőleges kombinációja kerül kiírásra.

A VGG16 és a MobileNetV2 próbálkozásai mellett felmerült ötletként, hogy nyerhető-e magasabb pontosság abból, ha a képeket nem 224x224-es méretűre kerülnek lecsökkentésre, hanem a képek felbontását figyelembe véve, aránytartó módon, a lehető legnagyobb képeket átadni tanításra és validációra. Ennek eléréséhez készült egy modell, amelyben ez megvalósításra kerül.

Nem elhanyagolandó az, hogy ezek a képek meglehetősen nagyok, ez magas memóriaigénnyel jár. Ez látszódt is, hiszen eredeti képméretként nem is lehetett átadni a modell számára ezt. Az egy iteráció alatt áthaladó képmennyiség, *batchsize* minél alacsonyabbra vétele mellett, annak végül 4-es értéket meghatározva sikerült minden oldalt negyedére zsugorítva megindítani a tanítást. Így összességében az eredeti képméretet 1/16-od akkorára csökkentve.

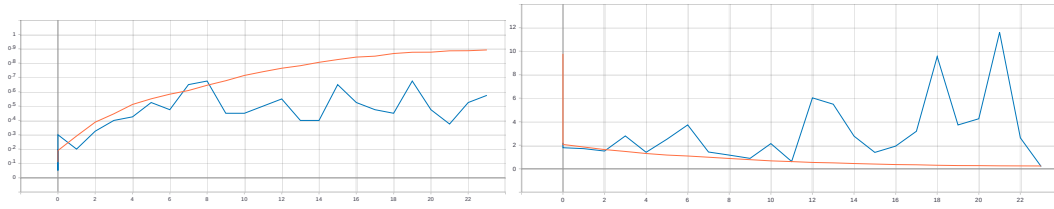
Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 714, 954, 128)	18944
max_pooling2d_1 (MaxPooling2D)	(None, 357, 477, 128)	0
conv2d_2 (Conv2D)	(None, 357, 477, 128)	802944
leaky_re_lu_1 (LeakyReLU)	(None, 357, 477, 128)	0
max_pooling2d_2 (MaxPooling2D)	(None, 178, 238, 128)	0
conv2d_3 (Conv2D)	(None, 178, 238, 128)	409728
leaky_re_lu_2 (LeakyReLU)	(None, 178, 238, 128)	0
max_pooling2d_3 (MaxPooling2D)	(None, 89, 119, 128)	0
conv2d_4 (Conv2D)	(None, 89, 119, 128)	147584
leaky_re_lu_3 (LeakyReLU)	(None, 89, 119, 128)	0
max_pooling2d_4 (MaxPooling2D)	(None, 44, 59, 128)	0
dropout_1 (Dropout)	(None, 44, 59, 128)	0
conv2d_5 (Conv2D)	(None, 44, 59, 256)	1605888

leaky_re_lu_4 (LeakyReLU)	(None, 44, 59, 256)	0
max_pooling2d_5 (MaxPooling2D)	(None, 22, 29, 256)	0
conv2d_6 (Conv2D)	(None, 22, 29, 256)	1638656
leaky_re_lu_5 (LeakyReLU)	(None, 22, 29, 256)	0
max_pooling2d_6 (MaxPooling2D)	(None, 11, 14, 256)	0
conv2d_7 (Conv2D)	(None, 11, 14, 256)	590080
leaky_re_lu_6 (LeakyReLU)	(None, 11, 14, 256)	0
max_pooling2d_7 (MaxPooling2D)	(None, 5, 7, 256)	0
dropout_2 (Dropout)	(None, 5, 7, 256)	0
conv2d_8 (Conv2D)	(None, 5, 7, 512)	6423040
leaky_re_lu_7 (LeakyReLU)	(None, 5, 7, 512)	0
max_pooling2d_8 (MaxPooling2D)	(None, 2, 3, 512)	0
dropout_3 (Dropout)	(None, 2, 3, 512)	0
conv2d_9 (Conv2D)	(None, 2, 3, 512)	2359808
leaky_re_lu_8 (LeakyReLU)	(None, 2, 3, 512)	0
max_pooling2d_9 (MaxPooling2D)	(None, 1, 1, 512)	0
flatten_1 (Flatten)	(None, 512)	0
dense_1 (Dense)	(None, 512)	262656
dense_2 (Dense)	(None, 256)	131328
dense_3 (Dense)	(None, 128)	32896
dense_4 (Dense)	(None, 9)	1161
=====		
Total params: 14,424,713		
Trainable params: 14,424,713		
Non-trainable params: 0		

57. ábra A lehető legnagyobb képméretet feldolgozó modell szerkezete

Ezen szerkezet kialakítása előtt törekedtem az eredeti VGG16 modellhez hasonlóan *ReLU* aktivációt alkalmazni az egyes konvolúciós rétegek esetén, ugyanakkor ahogy az annál is előfordult, a *dying ReLU* jelensége ennél a megoldásnál is megjelent, így minden aktiváció kiváltásra került *LeakyReLU* aktivációval az első réteget leszámítva. A konvolúciós-maxpool rétegek közé, amikor filterszám növekedés történt, egy-egy dropout réteg került alkalmazásra. A konvolúciós műveletek végén, a *Flatten* réteget követően piramis alakban egyre csökkenő neuronszámú, teljesen kötött hálózat lett kialakítva.

A tanítás elindulásakor célként a minél alacsonyabb validációs veszteség lett meghatározva minél amellet, hogy a validációs pontosság minél magasabb legyen.



58. ábra 9 kategóriára vonatkozó tanítási eredmények saját készítésű modell által, pontosság- (balra) és veszteségértékek szerint, narancsszínnel a tanítóhalmazra, kék színnel a validációs halmazra vonatkozóan

Mivel validációs veszteség szerinti, tanítás közbeni mentés történt, így végül egy olyan modell került mentésre, amely nem a legnagyobb validációs pontosságú értékkel rendelkezik. Ez ugyanis a legvégén következett be, ekkor 57,5%-os eredményt elérve. Ez magasabb, mint a VGG16 modellnél elért értéknél. Erre a modellre is megtörtént a validációs halmaz illesztése.

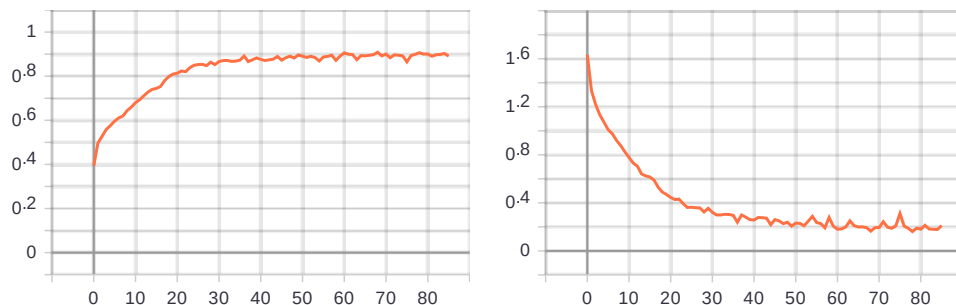
448 -- 50%
199 -- 22%
104 -- 12%

59. ábra Első stílusként, második stílusként, illetve harmadik stílusként megtalált zenestílusok darabszáma, valamint százalékos aránya saját modell szerint

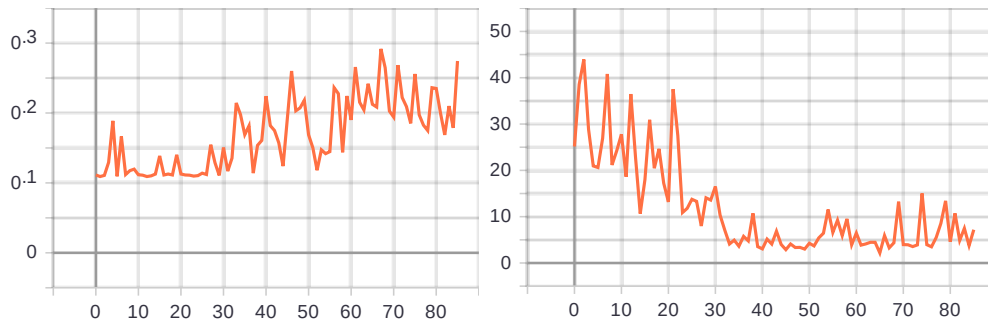
Összehasonlítva az előző eredményekkel, ennél a megoldásnál 4%-os javulás lett megfigyelhető, az első szintű pontosság majdnem elérte a kerek 50%-ot, második szinten 22%-ot, harmadik szinten 12%-ot eredményezve. Ez összesen 84%-os, első 3-szintű pontosságot mutat a validációs halmazra vonatkozóan.

Futtatva egy újabb tanítást ugyanezen a modellen sikerült elérni a tanítás végére egy 59,5%-os validációs pontosságot, amely bizonyította, hogy körülbelül ez az az érték, amelyet ez a modell produkálni képes. Tesztelés során ez is bekerült a tesztelendő modellek körébe.

Ezután a próbálkozás után, illetve ezzel egy időben került felvetésre a NASNet névre hallgató modell, illetve azok variánsai. Ezek közül a NASNetMobile változatával próbálkoztam, 32-es értékű *batchsize* tulajdonsággal, a tanítást pedig alapeseten 500 *epoch*-ra állítva.



60. ábra NASNetMobile modellel történt tanítás eredménye a tanítási halmazra vonatkozóan



61. ábra NASNetMobile modellel történt tanítás eredménye a validációs halmazra vonatkozóan, eredményezve egy legjobb 30% körüli értéket validációs pontosság szerint

Eredményként megállapítható, hogy a módszer körülbelül háromszoros javulást tudott mutatni az alap random klasszifikátor értékhez képest, ugyanakkor ennél már mutatkozott jobb eredmény, emiatt ez a fajta megközelítés elvetésre került.

Párhuzamosan e mellett megkezdődött a spotifyos adatokra vonatkozó tanítás is. Ez az után történt meg, miután véglegessé vált, hogy 9 stílusra fog történni tanítás, illetve hogy melyek ezek a stílusok pontosan. Előtte megtörtént a korrelációanalízis a megmaradt 9 stílusra vonatkozóan, illetve az adatok korrigálása, ezen eredmények olvashatók a 11. és 12. táblázat soraiban.

	acousticness	danceability	energy	instrumentalness	key	liveness	loudness	speechiness	tempo	valence
acousticness	1,000									
danceability	-0,536	1,000								
energy	-0,819	0,479	1,000							
instrumentalness	0,319	-0,331	-0,325	1,000						
key	-0,059	0,041	0,065	-0,024	1,000					
liveness	-0,156	0,001	0,211	-0,108	0,025	1,000				
loudness	-0,719	0,564	0,822	-0,554	0,052	0,172	1,000			
speechiness	-0,163	0,281	0,136	-0,262	0,013	0,059	0,199	1,000		
tempo	0,017	-0,003	-0,013	-0,023	0,009	0,003	0,003	0,019	1,000	
valence	-0,295	0,485	0,382	-0,460	0,045	0,045	0,422	0,178	0,012	1,000

11. táblázat Korrelációanalízis a 9 zenestílusra vonatkozóan

	danceability	energy	key	liveness	speechiness	tempo	valence
danceability	1,000						
energy	0,479	1,000					
key	0,041	0,065	1,000				
liveness	0,001	0,211	0,025	1,000			
speechiness	0,281	0,136	0,013	0,059	1,000		
tempo	-0,003	-0,013	0,009	0,003	0,019	1,000	
valence	0,485	0,382	0,045	0,045	0,178	0,012	1,000

12. táblázat Javított korrelációanalízis a 9 zenestílusra vonatkozóan, kivéve az acousticness, loudness és instrumentalness értékeket

A tanításhoz, mivel itt nem képek feldolgozása, hanem számértékek feldolgozása volt a cél, egy egyszerűbb, *fully-connected* neurális hálózat lett felállítva, amelyben az egyes rétegek *Dense* típusúak, bennük az egyes rétegekben szereplő neuronok mindegyike össze van kapcsolva. Annak érdekében, hogy az *overfitting* minél inkább el legyen kerülve, itt is használatra került *Dropout* a regularizáció eléréséért.

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 10)	110
dense_2 (Dense)	(None, 200)	2200

dense_3 (Dense)	(None, 200)	40200
dense_4 (Dense)	(None, 200)	40200
dropout_1 (Dropout)	(None, 200)	0
dense_5 (Dense)	(None, 100)	20100
dense_6 (Dense)	(None, 100)	10100
dense_7 (Dense)	(None, 100)	10100
dropout_2 (Dropout)	(None, 100)	0
dense_8 (Dense)	(None, 50)	5050
dense_9 (Dense)	(None, 50)	2550
dense_10 (Dense)	(None, 50)	2550
dropout_3 (Dropout)	(None, 50)	0
dense_11 (Dense)	(None, 9)	459
=====		
Total params: 133,925		
Trainable params: 133,925		
Non-trainable params: 0		

62. ábra A spotifyos adatokra vonatkozó modell szerkezete, szerepeltetve a tanító paraméterek számát

Az első tanítás *Adamax* optimalizációval, valamint 100-as *batchsize* paraméterrel történt meg, adva eredményül 54%-os első szintű pontosságot, amellyel közel ötszörös javulást lehet megfigyelni a random klasszifikátor 11%-os értékéhez képest. Mutakozott továbbá 20%-os második szintű, illetve 11%-os harmadik szintű pontossági érték, amely összességében kerekítve 86%-ra jött ki. Felmerült ugyanakkor, hogy jó döntés volt-e a korrelációs adatokat kivenni az adathalmazból. Minden adat ugyanis adat, főleg, hogy relatív nem kevésről van szó. Következő körökben a korreláló adatok megtartásával történt tanítás, ahol már az *Adamax* mellett az *RMSProp* nevű optimalizáló is használatra került, mint utólag kiderült, nem is hiába.

Modell elnevezése	top1	top2	top3	Összesen	Összesen (%)
Adamax_batch100_9cat_NoCorrel	54% (490)	74% (674)	86% (772)	772	86%
RMSProp_batch100_9cat_Correl_checkpoint	57% (517)	78% (710)	88% (788)	788	88%
RMSProp_batch100_9cat_Correl_endTrain	56% (507)	77% (698)	88% (793)	793	88%
Adamax_batch100_9cat_Correl_endTrain	55% (492)	77% (691)	86% (768)	768	86%
Adamax_batch100_9cat_Correl_checkpoint	54% (486)	77% (692)	87% (779)	779	87%
RMSProp_batch100_9cat_Correl_lowLR_endTrain	51% (458)	71% (641)	83% (749)	749	83%
RMSProp_batch100_9cat_Correl_lowLR_checkpoint	51% (458)	62% (650)	83% (751)	751	83%

13. táblázat Eredményül kapott értékek első szintű, második szintű, harmadik szintű pontosságokra adott modellek esetén a validációs halmazra vonatkozóan

Minden esetben a 100-as *batchsize* lett megtartva, ugyanakkor az látszik a 13. táblázat szerint, hogy a korreláló adatok megtartása pozitív irányba billentette a pontosságot többnyire. *RMSProp* optimalizálóval történő tanítás során sikerült elérni 88%-ot top3 pontosság szerint, amelynél még az első szintű pontosság is 3%-ot javult. Ez ugyanakkor nem minden esetben volt így. Amennyiben az optimalizáló tanulási rátája csökkentésre

került, úgy a pontosság rovására ment, gyengébb eredményeket produkált, mint a korrelálatlan adatok esetén, a 13. táblázat alsó két bejegyzése ezt mutatja.

Az elkészített modellek által adott eredmények grafikus megjelenítéséért egy C# nyelven íródott program készült el. A program egyszerű felépítéssel rendelkezik, MVVM struktúrában megalkotva. A Model részt egyetlen osztály, a *Music* alkot, amely egy adott zenével kapcsolatban tárolja el a különböző, stílusok meghatározásához szükséges információkat. A ViewModel részen található meg az összeköttetés a vizuális megjelenítésért felelős View és az adatok tényleges tárolásában részt vevő Model között, ahol minden, információátadáshoz, adatösszegyűjtéshez és prediktálás elvégzéséhez metódus is megtalálható.

A prediktáláshoz magához szükséges Python interpreter megadása. Az elkészített program mappájában található egy extra konfigurációs fájl, aminek segítségével meg lehet adni, hogy az alkalmazást használó számítógépén hol található a szükséges fájl. A szükséges fájl megadása a „C:\<elérési útvonala>\<ennek>\<a>\<fájlnak>\python.exe” módon történik, amely alapeseten „C:\Users\András\Virtual ENVs\Python3.6\Scripts\python.exe” útvonalra van beállítva. Ennek beállításához szükséges megnyitni a *MusicRecognizer_Ver1.exe.config* fájlt egy szövegszerkesztő segítségével, amely konfigurációs fájlban belül a *configuration* mezején belüli *appSettings* mezején belül hozzá van adva egy kulcsérték alapján ezen elérési útvonal, mint érték.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <appSettings>
    <add key="PathToPython" value="C:\Users\András\Virtual ENVs\Python3.6\Scripts\python.exe" />
  </appSettings>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
  </startup>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="Newtonsoft.Json" publicKeyToken="30ad4fe6b2a6aeed" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-12.0.0.0" newVersion="12.0.0.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

63. ábra A konfigurációs fájl szerkezete, kiemelve a megfelelő értéket, amely a helyes működéshez szükséges módosítani

A helyes működéshez szükséges az interpreter számára az alábbi csomagokat feltelepíteni a predikciók elvégzéséhez. Fontos, hogy a helyes működéshez megadott verziószám is szükségeltetik, valamint az ezen csomagok telepítéséhez szükséges, itt fel nem tüntetett segédcsomagok is. Fontos továbbá, hogy fejlesztés során 3.6-os verziószámú Python került felhasználásra a tanítás számára.

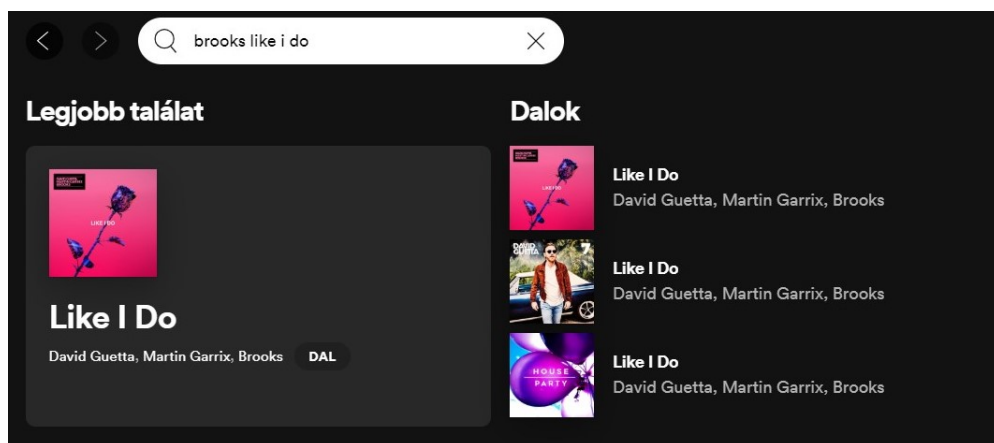
- Keras (2.2.4)
- Keras-Applications (1.0.8)
- Keras-Preprocessing (1.1.0)
- matplotlib (3.1.1)
- numpy (1.17.4)
- tensorflow (1.15.0)
- tensorflow-estimator (1.15.1)

- tensorflow-gpu (2.1.0)
- tensorflow-gpu-estimator (2.1.0)
- scipy (1.4.1)

Az alkalmazás elkészülése során figyelembe kellett venni, hogy milyen lesz majd a felhasználó által beadott zene, illetve rendelkezik-e internetkapcsolattal. Amennyiben nem, úgy a Spotify alapú online zenestílus meghatározás nem működőképes, ugyanakkor továbbra is működőképes marad az offline, spektrogram alapú. Előbbi kiegészült annyival, hogy amennyiben ismert számára, miszerint adott zene megtalálható Spotify-on, úgy mintegy keresést ejtve a zenére, kiolvashatóak annak adatai. Fontos információ, hogy amennyiben ez történik, úgy kooperáció esetén csak a megtalált zene első előadója kerül kiíratásra, illetve az első dal egyezésre kerül megállapításra.



64. ábra Felhasználó által megadott zenére történő keresésre, és annak eredménye



65. ábra Spotify alkalmazáson belüli keresés eredménye ugyanazon zenére. Látható, hogy az első előadó neve egyezik az alkalmazáson belül megadott névvel is.

Számos olyan eset lekezelésre került az alkalmazás készítése közben, amely előjöhet az egyes fájlok tallózása, illetve zenék keresése közben. Legfontosabb talán a zenefájl kiterjesztésének vizsgálata, hogy ténylegesen érvényes kerüljön betallózásra. Betallózni mp3, wav, m4a illetve flac formátumú zenéket lehet, amelyeknél a spektrogram alapú predikció mp3, wav, illetve flac formátum esetén támogatott, míg az online alapú stílusmeghatározáshoz betallózni mp3, m4a vagy flac formátumú fájlt szükségeltetik. Offline változat esetén fontos, hogy a betallózott zenének az érvényes formátum mellett

megfelelő hosszúnak is kell lennie. A tanítóminták esetén ez a hossz 45 másodperc és 90 másodperc közé volt tehető, azaz egy 45 másodperces részlet kell innen, így a bevitt zenének legalább 90 másodpercesnek kell lennie. Ha csak ilyen hosszú, és nem részletről van szó, akkor ez torzíthat az eredményen, mivel a tanítás során hosszabb zenék lettek legnagyobb részt felhasználva, azokból készültek a minták.

Online változat esetén több dolgot is figyelembe kell még venni. Egyrészt azt, hogy található-e a bevitt zenei fájlban érvényes metaadat, azaz eltárolásra került-e korábban az előadó(k) és a zeneszám címe legalább ID3 tag-ek formájában. A validációja ezen adatoknak nehezen állapítható meg, hiszen kézzel bárki által módosíthatóak ezen adatok, feltételeztem, hogy nem történt módosítás ilyen téren, és a beírt adat érvényes, adott zenéhez tartozik. Ha kézzel történt beírásra a keresési mezőbe a keresni kívánt zene, illetve sikeresen betallózásra került a feldolgozandó fájl, úgy megtörténik maga a keresés a Spotify adatbázisában. Itt ellenőrizendő, hogy a valamely formában bevitt zene megtalálható-e az adatbázisban, mert csak akkor kerülhetnek kiírásra egyes zenei tulajdonságok, ha jelen vannak ezen adatok, és ezen adatokhoz kapcsolódó zene. Ha megvannak az adatok, utána történhet meg a stílus meghatározása.

Technikai jellemzőket illetően, annak érdekében, hogy működhessen a C# alapú programból történő Python szkript lefuttatása, majd ezen futás eredményeképp eredményül kapott stílusok kiírására kerüljenek, segédfájlok kerülnek felhasználásra. Spektrogramoknál ez egy 45 másodperces wav formátumú zenefájl formájában, egy ezen zenéről készített tényleges spektrogram szerint, valamint egy, az eredményeket tartalmazó *predictions* szöveges fájl segítségével kerülnek megállapításra a stílusok. Ez a spotifyos megoldás esetén kicsit másképp alakul. Annyi megmarad, hogy az eredmények ugyanúgy egy *predictions* fájlba kerülnek kimentésre, majd onnan beolvasásra, csak itt még a modell és a predikció elvégzése számára elmentésre kerülnek egy *properties* szöveges dokumentumban a kiolvasott tulajdonságok értékei, ide nem szükséges külön zenefájl, illetve elkészített spektrogram sem. Mindkét megoldás esetén nemcsak a stílusok, hanem az egyes stílusokhoz tartozó valószínűségek is elmentésre, illetve kijelzésre kerülnek a folyamat végére.

6. Tesztelési dokumentáció

Tesztelési szempontból mindkét fajta modell esetén a legjobb 3 lett kiválasztva a validációs halmazon történő pontosságértékek szerint, ahol a figyelembe vett szempontok a top1 valamint a top3 pontosságok voltak. A kigyűjtést követően megtörtént a teszhalmaz összegyűjtése, melynek során 1036 darab spektrogramot sikerült kialakítani, míg Spotify felől további 965 darab zenéhez tartozó adat került összegyűjtésre.

Zene stílusa	Összegyűjtött spektrogram	Összegyűjtött Spotify adat
Classical	105	110
Country	83	95
Electronic	124	120
Hiphop	131	100
Jazz	116	100
Latin	114	100
Pop	127	130
Rap	117	100
Rock	119	110
összesen:	1036	965

14. táblázat Összegyűjtött teszhalmaz nagysága darabban kifejezve spektrogramokra és az online predikcióra vonatkozóan

A legjobb 3 spektrogram alapú modell az alábbi jellemzőkkel rendelkezik:

- VGG16 architektúrára épülő modell Adamax optimalizálással, ahol a modell validációs pontossága a tanítás közben 54,375%
- A lehető legnagyobb képméretet megtartó modell Adamax optimalizálással, ahol a tanítás végén elért validációs pontosság 57,5%
- A lehető legnagyobb képméretet megtartó modell Adamax optimalizálással, ahol a tanítás közben elért validációs pontosság 59,5%

A legjobb 3 Spotify adatokra alapú modellek pedig az alábbiak lettek:

- Adamax optimalizálással rendelkező modell a tanítás végén, ahol a validációs adatokra vonatkozó top1 pontosság 54%, a top3 pontosság 87%
- RMSProp optimalizálással rendelkező modell a tanítás közben, ahol a validációs adatokra vonatkozó top1 pontosság 57%, a top3 pontosság 88%
- RMSProp optimalizálással rendelkező modell a tanítás végén, ahol a validációs adatokra vonatkozó top1 pontosság 56%, a top3 pontosság 88%

```

1059 168. zene stílusai: (Country)
1060 1. legvalószínűbb műfaj: Country, ekkora esellyel: 55%
1061 2. legvalószínűbb műfaj: Jazz, ekkora esellyel: 21%
1062 3. legvalószínűbb műfaj: Rock, ekkora esellyel: 8%
1063
1064
1065 169. zene stílusai: (Country)
1066 1. legvalószínűbb műfaj: Country, ekkora esellyel: 48%
1067 2. legvalószínűbb műfaj: Latin, ekkora esellyel: 17%
1068 3. legvalószínűbb műfaj: Jazz, ekkora esellyel: 14%
1069
1070
1071 170. zene stílusai: (Country)
1072 1. legvalószínűbb műfaj: Country, ekkora esellyel: 48%
1073 2. legvalószínűbb műfaj: Latin, ekkora esellyel: 19%
1074 3. legvalószínűbb műfaj: Jazz, ekkora esellyel: 18%
1075
1076
1077 171. zene stílusai: (Country)
1078 1. legvalószínűbb műfaj: Country, ekkora esellyel: 26%
1079 2. legvalószínűbb műfaj: Pop, ekkora esellyel: 22%
1080 3. legvalószínűbb műfaj: Rock, ekkora esellyel: 17%
1081

```

66. ábra Részlet egy tesztelési halmazra történő predikciókról készített logfájlból

A predikciók eredményei logfájlokba történő írását követően megtörtént azok feldolgozása. Feldolgozáshoz egy külön C# alapú alkalmazás készült, amely lehetővé tette, hogy a top1-top2-top3 pontosságok mellett a top1 értékekre vonatkozó konfúziós mátrix is elkészülhessen. Konfúziós mátrix segítségével könnyedén kiolvasható, hogy hol hibázott a modell top1 szinten, hol történt félreklasszifikáció, illetve példát mutat arra, hogy melyek azok a stílusok, amelyek nem a legegységelműbben azonosíthatóak, és melyekkel „keveri össze” a modell.

A spektrogramokra vonatkozó teszteredményeket a 15. táblázat mutatja.

	top1	top2	top3	Nem azonosított	Nem azonosított (%)
Classical	60	77	81	24	22,86%
Country	42	58	62	21	25,30%
Electronic	73	92	102	22	17,74%
Hiphop	30	80	99	32	24,43%
Jazz	57	93	107	9	7,76%
Latin	34	78	90	24	21,05%
Pop	68	102	116	11	8,66%
Rap	48	68	87	30	25,64%
Rock	45	84	101	18	15,13%
Összesen:	457	732	845	191	
Összesen(%):	44,07%	70,59%	81,49%		

15. táblázat VGG16 architektúrára épülő modell teszteredményei top1, top2, illetve top3 szinten eltalált stílusok szerint

Az eredményekről ezen modell esetén általánosan megállapítható, hogy a zenék stílusa meglehetősen egyenletesen oszlik el hibaszázalékokat tekintve. A legmagasabb ilyen érték 25% körüli, amely a *Rap* és a *Country* stílusoknál fordul elő, ezeken kívül minden stílus ennél alacsonyabb aránnyal rendelkezik.

	top1	top2	top3	Nem azonosított	Nem azonosított (%)
Classical	67	70	73	32	30,48%
Country	32	40	44	39	46,99%
Electronic	93	100	101	23	18,55%
Hiphop	26	98	105	26	19,85%
Jazz	59	81	87	29	25,00%
Latin	14	36	46	68	59,65%
Pop	51	87	104	23	18,11%
Rap	68	85	95	22	18,80%
Rock	76	96	104	15	12,61%
összesen:	486	693	759	277	
összesen (%):	46,87%	66,83%	73,26%		

16. táblázat A lehető legnagyobb képméretet megtartó modell teszteredményei (validációs pontosság: 59,5%)

	top1	top2	top3	Nem azonosított	Nem azonosított (%)
Classical	71	75	79	26	24,76%
Country	27	33	44	39	46,99%
Electronic	105	111	114	10	8,06%
Hiphop	66	106	109	22	16,79%
Jazz	45	64	70	46	39,66%
Latin	20	43	59	55	48,25%
Pop	58	85	96	31	24,41%
Rap	48	85	96	21	17,95%
Rock	47	71	92	27	22,69%
összesen:	487	673	759	277	
összesen (%):	47,01%	64,96%	73,26%		

17. táblázat A lehető legnagyobb képméretet megtartó modell teszteredményei (validációs pontosság: 57,5%)

Ezen két modell esetén általánosságban az mutatkozik, hogy a *Latin* és a *Country* zenék esetén meglehetősen magas arányban nem kerülnek bele az első 3 stílus közé, itt az arányok 50% körüliek, de közel 60%-os érték is előfordul. Amennyiben ez a két kategória nincs figyelembe véve az eredmények kiértékelésében, úgy itt is meglehetősen egyenletesen oszlanak el arányait tekintve a helytelenül nem azonosított stílusok zenéi. Ahogy az a validációs adatokból is mutatkozott, legalább 4-szeres javulás figyelhető meg a tesztadatokon is a véletlen klasszifikátorhoz képest a top1 pontosságot illetően. Ez a pontosság azon modelleknél még ennél is magasabb, amely igyekezett a lehető legnagyobb képeket feldolgozni tanítás során, itt közelítettek az értékek az 50%-hoz is. Itt mutatkozott egy olyan jellegzetesség, hogy a modell gyakran mutatkozott nagyon biztosnak a dolgában, ami azt jelentette, hogy gyakran az első helyre várt stílust 100%-

osan várta, ami abban az esetben nem probléma, ha azt jól várja az első helyen, ugyanakkor, ha nem jól várja oda, hanem esetleg a második vagy harmadik helyen, úgy 0%-os pontossággal nem volt figyelembe vehető ezen érték. Így fordulhatott elő az, hogy a top3 pontosság szerint a VGG16-ra épülő modell mutatkozott a legjobbnak, pontosságát illetően elérve közel 81,5%-ot, nagyjából 7-8%-kal jobb top3 pontossággal kapcsolatos eredményt mutatva. Amennyiben véletlenszerűen lennének kiválasztva ezen stílusok, úgy 33,33% pontossággal lehetne szerepeltetni a hármasban azt a stílust, amely várva van eredményként. Ez a 80%-ot meghaladó érték ennek közel 2,5-szerese, annálval jobbnak bizonyul ennél.

A konfúziós mátrixok ettől függetlenül mindhárom modell típusra elkészült, ezek közül a legjobbnak vélt VGG16-os modell mátrixa a 18. táblázatból olvasható eredményeket mutatta.

	Classical	Country	Electronic	Hiphop	Jazz	Latin	Pop	Rap	Rock
Classical	60	0	7	1	29	0	4	0	4
Country	6	41	0	3	9	2	13	0	9
Electronic	2	0	73	2	32	1	13	1	0
Hiphop	0	4	3	31	7	3	31	52	0
Jazz	18	2	0	5	57	0	31	0	3
Latin	1	3	1	1	12	34	43	15	4
Pop	0	14	1	13	12	2	68	9	8
Rap	0	0	19	22	1	4	20	48	3
Rock	0	19	0	0	9	0	46	0	45

18. táblázat VGG16 alapú modell top1 eredményeinek konfúziós mátrixa

A konfúziós mátrix vízszintes sorait összeadva meg lehet kapni az adott stílusra prediktált zenék számát, és az egyes helyeken mutatva, hogy hány olyan stílusú jött ki végül, másképp megfogalmazva, a függőleges tengelyen olvashatóak az elvárt stílusok, míg azoknak vízszintesen megfeleltetve a prediktált stílust. Például a 105 db *Classical* zenestílust képviselő zenék esetén 60 db *Classical*, 7 db *Electronic*, 1 db *Hiphop*, 29 db *Jazz*, valamint 4-4 db *Pop* és *Rock* stílus jött ki első helyen. Főatlóban szerepelnek a helyesen prediktált zenék mennyisége. További jelentős következtetéseket lehet ezen eredményekből levonni. Ahogy az a stílusok számának csökkenésével említésre került, úgy ténylegesen kiderült, hogy a *Hiphop* és a *Rap* műfajok nagyban hasonlítanak egymásra spektrogramok szerint. *Hiphop* zenék esetén többször is *Rap* stílus jött ki eredményül első helyen, míg *Rap* stílus esetén a *Hiphop* fordult elő többször is. Ebből is látszik hasonlóságuk, ugyanakkor egy adott arányban mégis csak megfelelő stílusdetektálás történt. Az eredményekből megállapítható még, hogy a *Pop*, mint stílus, igen általánosnak számít stílusok tekintetében. Ez sem számít feltétlen nagy hibának. A

pop, mint stílus, populáris, népszerű zenét jelent. A stílusok határai pedig nem annyira élesek és tudományos alapon elkülönítettek, hogy teljesen el lehessen különíteni például a latin zenétől. A *Latin* stílust említve, megállapításra került, hogy önmagában attól, hogy leginkább spanyolul vagy valamely Latin-Amerikában népszerű nyelven éneklük a dalszöveget, amennyiben található egy zenében, vagy az arra a területre jellegzetes hangszer szólal meg, úgy ez nem számít döntő tulajdonságnak és kiolvashatónak spektrogramokból. Sokkal inkább számít az énekhangok lejtése és az előadásmód. Több stíluspárost is lehet említeni, amelyek párban kerülnek helyesen, illetve helytelenül prediktálásra. Ilyen a már szóba hozott *Hiphop-Rap* páros, de a *Country-Rock*, a *Country-Pop* és a *Classical-Jazz* stíluspárosok is ehhez a nézethez társíthatók.

Az online, Spotify adatai alapján készített, legjobbnak vélt modellek az alábbi teszteredményekkel jártak, hasonlóképp kiértékelve, mint a spektrogramok modelljei esetén.

	top1	top2	top3	Nem azonosított	Nem azonosított (%)
Classical	97	106	106	4	3,64%
Country	67	85	90	5	5,26%
Electronic	91	99	104	16	13,33%
Hiphop	34	71	83	17	17,00%
Jazz	62	92	95	5	5,00%
Latin	50	56	69	31	31,00%
Pop	27	57	89	41	31,54%
Rap	37	69	83	17	17,00%
Rock	25	57	67	43	39,09%
összesen:	490	692	786	179	
összesen (%):	50,78%	71,71%	81,45%		

19. táblázat Adamax optimalizálással rendelkező modell teszteredményei top1, top2, illetve top3 szinten eltalált stílusok szerint

Az Adamax optimalizálással tanított modell, amelynek eredményei a 19. táblázatban találhatóak, leginkább a rockzenét nem találja meg az első 3 stílus között, itt 10-ből közel 4 zenét kategorizál félre úgy. Ezekon kívül kilengést mutatnak a latin- illetve a popzenék, itt közel a zenék harmada nem volt szerepeltetve az első 3 helyen. Ezeket leszámítva az eredmények megfelelőnek mutatkoznak, 80% feletti top3 pontossággal, amelyből a zenék közel 51%-át az első helyen várja, a top2 pontosság összesítve közel 72%-on áll. Kiemelendő lehet még, hogy a *Classical*, *Country* és *Jazz* stílusú zenék esetén csak néhány esetben nem került azonosításra a megfelelő stílus.

	top1	top2	top3	Nem azonosított	Nem azonosított (%)
Classical	96	105	107	3	2,73%
Country	63	76	82	13	13,68%
Electronic	85	96	102	18	15,00%
Hiphop	46	72	82	18	18,00%
Jazz	72	90	94	6	6,00%
Latin	58	76	84	16	16,00%
Pop	26	53	71	59	45,38%
Rap	40	62	82	18	18,00%
Rock	34	42	51	59	53,64%
összesen:	520	672	755	210	
összesen (%):	53,89%	69,64%	78,24%		

20. táblázat RMSProp optimalizálással rendelkező modell teszteredményei (tanítás végi állapot)

Egyfajta anomáliát mutatnak azon teszteredmények, amelyek azon modell esetén jöttek ki, amelynél RMSProp optimalizáló került felhasználásra (20. táblázat), és a tanítás végi mentés eredményezte a modellt. Validációs halmazra vonatkozóan top1 pontosság szerint az Adamax optimalizációs modell 3%-kal jobbnak bizonyult ennél a megoldásnál. Tesztknél viszont az adamaxoshoz képest több, mint 3%-os top1 pontosság javulás figyelhető meg, viszont összességében, ha top3 pontosság kerül figyelembe vételre, úgy ez a 3% hátrányban mutatkozik meg, mindössze 78%-ot mutat. Ez azt jelenti, hogy top2 és top3 pontosságok szerint nagyjából 24% jött még össze, míg ez adamaxos változatánál 30%-os értékű. Ez a fajta modell is nehezebben ismeri fel a rockzenét, de ez a fel nem ismerési érték is jóval magasabb, több, mint 50%-os. 50% közelébe mozgott a popzene is a maga 45%-os hibázási eredményével. A többi 7 stílusra vonatkozóan viszont minden esetben 20% alatti azoknak az aránya, amelynél nem kerül az első 3 hely valamelyikére a várt stílus.

	top1	top2	top3	Nem azonosított	Nem azonosított (%)
Classical	99	108	109	1	0,91%
Country	58	78	88	7	7,37%
Electronic	93	98	103	17	14,17%
Hiphop	36	69	86	14	14,00%
Jazz	69	95	96	4	4,00%
Latin	47	54	72	28	28,00%
Pop	30	57	85	45	34,62%
Rap	27	60	76	24	24,00%
Rock	40	60	74	36	32,73%
összesen:	499	679	789	176	
összesen (%):	51,71%	70,36%	81,76%		

21. táblázat RMSProp optimalizálással rendelkező modell teszteredményei (tanítás közbeni, checkpoint állapot)

Validációs veszteség szerinti legalacsonyabb értékkel rendelkező mentés során valamivel jobb eredmények mutatkoznak, mint amit ugyanaz az optimalizáló produkált a tanítás végére, ahogy azt a 21. táblázat mutatja. Az adamaxos variánshoz képest mindössze 0.3%-kal tudott jobb top3 pontosságot felmutatni a tesztadatokra vonatkozóan. Ez elosztás szerint 1%-kal jobb top1 pontosságot, közel ugyanakkora top2 pontosságot mutat. Stílusok szerint továbbra is a *Rock* az egyik olyan stílus, amely nehezebben kerül felismerésre, viszont itt már a rockzenék kétharmada felismerésre kerül az első 3 stílusok között. Ugyanez a jellemző a popzenék esetén is. Említésre méltó, hogy *Classical* stílus esetén csak 1 olyan zene volt, amelyet nem várt az első 3 közé, ehhez hozzátevé azt, hogy harmadik szinten csak 1 zenét talált, így 2 kivétellel a legjobb 2 között szerepel megfelelően a stílus.

A magasabb top1, valamint top3 pontosság miatt a legutóbbi, RMSProp optimalizációval rendelkező, tanítás közbeni modell került további vizsgálatra konfúziós mátrix segítségével, ennek eredménye a 22. táblázatban található.

	Classical	Country	Electronic	Hiphop	Jazz	Latin	Pop	Rap	Rock
Classical	99	0	0	0	11	0	0	0	0
Country	0	58	2	0	9	5	8	1	12
Electronic	0	0	93	3	0	6	7	4	7
Hiphop	0	4	0	36	5	15	9	29	2
Jazz	21	6	0	0	69	0	2	1	1
Latin	0	18	3	3	9	47	14	1	5
Pop	2	22	15	6	7	12	30	1	35
Rap	1	8	4	30	3	16	10	27	1
Rock	9	25	8	2	6	9	9	2	40

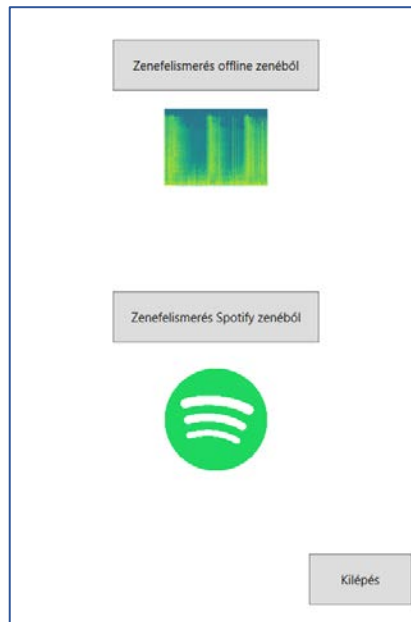
22. táblázat RMSProp alapú modell teszteredményeire vonatkozó top1 eredmények konfúziós mátrixa

Ismételten a főátlóval kezdve az elemzést, legtöbb stílusnál megfigyelhető, hogy a stílusokat megfelelően találta meg első szinten. Két kivételt lehet csak említeni. Az egyik várható volt az előző konfúziós mátrixból is, hiszen ott is jellemző volt, hogy a *Rap* és a *Hiphop* stílusú zenék oda-vissza kerülnek helyesen-helytelenül prediktálásra. *Hiphop* többet talált el helyesen, mint amennyit többé-kevésbé helytelenül, *Rap* stílusként. *Rap* esetén viszont ez az arány kissé fordított, 3 darabbal kevesebbet sikerült helyesen azonosítani, mint amennyit tette *Hiphop* stílusra. Ebből újból lehet következtetni ezen stílusok rendkívül hasonló felépítésére, módjára, előadásmódjára, amely már nemcsak kép alapján, spektrogramokból kerül megállapításra, hanem adatbázisban található kiolvasott adatokra történő illesztés során is. Tény, hogy a popzenék ennél a változatnál nem kerülnek túlnyomó többségben eredményként meghatározásra, még a saját stílusra vonatkozóan is konfúzió figyelhető meg a rockzenékkal. Spektrogramok alapján

is megfigyelhető volt, és itt is egyfajta hasonlóság a *Country* és *Rock* stílusú zenék között, az alapesetben rockzenének mondott zenék többször country stílusúak, míg az alapvetően *Country* zenék is gyakran értékelődtek ki rockzeneként. A *Latin* stílusú zenék is ezen modell esetén többször kerülnek felismerésre, a legtöbbször helyesen eltalálva azt.

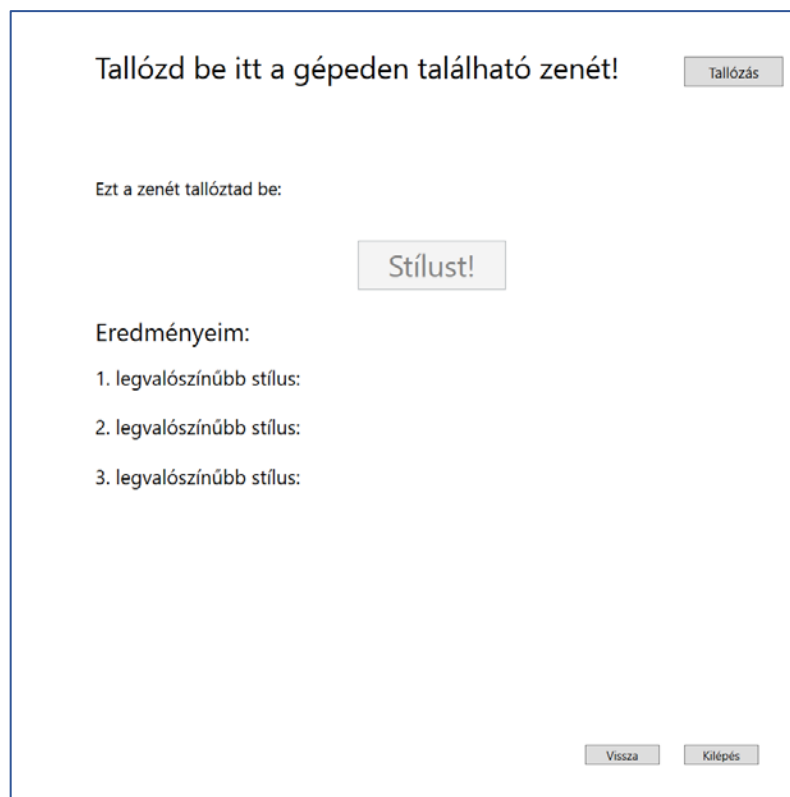
7. Felhasználói dokumentáció

Az elkészített program a megfelelő paraméterek, konfigurációs fájl beállítását követően az alábbi funkciókat engedélyezi a felhasználó számára.



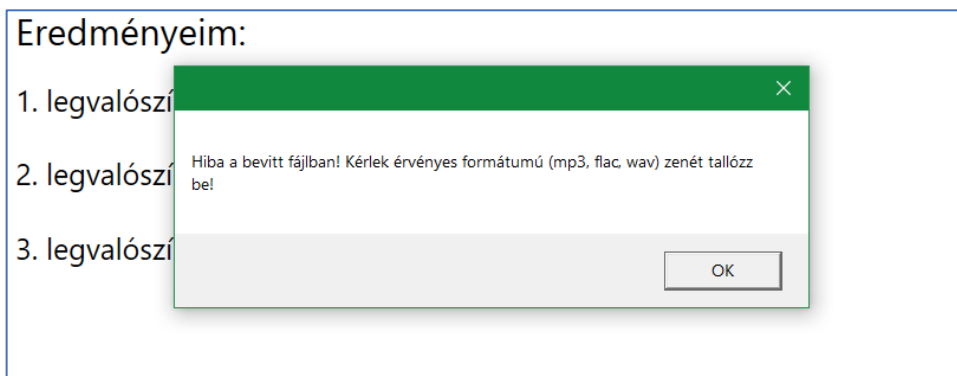
67. ábra A program elindítását követő kezdőképernyő, kiválasztható az offline stílusprediktálás spektrogramokból, valamint online, Spotify felhasználásával

Offline zenemeghatározás esetén betallózható a megfelelő fájl a számítógépről.

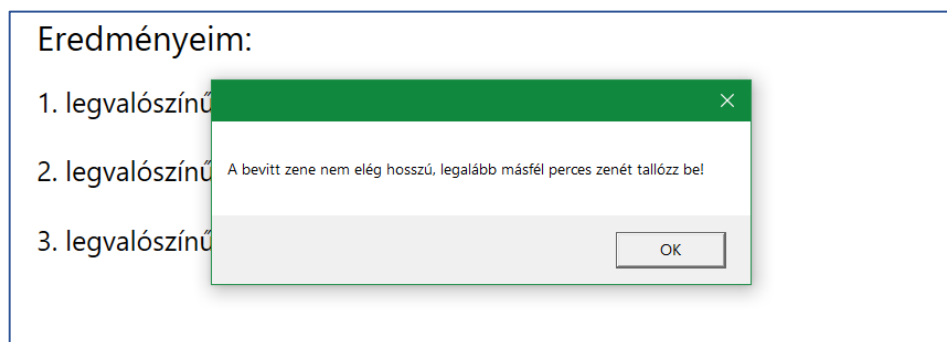


68. ábra Offline zenemeghatározás ablakának kezdőképernyője

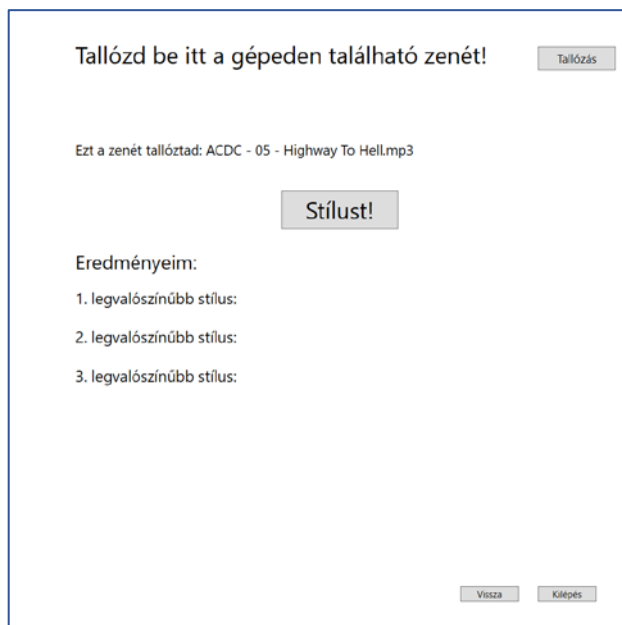
Egész addig inaktív marad a „Stílust!” gomb, amíg betallózásra nem kerül egy megfelelő zenefájl. Amennyiben nem megfelelő zene kerül betallózásra, úgy a program figyelmeztetést tesz.



69. ábra Példa hibásan bevitt fájlra

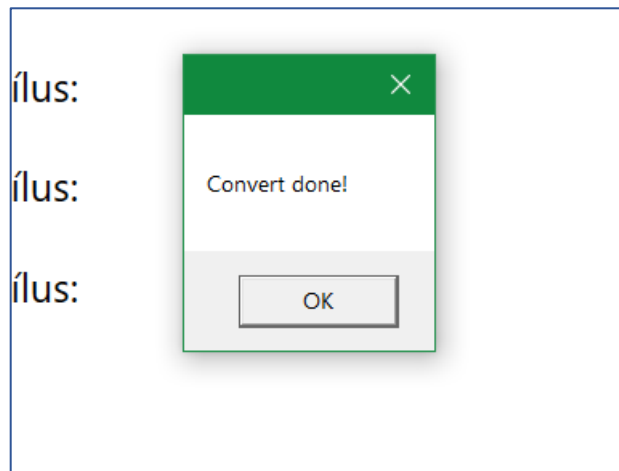


70. ábra Példa helyes formátumú zenefájltra, ami nem megfelelően hosszú

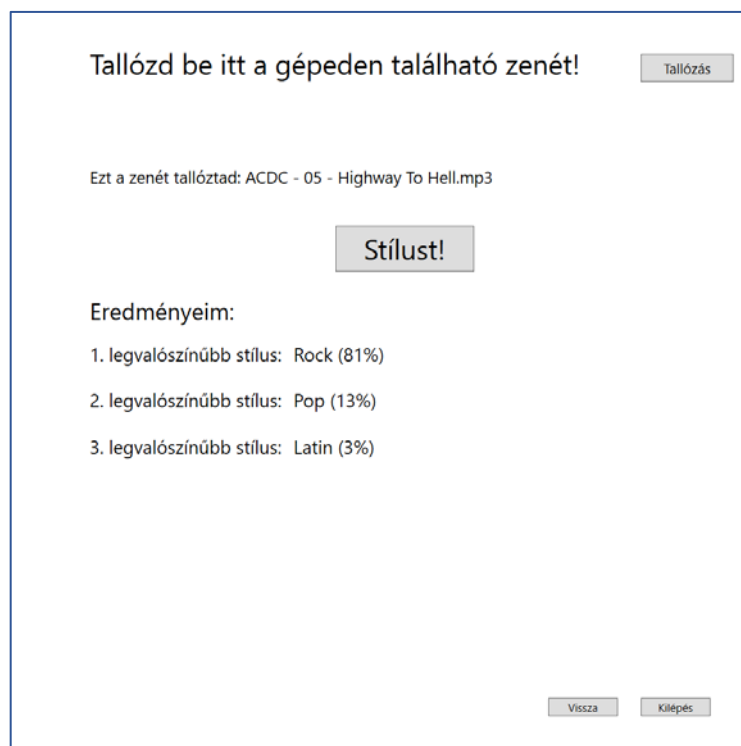


71. ábra Példa helyesen bevitt, kellően hosszú zenefájltra, a „Stílust!” gomb aktiválódásával

Legutóbbi esetben, a „Stílust!” gomb lenyomásával megtörténik a zene átkonvertálása, a folyamatot egy ablakon belüli „OK” gomb lenyomásával lehet folytatni. Ezt követően egy kis idő elteltével, amennyiben előállnak a megfelelő stílusok, úgy kijelzésre kerülnek.



72. ábra Sikeres konvertálás esetén megjelenő ablak, az OK gomb lenyomására megkezdődik az illesztés a modellre



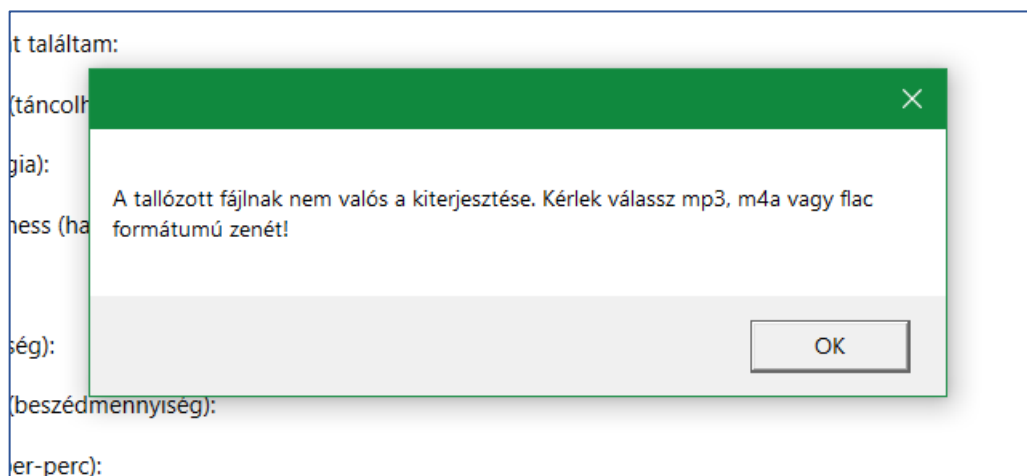
73. ábra Eredmények kijelzése a folyamat végén a felhasználó számára

A megkapott stílusokat követően továbbra is lehet új fájlt betallózni, amely annak validálását követően ismételten végezhető predikció.

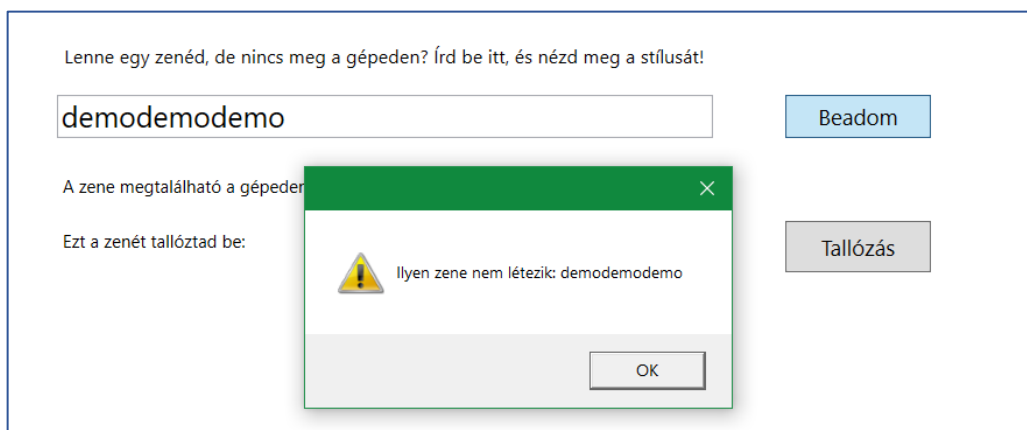
A megfelelő oldalakon jobb alsó sarokban megjelenő Vissza gombbal vissza lehet térni a kezdőképernyőre, míg a Kilépés gomb által az alkalmazásból történő kilépés tehető meg annak megerősítését követően.

Online stílusmeghatározás esetén, amikor Spotify adatokból kerülnek megállapításra a stílusok illetve annak valószínűségei, két opcióból választhat a felhasználó. Egyrészt

rákereshet mintegy közvetlenül a Spotify-on, megtalálható-e rajta zenéje, illetve be is tallózzhatja zenefájlját, amennyiben az számítógépén megtalálható. Fontos, hogy a zene validációja ebben az esetben is szükséges, csak a felsorolt formátumok (mp3, m4a és flac) támogatottak. Bármelyiket is választja a felhasználó, amennyiben megtalálható zenéje Spotify-on, amihez lekérdezhető adat, úgy egyrészt megjelenik a megtalált zene egy előadója a zeneszám címével, valamint az adatok, amelyek a lekérdezés során érkeztek. Fontos, hogy amennyiben többször egymás után kerül betallózásra egy-egy zene, vagy történik egy-egy keresés zenékre, úgy mindig a legutóbbi zenének megfelelő adatok fognak megjelenni a felhasználó számára.



74. ábra Példa nem megfelelő formátumú fájlra – amennyiben nem történik betallózásra fájl, úgy ugyanezen ablak fogadja a felhasználót



75. ábra Példa beírt, de Spotify-on nem létező „zenére”

Lenne egy zenéd, de nincs meg a gépeden? Írd be itt, és nézd meg a stílusát!

A zene megtalálható a gépeden? Ha igen, akkor tallózd be itt!

Ezt a zenét tallóztad be:

A Spotify-on talált zene: Kygo-Firestone (feat. Conrad Sewell)

Ilyen adatokat találtam:

Danceability (táncolhatóság):	0.486
Energy (energia):	0.652
Instrumentalness (hangszerezettség):	6.63E-05
Key (kulcs):	11
Liveness (élőség):	0.077
Speechiness (beszédmennyiség):	0.052
BPM (ütem-per-perc):	113.055

Eredményeim erre a zenére:

1. legvalószínűbb stílus:

2. legvalószínűbb stílus:

3. legvalószínűbb stílus:

76. ábra Példa beírt zenére, amely megtalálható Spotify-on

Lenne egy zenéd, de nincs meg a gépeden? Írd be itt, és nézd meg a stílusát!

A zene megtalálható a gépeden? Ha igen, akkor tallózd be itt!

Ezt a fájlt tallóztad: 02 - All Night.flac

A Spotify-on talált zene: Parovoz Stelar-All Night

Ilyen adatokat találtam:

Danceability (táncolhatóság):	0.793
Energy (energia):	0.578
Instrumentalness (hangszerezettség):	0.926
Key (kulcs):	7
Liveness (élőség):	0.293
Speechiness (beszédmennyiség):	0.0428
BPM (ütem-per-perc):	125.021

Eredményeim erre a zenére:

1. legvalószínűbb stílus:

2. legvalószínűbb stílus:

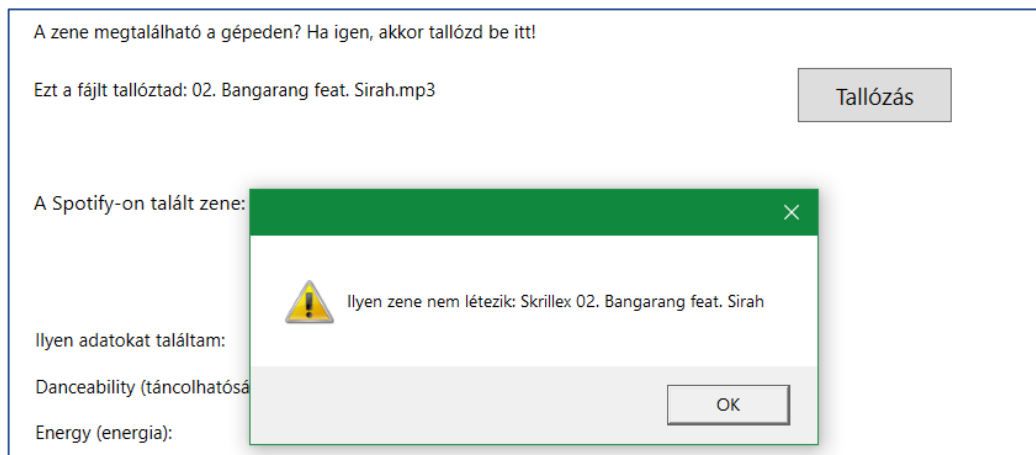
3. legvalószínűbb stílus:

77. ábra Példa helyesen betallózott zenére. Látható, hogy a fenti mező még az előző keresésnek megfelelő, de az adatok lejjebb frissültek a legutóbbi tallózott zenének megfelelően

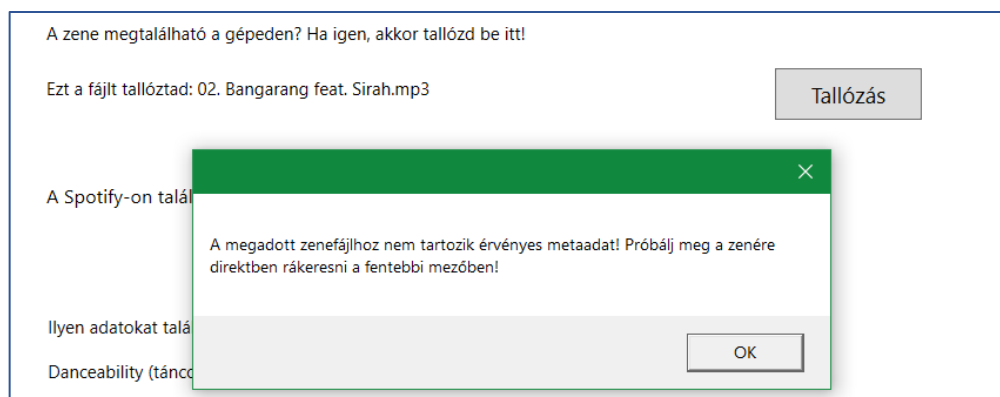
A programban megjelenő tulajdonságok mindegyike részt vesz egy adott stílus meghatározásában, ugyanakkor nem csak ezek a tulajdonságok számítanak, hanem, mivel teszteredmények szerint egy olyan modell bizonyult legjobbnak top3 pontosság szerint, amelyben a jellemzők mindegyike meg van tartva, és így került a programba

bekötésre. Az itt nem említett adatok a háttérben kerülnek eltárolásra, amelyek ugyanúgy számításba vannak véve a predikciók meghatározásában.

Amennyiben „hibás” a beírt metaadat, és nem szabványos, úgy hibaüzenet formájában figyelmeztet a program, egyrészt arra, hogy ilyen zene nem található meg a Spotify-on, másrészt pedig figyelmeztet a nem megfelelően beírt metaadatra. Ekkor továbbra is rá tud keresni a felhasználó a felette lévő mezőben arra a zenére, amelynek stílusára kíváncsi.



78. ábra Figyelmeztető ablak Spotify-on meg nem talált zene esetén



79. ábra Figyelmeztető ablak nem megfelelően beírt metaadat esetén, amelyben javaslatot tesz a folytatásra

Lenne egy zenéd, de nincs meg a gépeden? Írd be itt, és nézd meg a stílusát!

A zene megtalálható a gépeden? Ha igen, akkor tallózd be itt!

Ezt a fájlt tallóztad: 02. Bangarang feat. Sirah.mp3

A Spotify-on talált zene: Skrillex-Bangarang (feat. Sirah)

Ilyen adatokat találtam:

Danceability (táncolhatóság):	0.716
Energy (energia):	0.972
Instrumentalness (hangszerezettség):	3.22E-05
Key (kulcs):	7
Liveness (élőség):	0.317
Speechiness (beszédmennyiség):	0.196
BPM (ütem-per-perc):	110.026

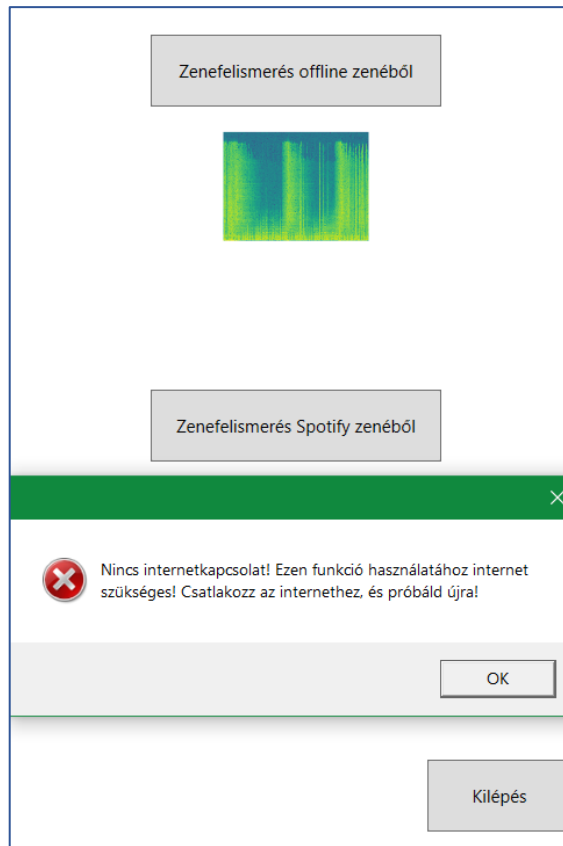
80. ábra Keresés utáni állapot eredetileg nem megfelelő metaadattal rendelkező zene esetén

Bármelyik opcióval sikerül engedélyezni az ablak jobb alsó részén található „Stílust!” gombot, rákattintva, várakozva egy keveset, megjelenésre kerül a három legvalószínűbb stílus a megfelelő valószínűségekkel, hasonlóan, mint a spektrogramok esetén.

Speechiness (beszédmennyiség):	0.196
BPM (ütem-per-perc):	110.026
Eredményeim erre a zenére:	Skrillex - Bangarang (feat. Sirah)
1. legvalószínűbb stílus:	Electronic (47%)
2. legvalószínűbb stílus:	Latin (34%)
3. legvalószínűbb stílus:	Pop (8%)

81. ábra Eredmények kijelzése a folyamat végén a felhasználó számára online alapú meghatározás esetén

Fontos tényező, hogy ezen utóbbi megoldáshoz internetkapcsolat szükséges. Ennek tényét már a megfelelő opció kiválasztásával vizsgálandó a program megnyitásakor, amennyiben a felhasználó internetkapcsolat nélkül próbálná használni ezen funkciót, úgy egy ablakban jelzésre kerül az internetkapcsolat hiánya.



82. ábra Példa Spotify adatokon alapuló predikcióra internetkapcsolat hiányában

8. Kutatási-fejlesztési lehetőségek

A szakdolgozat elkészültével felmerültek olyan fejlesztési lehetőségek, amelyek abban az esetben kerülhetnek leghatékonyabban kivitelezésre, ha adott lenne még a készítéshez egy tágabb időkeret, lehetne még rajta dolgozni.

Az egyik ilyen lehetőséget általánosan társítani lehet az olyan jellegű feladatok megoldásához, amelyhez a gépi tanulás mint alap eszköz kerül felhasználásra. A tanítás a jelenlegi adatokkal biztosított egyfajta pontosságot, ugyanakkor ha még több adat állna rendelkezésre a tanításhoz, valamint az azt ellenőrző validációs halmazhoz, úgy nagyobb pontosságot lehet elérni. 100%-os pontosságot még így sem lehetne elérni, hiszen ahhoz az összes zene ismerete a megfogalmazandó cél, és a tárhely, amely tárolja a mintákat, véges, sosem végtelen.

A tanítások során bizonyossá vált, hogy azon változatok, amelyek alacsony *learning rate* tulajdonsággal rendelkeznek, képesek finomabban megtalálni a jobbnál jobb állapotokat, Viszont ezen érték csökkentésével a tanítási idő fordítottan arányosan növekszik, minél kisebb, annál több időt vesz igénybe egy-egy ilyen tanítás. Több idő rendelkezésre állásával meg lehet azt csinálni, hogy egy ilyen tanítást megindítva, egy jelentősebb toleranciát választva megállási feltételnek, több napig, akár több hétig lehetne tanítani. Ezzel kapcsolatosan, előnyre válhat, ha a *batchsize* is magasabb értéket vehetne fel, viszont ez a memória igény növekedésével jár. Ehhez egy nagyobb dedikált memóriával rendelkező videokártya, vagy videokártyák lehetnének segítségére.

A fejlesztés során több konvolúciós neurális hálózat is ki lett próbálva, és össze lett hasonlítva, melyik rendelkezik a legnagyobb pontossággal, hogyan viszonyulnak egymáshoz a tanítási idők. Ugyanakkor nem csak ezek a kész modellek állnak rendelkezésre, több state-of-the-art módszer is kipróbálásra kerülhetne, nem csupán néhány változata.

A zenével kapcsolatos tulajdonságok egy online zene streaming szolgáltatáson keresztül kerültek lekérdezésre, tárolásra, majd pedig a tanítás során feldolgozásra. Ez tíz kategóriát jelentett. További fejlesztési lehetőségként merül fel, hogy további adatok kinyerése során pontosabb megoldás születne-e. Ehhez a zenék mindegyikén végrehajtható lenne egy-egy komolyabb komponens analízis, melynek során a zenei hangfájlból közvetlenül kerülnének kiolvasásra egyes tulajdonságok számértékei. Ehhez viszont a zenék tömörítetlen formátumban tárolása szükségeltetik, amely a tárhely igény növekedésével járna együtt.

Alapesetben az alkalmazás egy asztali program formájában került elkészítésre, viszont felmerülhet, hogy magát a rendszert egy felhő alapú szolgáltatásként meg lehetne valósítani hatékony módon. Ennek keretében elkészülhetne a program mobilos változata is, hasonlóan a Shazam alkalmazáshoz, amelynek során nem konkrét zene, hanem zenestílus kerülne meghatározásra, továbbá hasonló stílusú zenék ajánlása is megtörténhetne, ha az adatbázist alkotó zenék rendszere megfelelően robusztus, hogy erre képes legyen.

9. Összefoglalás

Szaktervezőm kísérletet tett arra, hogy a stílusokat egyértelműen megnevezni meg nem tudó felhasználók számára egyfajta alternatívát adjon, hogy a jelenleg elérhető adatok és technológia segítségével a ma is ismert zenék milyen zenei stílussal rendelkeznek. Ez azonban nem 100%-os pontosságú, ahogyan nem is lehet 100%-os pontossággal valószínűségeket megállapítani, több oknál fogva is. Egyrészt a zenei stílusok meghatározása nem teljes mértékben egyértelmű, az ismert stílusok között nem említhető meg egy éles határvonal, amely tudományos módon is alátámasztott. Másrészt ahhoz, hogy egy zenéről 100%-osan meg lehessen állapítani egy konkrét stílust, úgy a lehető legtöbb zenét a tanító modell számára kellett volna biztosítani, amely azonban tárhely hiányában meglehetősen nehéz feladatnak minősül. Ellenben egy-egy meglehetősen pontos modell készült, relatív kevés zene felhasználásával, amelyekkel általánosan meg lehet állapítani igen ismert stílusokat.

Az irodalomkutatásból számos dologra fény derült. A zenei stílusok meghatározása már régen foglalkoztatja az ezzel foglalkozókat, hogyan lehet minél több stílusra vonatkozóan minél nagyobb pontosságú modellt felállítani. Az eddigi eredmények szerint a zenékből mégis olvasható ki olyan adat, amelyek segítségével tanítás indítható meg ezzel kapcsolatban. Van olyan tanulmány, amelyben konkrét, számszerű adatok segítségével történik ez meg, vannak olyan módszerek, amelyek a zenét képként reprezentált spektrogramok formájában dolgozza fel, és szerzi meg a lehető legtöbb információt. Sor derült arra is, mi a rendkívül népszerűnek számító Shazam alkalmazásának módja, mely ugyanúgy spektrogramokból, és az azok által hordozott információkból indul ki, és határoz meg egy-egy zenét meglehetősen nagy pontossággal. Mindegyik megközelítés az információkat neurális hálózatok számára adta át, a képeket ennek egy továbbgondolása szerint konvolúciós neurális hálózatban tanította be. Az egyes zenei stílusokkal kapcsolatos információk összeszedése is kihívás volt, hiszen minden bizonnyal megalapozott cikkek, beszámolók, könyvrészletek csak nagyon alacsony számban figyelhetőek meg, illetve csak néhány stílus esetén. Már abból is kiolvasásra került, hogy az egyes stílusok között figyelhető meg összefüggés, azonos elemek kerülnek felsorakoztatásra, előadásmódtól kezdve a hangszereken át.

A program két fő komponensre, modellre alapoz. Egy olyanra, amelyben a zenei stílusok meghatározása az internet nélkül, offline, spektrogramok segítségével képek formájában kerül megalkotásra, továbbá egy olyanra, amelyben az internetet felhasználva kerülnek beszerzésre a zenékhez tartozó adatok megadott tulajdonságok szerint. A spektrogram alapú tanításhoz a mintákat ismert zenék 45 másodperces részletei határozták meg, melyekből egy egyszerű, nem többlépcsős feldolgozást követően alakultak ki a megfelelő képek. Az online alapú stílusdetektáláshoz pedig az adatokat egy ismert zene streamer szolgáltatás, a Spotify adatbázisából kerültek meghatározásra 10 tulajdonság szerint. A pusztán adatokra történő tanítás működésére bizonyosságot a detektálni kívánt stílusok összegyűjtött adataira vonatkozó statisztikai

elemzések bizonyították be, egyes zenék jellemzői ugyanis különböztek annyival, hogy ezen adatokból lehessen utalni egy-egy jellemző stílusra. Ezek közül a gyakorlatban leginkább szemléltethető percenkénti ütemszám, avagy BPM egy bővebb elemzésen esett át, amellyel alátámasztásra került, hogy az adatok ténylegesen eltérnek egyes stílusok esetén, így több tulajdonság figyelembevételével elkülöníthetőnek mutatkoztak az egyes stílusok egymástól.

A tanítás kezdetben 15 kategóriára indult meg. Legelőször egy alacsony számú tanító paraméterrel rendelkező modell, a *MobileNetV2* lett kipróbálva, melynek során nagyon alacsony validációs pontosság került meghatározásra a nagyjából 10-15%-os értékekkel. Ezt követően, alapul véve a *VGG16* nevű modellt, az eredmények úgy is javulásnak indultak a maguk 40-45%-os értékével, hogy a modell egyszerűsítések során ment át, melynek oka a rendkívül összetett alap architektúra, valamint a jelenleg megoldandó feladat kisebb komplexitása. Ezen eredmények alapján bebizonyosodott, hogy a feldolgozandó stílusok között igen nagy összefüggés és hasonlóság figyelhető meg. Egyes tanulmányokat, kutatásokat és további elemzéseket felhasználva 9 kategóriára zsugorodott a felismerni kívánt stílusok mennyisége.

A fennmaradó 9 stílusra vonatkozóan state-of-the-art módszerek mellett saját készítésű konvolúciós hálózat is használatra került. A *VGG16* megoldása mellett alkalmazásra került a *NASNet* nevű modellcsoport legegyszerűbbnek vélt, legkevesebb paraméterrel rendelkező, de így is számítási igényét hatékonyabbnak tekinthető *NASNetMobile*-féle modell, amelynek 30%-hoz közeli eredményei gyengébbnek bizonyultak az eddig megfigyelteknél. A *VGG16* és a saját megoldásokkal tovább kísérletezve és tanítva eredményei a különböző optimalizálókat figyelembe véve határozott javulásnak indultak, az eredmények top1 pontosság szerint közelítettek a 60%-hoz, míg a top3 pontosság szerint a 80%-ot is meghaladta. Ez utóbbi mérőszáma a felhasználónak megjelenő stílusoknak, hiszen a 3 leginkább várt stílus, valamint azokhoz tartozó valószínűségek kerülnek kiíratásra.

A spotifyos adatokra vonatkozóan megállapításra került, hogy a korrelált, de mégis több adat hatékonyabbnak bizonyul tanítás szempontjából. Korrelációanalízist követő *fully-connected* neurális hálózat valamennyivel gyengébb eredményt mutattak, mint azon modellek, amelyeknél ezen adatok engedélyezettek. Korrelált adatok szerint a top1 pontosság közel volt a 60%-hoz, a top3 pontosság pedig majdnem elérte a 90%-ot, amely esetenél megállapítható, hogy nagy általánosságban 10-ből mindössze egyszer nem kerül kiíratásra az a várt stílus, amely egy adott zenére meg van adva. Mind a validációs halmazra vonatkozó pontosságok, mind pedig a teszhalmazra vonatkozó eredmények szerint megállapítható, hogy általánosságban a zenei adatokból nagyobb pontosság érhető el, mint spektrogramok szerint.

A top1 pontosságok egy más ábrázolásából, a konfúziós mátrixból kiolvasásra kerülhettek, hogy melyek azok a stílusok, amik nagy pontossággal megállapításra kerültek, és melyek között figyelhető meg egyfajta összefüggés a stílusok között. Spektrogramok szerint általánosságban megfigyelhető, hogy a gyűjtőfogalomként is

megfeleltethető populáris, népszerű zenék, azaz popzenék gyakran kerülnek felismerésre. A spektrogram alapú és adatalapú mátrixokból pedig következik, hogy a hiphop és a rap stílusok nemcsak leírásukban hasonlítanak egymásra, hanem eredmények szerint is.

Fontos kihangsúlyozni, hogy a várt stílus továbbra sem jelenti azt, hogy annak a zenének ténylegesen és kizárólag az a stílusa, amely az meg van adva az adott zene számára. Egy zene több stílus jegyeit is hordozhatja magán, illetve, ha egy felhasználó meg van kérdezve, hogy mondja meg egy adott zene stílusát, úgy többen többfajta mondanának első helyen. Nincs konkrét stílusmeghatározás, amely tudományos módon alá lenne támasztva, és amellyel ezek a zenék egyértelműen diszkriminálhatóak lennének, meglehetősen szubjektív vélemény van alkotva az egyes stílusokról. A megoldás pontosan erre akart egyfajta megközelítést mutatni. Megmondani, hogy adott zenével kapcsolatosan mi lehet annak stílusa a benne felismert különböző tulajdonságok és adatok szerint.

10. Summary

My thesis work attempted to provide a kind of alternative with the available data and technology for users who cannot clearly name those music genres that what kind of genre or genres have those, well-known musics. However, this is not 100% accurate, just as it is almost impossible to determine genres with 100% accuracy, for several reasons. The definition of musical genres are not entirely clear, there are no sharp lines between the known genres that has been scientifically substantiated, and the ability to be able to determine a specific genre of a music with a 100% of accuracy, as much music as possible should have been provided for the training model, which is, however, considered to be a rather difficult task in the absence of storage space. On the other hand, fairly accurate models were made, using relatively few music, but very well-known genres could be determined in general.

A number of findings have been made from the literature research. Defining music genres has long preoccupied practitioners how to set up a model with the highest possible accuracy for as many genres as possible. According to the results so far, some kind of data can be read from the music to start training models with those. There is a study in which this procedure is done with the help of specific, numerical data, there are methods that process music in the form of spectrograms represented as images and obtain as much information as possible. It was also revealed that how the famous and popular music recognizer, the Shazam application works, which, in the same way, starts from spectrograms and the information those carry, and defines a piece of music with a fairly high percent of accuracy. Each approach passed the information to neural networks, and according to a further thought, training the images in a convolutional neural network. Gathering information about each genre of music was also a challenge, as well-founded articles, reports, book excerpts can only be observed in very small numbers, or only for a few genres. It has been read from the fact that connections can be observed between the individual genres, same elements are lined up, starting from the mode of performance, the used instruments and so on.

The program is based on two main components, models. One of these defines music genres offline in the form of images without the use of internet, with spectrograms, and other of these, data relating to music is obtained using the internet according to specified properties. For spectrogram-based training, the samples were determined by 45-second pieces of known music, from which the corresponding images were formed after a simple, non-multi-step processing. For online-based genre detection, the data was gathered from the database of the well-known music streamer service, Spotify, by 10 properties. The certainty of the operation of training on data alone was demonstrated by statistical analyzes of the collected data of the genres to be detected, as the characteristics of some music differed so much that these data could be used to refer to a characteristic genre. One, and maybe the most illustrated property of all, the beats-per-minute, or BPM, underwent a more detailed analysis to show that the data

actually differed for each genre, so that several genres could be distinguished from each other by considering a few other properties.

Training initially started for 15 categories. First, a model with a low number of trainable parameters, MobileNetV2 was tested, but a very low validation accuracy was resulted with values of approximately 10-15%. Subsequently, based on the model, called VGG16, the results started to improve with their 40-45% values, with the model which underwent simplifications due to extremely complex basic architecture and the less complexity of the task currently to be solved. Based on these results, it has been shown that there is a measurable correlation and similarity between the genres to be processed. Using some other studies, researches and analysis, the amount of genres to be recognized has shrunk to 9 categories. For the remaining 9 genres, in addition of state-of-the-art methods, a self-made convolutional network was also used. Next to the solution of VGG16, the NASNetMobile model, considered to be the simplest model type of the NASNet group of models, also has the fewest parameters, but still considered more efficient in terms of computational requirements, was used, the results were close to 30%. Further experimenting and training with VGG16 and the own-made solutions, the results started to show a definite improvement considering the different optimizers, the results approached 60% in terms of top1 accuracy, and even exceeded 80% in terms of top3 accuracy. The top3 accuracy is the measure of the genres that appear to the user, as the 3 most expected genres and their associated probabilities are displayed.

For data which gathered through Spotify, it was found that correlated, but more kind of data, proved to be more effective for training. Fully-connected neural network following the correlation analysis showed slightly weaker results than models for which these data were allowed. According to correlated data, the top1 accuracy was close to 60%, the top3 accuracy was almost 90%, in which case, it can be stated, only once out of 10, the expected genre of a music is not printed for the user, in general. Both the accuracies for the validation set and the results for the test set showed that greater accuracy can be obtained from music data than from spectrograms, also in general. It was possible to read the results from another representation of the top1 accuracies, the confusion matrix, which genres were determined with high accuracy and what were the genres that showed correlation. According to the results from spectrograms, it can be generally observed that popular music, pop music, which can also be equated as a collective name, is often recognized. Also it follows from the spectrogram-based and data-based matrices that hiphop and rap genres are similar not only in their description, but also in terms of results.

It is important to emphasize that the expected genre still does not mean that the genre of music that is actually and exclusively given to that particular music. A music can carry features of more than one genre, or if a user is asked to say the exact genre of music, the more people is asked, the more genre is mentioned first. There is no specific definition of genres that is scientifically substantiated, and with these music can be clearly discriminated against, a rather subjective opinion is formed about each genre.

The solution wanted to show a kind of approach for that. In order to say that what genre can be in relation to a given music according to the different characteristics and data recognized in it.

Ábrajegyzék

1. ÁBRA „CASCADE-CORRELATION” ARCHITEKTÚRA SZEMLÉLETES ÁBRÁN, AHOL KÉT REJTETT RÉTEG KERÜL HOZZÁADÁSRA A RENDSZERHEZ [2].....	5
2. ÁBRA A 20 LEGINKÁBB BEFOLYÁSOS TULAJDONSÁG MEGNEVEZÉSE, F-SCORE ÉRTÉK ALAPJÁN [4].....	10
3. ÁBRA EGY VGG16 MODELLRE TÖRTÉNŐ TANÍTÁS LEGJOBB EREDMÉNYEIT TARTALMAZÓ KONFÚZIÓS MÁTRIX – A FŐÁTLÓBAN A HELYES PREDIKCIÓK SZÁMÁVAL [4]	11
4. ÁBRA EGY ADOTT ZENE SPEKTROGRAMJA AZ IDŐ-FREKVENCIA FÜGGVÉNYBEN [28].....	16
5. ÁBRA A KOMBINATORIKUS HASH GENERÁLÁSA EGY KIINDULÓ, "ANCHOR" PONTBÓL [28].....	17
6. ÁBRA SIKERTELEN ILLESZTÉS, A HASH ÉRTÉKEK SZÉTSZÓRÓDVA HELYEZKEDNEK EL. [28] X TENGELY MENTÉN SZEREPELNEK AZ ADATBÁZISBAN LÉVŐ ADATOK, MÍG Y TENGELY MENTÉN A MINTAKÉNT BEADOTT ZENÉNEK MEGFELELŐ ÉRTÉKEK	18
7. ÁBRA SIKERES ILLESZTÉS - JÓL LÁTHATÓ A 40. ÉS 60. MÁSODPERC KÖZT AZ ÁTLÓBAN SÚRÚSODÓ PONTOK HALMAZA, AZ ILLESZTÉS A ZENE 40. MÁSODPERCÉTŐL KEZDŐDIK MEG. [28].....	18
8. ÁBRA PÉLDA EGY KONVOLÚCIÓS NEURÁLIS HÁLÓZATRA – KLASSZIFIKÁCIÓ KÜLÖNBÖZŐ JÁRMŰVEKRE [30].....	19
9. ÁBRA EGY 6x6-OS KÉPEN 3x3-AS MÉRETŰ KERNELLEL VÉGZETT KONVOLÚCIÓS MŰVELETEK ÁLTAL ELŐÁLLÓ 4x4-ES MÁTRIX [32].....	20
10. ÁBRA PÉLDAKÉPEK A ZERO-PADDING TULAJDONSÁG TÍPUSAIRA ÉS LEHETSÉGES DIMENZIÓ-KITERJESZTÉSEIRE: VALID (BAL), SAME (KÖZÉP) ÉS FULL (JOBB) [32].....	21
11. ÁBRA A KONVOLÚCIÓS MÁTRIXON ALKALMAZHATÓ AKTIVÁCIÓK FÜGGVÉNYEI, ÁLTALUK ADOTT KIMENETI ÉRTÉKEIK AZ Y TENGELY MENTÉN SZEREPELTETVE: RELU (BAL), LEAKYRELU (KÖZÉP) ÉS ELU (JOBB) [32]	22
12. ÁBRA A VGG16 ARCHITEKTÚRÁJA SZEMLÉLETES ÁBRÁN – ITT AZ UTOLSÓ SOFTMAX AKTIVÁCIÓ AZ IMAGE NET KIHÍVÁSNAK MEGFELELŐEN 1000 KATEGÓRIÁRA VONATKOZIK [34]	23
13. ÁBRA A VGG16 ARCHITEKTÚRÁJÁNAK ÖSSZEFOGLALÓJA A BENNE SZEREPLŐ RÉTEGEK SZERINT, 15 KATEGÓRIÁRA VONATKOZÓAN	24
14. ÁBRA MNIST KÉPEKEN TÖBBRÉTEGŰ NEURÁLIS HÁLÓZATTAL VÉGREHAJTOTT TANÍTÁS EREDMÉNYEI A KÜLÖNBÖZŐ OPTIMALIZÁLOK EREDMÉNYEIVEL A TANÍTÁSI VESZTESÉGÉRTÉKEKEN. LILA SZÍNNEL JELÖLVE AZ ADAM OPTIMALIZÁLÓ [38]	26
15. ÁBRA RESNET50 MODELLEN ALAPULÓ TANÍTÁS EREDMÉNYEI 512x512 NAGYSÁGÚ KÉPEKEN, TANÍTÁSI PONTOSSÁGÉRTÉKEK (BAL OLDALON) ÉS TANÍTÁSI VESZTESÉGÉRTÉKEK TEKINTETÉBEN. EZEN MODELLEN A LEGJOBB OPTIMALIZÁLOKÉNT AZ ADAMAX (KÉK) KERÜLT KI [39]	26
16. ÁBRA MOBILENETV2 ARCHITEKTÚRÁJA, AHOL „C: A KIMENETI CSATORNÁK SZÁMA, T: AZ ÚGYNEVEZETT KITERJESZTÉSI FAKTOR, N: AZ ISMÉTLÉS DARABSZÁMA, ÉS S: A STRIDE MÉRETE AZ ELSŐ ISMÉTLÉS ALKALMÁVAL, AHOL $N > 1$, AZ EZT KÖVETŐ ISMÉTLÉSEKBE PEDIG 1” [41].....	27
17. ÁBRA A BOTTLENECK RÉTEG FELÉPÍTÉSE, A TRANSZFORMÁCIÓS ABLAKBAN KONVOLÚCIÓS - MÉLYSÉGI („DWISE”) KONVOLÚCIÓS - KONVOLÚCIÓS RÉTEGEK VÁLTOZÁSÁVAL ALAKUL KI A KIMENET [42].....	27
18. ÁBRA NASNET MODELL PONTOSSÁGA PIROSSAL KIEMELVE AZ ÖSSZEADÁS-SZORZÁS (MULTIPLY-ADD) MŰVELETEK SZERINT, LÁTHATÓ MÓDON A MOBILE VÁLTOZATA IS KÖZEL 75%-OS EREDMÉNYT ADVA [43]	28
19. ÁBRA AZ ACOUSTICNESS TULAJDONSÁG JELLEMZŐ ÉRTÉKEINEK ELOSZLÁSA [45]	30
20. ÁBRA A DANCEABILITY TULAJDONSÁG JELLEMZŐ ÉRTÉKEINEK ELOSZLÁSA [45].....	30
21. ÁBRA AZ ENERGY TULAJDONSÁG JELLEMZŐ ÉRTÉKEINEK ELOSZLÁSA [45]	31
22. ÁBRA AZ INSTRUMENTALNESS TULAJDONSÁG JELLEMZŐ ÉRTÉKEINEK ELOSZLÁSA [45].....	31
23. ÁBRA A LIVENESS TULAJDONSÁG JELLEMZŐ ÉRTÉKEINEK ELOSZLÁSA [45]	32
24. ÁBRA A LOUDNESS TULAJDONSÁG JELLEMZŐ ÉRTÉKEINEK ELOSZLÁSA [45]	32
25. ÁBRA A SPEECHINESS TULAJDONSÁG JELLEMZŐ ÉRTÉKEINEK ELOSZLÁSA [45]	33
26. ÁBRA A TEMPO (BPM) TULAJDONSÁG JELLEMZŐ ÉRTÉKEINEK ELOSZLÁSA [45]	33
27. ÁBRA A VALENCE TULAJDONSÁG JELLEMZŐ ÉRTÉKEINEK ELOSZLÁSA [45]	33
28. ÁBRA A TANÍTÓ ADATHALMAZRA VONATKOZÓ BPM ÉRTÉKEK PONTDIAGRAMON ÁBRÁZOLVA, EGY-EGY ZENESTÍLUST KÖRVONALAZVA, A KÜLÖN STÍLUSOK KÜLÖN SZÍNNEL JELÖLVE.....	34
29. ÁBRA DANCEABILITY ÉRTÉKEK A 15 KATEGÓRIÁNAK TANÍTÓHALMAZÁRA VONATKOZÓAN PONTDIAGRAMON ÁBRÁZOLVA	35
30. ÁBRA ENERGY ÉRTÉKEK A 15 KATEGÓRIÁNAK TANÍTÓHALMAZÁRA VONATKOZÓAN PONTDIAGRAMON ÁBRÁZOLVA	35
31. ÁBRA INSTRUMENTALNESS ÉRTÉKEK A 15 KATEGÓRIÁNAK TANÍTÓHALMAZÁRA VONATKOZÓAN PONTDIAGRAMON ÁBRÁZOLVA.....	36
32. ÁBRA LIVENESS ÉRTÉKEK A 15 KATEGÓRIÁNAK TANÍTÓHALMAZÁRA VONATKOZÓAN PONTDIAGRAMON ÁBRÁZOLVA	36

33. ÁBRA LOUDNESS ÉRTÉKEK A 15 KATEGÓRIÁNAK TANÍTÓHALMAZÁRA VONATKOZÓAN PONTDIAGRAMON ÁBRÁZOLVA.....	36
34. ÁBRA SPEECHINESS ÉRTÉKEK A 15 KATEGÓRIÁNAK TANÍTÓHALMAZÁRA VONATKOZÓAN PONTDIAGRAMON ÁBRÁZOLVA..	37
35. ÁBRA VALENCE ÉRTÉKEK A 15 KATEGÓRIÁNAK TANÍTÓHALMAZÁRA VONATKOZÓAN PONTDIAGRAMON ÁBRÁZOLVA.....	37
36. ÁBRA ACOUSTICNESS ÉRTÉKEK A 15 KATEGÓRIÁNAK TANÍTÓHALMAZÁRA VONATKOZÓAN PONTDIAGRAMON ÁBRÁZOLVA	37
37. ÁBRA A RENDSZER KOMPONENS-DIAGRAMJA	41
38. ÁBRA A RENDSZER SZEKVENCIA-DIAGRAMJA EGY OFFLINE ALAPÚ, SPEKTROGRAM SZERINTI PREDIKCIÓ ESETÉN	42
39. ÁBRA A RENDSZER SZEKVENCIA-DIAGRAMJA EGY ONLINE ALAPÚ, SPOTIFY ADATOK SZERINTI PREDIKCIÓ ESETÉN	43
40. ÁBRA A RENDSZER FEJLESZTÉSÉRE VONATKOZÓ GANTT-DIAGRAM	44
41. ÁBRA PÉLDA EGY BEÍRT STÍLUSSAL NEM RENDELKEZŐ MAPPÁN BELÜLI ZENÉK FELCÍMKÉZÉSÉRE. ITT MINDEN, CHANSON MAPPABELI ZENE ELLÁTÁSRA KERÜL JAZZ, ILLETVE POP STÍLUSSAL.....	46
43. ÁBRA PÉLDA ELKÉSZÜLT SPEKTROGRAMOKRÓL.....	47
44. ÁBRA AZ ÖSSZEGYÜJTÖTT ALBUMLINKEKEN VÉGIG ITERÁLVA ÖSSZESZEDENDŐ AZOK ZENÉINEK AZONOSÍTÓJA, AMENNYIBEN AZOK SPOTIFY-ON ÉS NEM A FELHASZNÁLÓK HELYI TÁRHELYÉN TALÁLHATÓ MEG	49
45. ÁBRA AZ ÖSSZEGYÜJTÖTT AZONOSÍTÓK ALAPJÁN A MEGFELELŐ INFORMÁCIÓK ÖSSZESZEDÉSE, KIGYÚJTÉSE, AMELYEK EGY FÁJLBA KERÜLNEK KIÍRÁSRA, HA ANNAK HOSSZA EGY PERCNÉL NAGYOBB	50
46. ÁBRA TANÍTÁS 20 EPOCHON KERESZTÜL MOBILENetV2 ARCHITEKTÚRÁVAL, A PONTOSSÁG- (BALRA) ÉS A VESZTESÉGÉRTÉKEK TEKINTETÉBEN. A NARANCS SZÍNŰ ÉRTÉKEK A TANÍTÓHALMAZHOZ, MÍG A KÉK SZÍNŰ ÉRTÉKEK A VALIDÁCIÓS HALMAZHOZ TARTOZNAK.	50
47. ÁBRA TANÍTÁS 100 EPOCHON KERESZTÜL MOBILENetV2 ARCHITEKTÚRÁVAL, A PONTOSSÁG- (BALRA) ÉS A VESZTESÉGÉRTÉKEK TEKINTETÉBEN. A BORDÓ SZÍNŰ ÉRTÉKEK A TANÍTÓHALMAZHOZ, MÍG A KÉK SZÍNŰ ÉRTÉKEK A VALIDÁCIÓS HALMAZHOZ TARTOZNAK.	50
48. ÁBRA TANÍTÁS 150 EPOCHON KERESZTÜL MOBILENetV2 ARCHITEKTÚRÁVAL, A PONTOSSÁG- (BALRA) ÉS A VESZTESÉGÉRTÉKEK TEKINTETÉBEN. A KÉK SZÍNŰ ÉRTÉKEK A TANÍTÓHALMAZHOZ, MÍG A LILA SZÍNŰ ÉRTÉKEK A VALIDÁCIÓS HALMAZHOZ TARTOZNAK.	50
49. ÁBRA VGG16 ARCHITEKTÚRÁJÁNAK EGY EGYSZERŰSÍTÉSÉVEL MEGKEZDETT TANÍTÁS EREDMÉNYEI A PONTOSSÁG (BALRA) ÉS A VESZTESÉGÉRTÉKEK TEKINTETÉBEN. SZÜRKE SZÍNNEL JELÖLVE A TANÍTÓ HALMAZHOZ, MÍG NARANCSSZÍNNEL JELÖLVE A VALIDÁCIÓS HALMAZHOZ TARTOZÓ ÉRTÉKEKET.....	51
50. ÁBRA VGG16 ARCHITEKTÚRÁJÁNAK EGY EGYSZERŰSÍTÉSÉVEL MEGKEZDETT TANÍTÁS EREDMÉNYEI A PONTOSSÁG (BALRA) ÉS A VESZTESÉGÉRTÉKEK TEKINTETÉBEN, ADAM OPTIMALIZÁLÓVAL. KÉK SZÍNNEL JELÖLVE A TANÍTÓ HALMAZHOZ, MÍG RÓZSASZÍNNEL A VALIDÁCIÓS HALMAZHOZ TARTOZÓ ÉRTÉKEKET.....	52
51. ÁBRA STÍLUSOK MEGHATÁROZÁSÁHOZ DEKÓDOLÓ KÓDRÉSZLET 15 KATEGÓRIA ESETÉN, AMELYBEN PARAMÉTERÜL AZ UTOLSÓ, SOFTMAX RÉTEG ÁLTAL ADOTT EREDMÉNYEK KERÜLNEK MEGADÁSRA, ABBÓL A LEGJOBB 3 KERÜL A FELHASZNÁLÓ SZÁMÁRA KIJELEZÉSRE.....	52
52. ÁBRA KÜLÖNBÖZŐ STATISZTIKAI MEGHATÁROZÁSOK AZ EGYES STÍLUSOKRA VONATKOZÓAN, LEGGYAKRABBAN, ILLETVE LEGTÖBBET HALLGATOTT ZENÉK STÍLUSAI SZERINT [51] [52].....	53
53. ÁBRA STÍLUSCSOPORTOSULÁSOK A MYSPACE KÖZÖSSÉGI OLDALON AKTÍV ZENÉSZEK, ELŐADÓK KÖZÖTT, CSOMÓPONTI SZÍNNEL JELÖLVE A NAGYOBB CSOPORTOKAT, TOVÁBBÁ AZ ÉLEK VASTAGSÁGA A STÍLUSOK EGYÜTTES SZEREPLÉSÉNEK GYAKORISÁGÁVAL ARÁNYOS [53]	53
54. ÁBRA 9 KATEGÓRIÁRA VONATKOZÓ TANÍTÁS EREDMÉNYEI PONTOSSÁG- (BALRA) ÉS VESZTESÉGÉRTÉKEK SZERINT, RÓZSASZÍNNEL JELÖLVE A TANÍTÓHALMAZRA, ZÖLD SZÍNNEL JELÖLVE A VALIDÁCIÓS HALMAZRA VONATKOZÓ ÉRTÉKEK, AHOL A BATCHSIZE 8-AS ÉRTÉKRE VÁLASZTOTT	55
55. ÁBRA 9 KATEGÓRIÁRA VONATKOZÓ TANÍTÁS EREDMÉNYEI PONTOSSÁG- (BALRA) ÉS VESZTESÉGÉRTÉKEK SZERINT, KÉK SZÍNNEL JELÖLVE A TANÍTÓHALMAZRA, NARANCS SZÍNNEL JELÖLVE A VALIDÁCIÓS HALMAZRA VONATKOZÓ ÉRTÉKEK, A BATCHSIZE 16-OS ÉRTÉKŰ	55
56. ÁBRA ZENESTÍLUSOK, VALAMINT ANNAK VALÓSZÍNŰSÉGEIK KIÍRATÁSA EGY ADOTT ZENEHALMAZRA, AMELYEK ELÉRÉSI ÚTVONALA A FILES-BAN TALÁLHATÓ, NORMALIZÁLT ÉRTÉKEKKÉNT VIZSGÁLVA.....	55
57. ÁBRA ELSŐ STÍLUSKÉNT, MÁSODIK STÍLUSKÉNT, ILLETVE HARMADIK STÍLUSKÉNT MEGTALÁLT ZENESTÍLUSOK DARABSZÁMA, VALAMINT SZÁZALÉKOS ARÁNYA VGG16 ARCHITEKTÚRÁN VÉGZETT TANÍTÁS ALAPJÁN	56
58. ÁBRA A LEHETŐ LEGNAGYOBB KÉPMÉRETET FELDOLGOZÓ MODELL SZERKEZETE	57
59. ÁBRA 9 KATEGÓRIÁRA VONATKOZÓ TANÍTÁSI EREDMÉNYEK SAJÁT KÉSZÍTÉSŰ MODELL ÁLTAL, PONTOSSÁG- (BALRA) ÉS VESZTESÉGÉRTÉKEK SZERINT, NARANCSSZÍNNEL A TANÍTÓHALMAZRA, KÉK SZÍNNEL A VALIDÁCIÓS HALMAZRA VONATKOZÓAN	58

60. ÁBRA ELSŐ STÍLUSKÉNT, MÁSODIK STÍLUSKÉNT, ILLETVE HARMADIK STÍLUSKÉNT MEGTALÁLT ZENESTÍLUSOK DARABSZÁMA, VALAMINT SZÁZALÉKOS ARÁNYA SAJÁT MODELL SZERINT.....	58
61. ÁBRA NASNetMOBILE MODELLEL TÖRTÉNT TANÍTÁS EREDMÉNYE A TANÍTÁSI HALMAZRA VONATKOZÓAN	58
62. ÁBRA NASNetMOBILE MODELLEL TÖRTÉNT TANÍTÁS EREDMÉNYE A VALIDÁCIÓS HALMAZRA VONATKOZÓAN, EREDMÉNYEZVE EGY LEGJOBB 30% KÖRÜLI ÉRTÉKET VALIDÁCIÓS PONTOSSÁG SZERINT	59
63. ÁBRA A SPOTIFYOS ADATOKRA VONATKOZÓ MODELL SZERKEZETE, SZEREPELTETVE A TANÍTÓ PARAMÉTEREK SZÁMÁT	60
64. ÁBRA A KONFIGURÁCIÓS FÁJL SZERKEZETE, KIEMLelve A MEGFELELŐ ÉRTÉKET, AMELY A HELYES MŰKÖDÉSHEZ SZÜKSÉGES MÓDOSÍTANI	61
65. ÁBRA FELHASZNÁLÓ ÁLTAL MEGADOTT ZENÉRE TÖRTÉNŐ KERESÉSRE, ÉS ANNAK EREDMÉNYE.....	62
66. ÁBRA SPOTIFY ALKALMAZÁSON BELÜLI KERESÉS EREDMÉNYE UGYANAZON ZENÉRE. LÁTHATÓ, HOGY AZ ELSŐ ELŐADÓ NEVE EGYEZIK AZ ALKALMAZÁSON BELÜL MEGADOTT NÉVEL IS.	62
67. ÁBRA RÉSZLET EGY TESZTELÉSI HALMAZRA TÖRTÉNŐ PREDIKCIÓKRÓL KÉSZÍTETT LOGFÁJLBÓL.....	65
68. ÁBRA A PROGRAM ELINDÍTÁSÁT KÖVETŐ KEZDŐKÉPERNYŐ, KIVÁLASZTHATÓ AZ OFFLINE STÍLUSPREDIKTÁLÁS SPEKTROGRAMOKBÓL, VALAMINT ONLINE, SPOTIFY FELHASZNÁLÁSÁVAL.....	72
69. ÁBRA OFFLINE ZENEMEGHATÁROZÁS ABLAKÁNAK KEZDŐKÉPERNYŐJE.....	72
70. ÁBRA PÉLDA HIBÁSAN BEVITT FÁJLRA.....	73
71. ÁBRA PÉLDA HELYES FORMÁTUMÚ ZENEFÁJLRA, AMI NEM MEGFELELŐEN HOSSZÚ	73
72. ÁBRA PÉLDA HELYESEN BEVITT, KELLŐEN HOSSZÚ ZENEFÁJLRA, A „STÍLUST!” GOMB AKTIVÁLÓDÁSÁVAL	73
73. ÁBRA SIKERES KONVERTÁLÁS ESETÉN MEGJELENŐ ABLAK, AZ OK GOMB LENYOMÁSÁRA MEGKEZDŐDIK AZ ILLESZTÉS A MODELLRE	74
74. ÁBRA EREDMÉNYEK KIJELZÉSE A FOLYAMAT VÉGÉN A FELHASZNÁLÓ SZÁMÁRA.....	74
75. ÁBRA PÉLDA NEM MEGFELELŐ FORMÁTUMÚ FÁJLRA – AMENNYIBEN NEM TÖRTÉNIK BETALLÓZÁSRA FÁJL, ÚGY UGYANEZEN ABLAK FOGADJA A FELHASZNÁLÓT	75
76. ÁBRA PÉLDA BEÍRT, DE SPOTIFY-ON NEM LÉTEZŐ „ZENÉRE”	75
77. ÁBRA PÉLDA BEÍRT ZENÉRE, AMELY MEGTALÁLHATÓ SPOTIFY-ON	76
78. ÁBRA PÉLDA HELYESEN BETALLÓZOTT ZENÉRE. LÁTHATÓ, HOGY A FENTI MEZŐ MÉG AZ ELŐZŐ KERESÉSNEK MEGFELELŐ, DE AZ ADATOK LEJEBB FRISΣÜLTEK A LEGUTÓBBI TALLÓZOTT ZENÉNEK MEGFELELŐEN	76
79. ÁBRA FIGYELMEZTETŐ ABLAK SPOTIFY-ON MEG NEM TALÁLT ZENE ESETÉN	77
80. ÁBRA FIGYELMEZTETŐ ABLAK NEM MEGFELELŐEN BEÍRT METAADAT ESETÉN, AMELYBEN JAVASLATOT TESZ A FOLYTATÁSRA	77
81. ÁBRA KERESÉS UTÁNI ÁLLAPOT EREDETILEG NEM MEGFELELŐ METAADATTAL RENDELKEZŐ ZENE ESETÉN.....	78
82. ÁBRA EREDMÉNYEK KIJELZÉSE A FOLYAMAT VÉGÉN A FELHASZNÁLÓ SZÁMÁRA ONLINE ALAPÚ MEGHATÁROZÁS ESETÉN ..	78
83. ÁBRA PÉLDA SPOTIFY ADATOKON ALAPULÓ PREDIKCIÓRA INTERNETKAPCSOLAT HIÁNYÁBAN	79
1. TÁBLÁZAT AZ 1997-ES TANULMÁNY ÉS KUTATÁS EREDMÉNYE 4-, ILLETVE 8-OSZTÁLYOS KLASSZIFIKÁCIÓ ESETÉN [1]	5
2. TÁBLÁZAT A KÜLÖNBÖZŐ HMM ÉS AZ ETM-NN MÓDSZEREK FELISMERÉSI ARÁNYAI [3]	7
3. TÁBLÁZAT A FELISMERÉSEK KÖZTI KONFÚZIÓS MÁTRIX (AZAZ MEKKORA VALÓSZÍNŰSÉGGEL KÖVETKEZETT BE AZ, HOGY ADOTT STÍLUS HELETT EGY MÁSIK STÍLUS KERÜLT FELISMERÉSRE) [3]	7
4. TÁBLÁZAT STATISZTIKAI VIZSGÁLAT EREDMÉNYEI A 15 STÍLUSRA VONATKOZÓAN	35
5. TÁBLÁZAT KORRELÁCIÓANALÍZIS EREDMÉNYE A ZENÉK ADATAIRA VONATKOZÓAN	38
6. TÁBLÁZAT JAVÍTOTT KORRELÁCIÓANALÍZIS EREDMÉNYE A ZENÉK ADATAIRA VONATKOZÓAN	38
7. TÁBLÁZAT A ZENÉK SZÉTOSTZÁSA A METAADATBAN SZEREPLŐ STÍLUS MEGNEVEZÉS ALAPJÁN TÖRTÉNT – NÉHÁNY PÉLDA LÉTREJÖTT MAPPA ELNEVEZÉSEKBŐL.....	45
8. TÁBLÁZAT AZ ELSŐ LETÖLTÉSEKKEL BESZERZETT ZENÉK DARABSZÁMA, ÉS A SZÜKSÉGES 1000 DARABHOZ LÉVŐ HÁTRALÉVÓ MENNYISÉG. A NEGATÍV ÉRTÉKEK MÉG HIÁNYRA UTALNAK, A PIROS SZÍNNEL KIEMELT STÍLUSOK PEDIG AZ ÚJONNAN FELVETT STÍLUSOK.....	46
9. TÁBLÁZAT ADOTT ZENESTÍLUSOKHOZ KÉSZÜLT ÖSSZES SPEKTROGRAM, HIBÁS SPEKTROGRAMOK, ÉS VÉGLEGES SPEKTROGRAM MENNYISÉG, AMELY TANÍTÁSHOZ ÉS VALIDÁCIÓHOZ FELHASZNÁLHATÓ	48
10. TÁBLÁZAT METAADATBÓL TALÁLT ZENÉK MENNYISÉGE STÍLUSOK SZERINT, ÖSSZES ÖSSZEGYŰJTÖTT ZENE MENNYISÉGE, HIBÁS ZENÉK METAADAT ALAPJÁN, VALAMINT A HIÁNY AZ 1000 DARAB SZÜKSÉGES, STÍLUSONKÉNTI ZENÉHEZ	49
11. TÁBLÁZAT KORRELÁCIÓANALÍZIS A 9 ZENESTÍLUSRA VONATKOZÓAN	59

12. TÁBLÁZAT JAVÍTOTT KORRELÁCIÓANALÍZIS A 9 ZENESTÍLUSRA VONATKOZÓAN, KIVÉVE AZ ACOUSTICNESS, LOUDNESS ÉS INSTRUMENTALNESS ÉRTÉKEKET	59
13. TÁBLÁZAT EREDMÉNYÜL KAPOTT ÉRTÉKEK ELSŐ SZINTŰ, MÁSODIK SZINTŰ, HARMADIK SZINTŰ PONTOSÁGOKRA ADOTT MODELLEK ESETÉN A VALIDÁCIÓS HALMAZRA VONATKOZÓAN	60
14. TÁBLÁZAT ÖSSZEGYŰJTÖTT TESZTHALMAZ NAGYSÁGA DARABBAN KIFEJEZVE SPEKTROGRAMOKRA ÉS AZ ONLINE PREDIKCIÓRA VONATKOZÓAN.....	64
15. TÁBLÁZAT VGG16 ARCHITEKTÚRÁRA ÉPÜLŐ MODELL TESZTEREDMÉNYEI TOP1, TOP2, ILLETVE TOP3 SZINTEN ELTALÁLT STÍLUSOK SZERINT.....	65
16. TÁBLÁZAT A LEHETŐ LEGNAGYOBB KÉPMÉRETET MEGTARTÓ MODELL TESZTEREDMÉNYEI (VALIDÁCIÓS PONTOSÁG: 59,5%)	66
17. TÁBLÁZAT A LEHETŐ LEGNAGYOBB KÉPMÉRETET MEGTARTÓ MODELL TESZTEREDMÉNYEI (VALIDÁCIÓS PONTOSÁG: 57,5%)	66
18. TÁBLÁZAT VGG16 ALAPÚ MODELL TOP1 EREDMÉNYEINEK KONFÚZIÓS MÁTRIXA	67
19. TÁBLÁZAT ADAMAX OPTIMALIZÁLÓVAL RENDELKEZŐ MODELL TESZTEREDMÉNYEI TOP1, TOP2, ILLETVE TOP3 SZINTEN ELTALÁLT STÍLUSOK SZERINT.....	68
20. TÁBLÁZAT RMSPROP OPTIMALIZÁLÓVAL RENDELKEZŐ MODELL TESZTEREDMÉNYEI (TANÍTÁS VÉGI ÁLLAPOT)	69
21. TÁBLÁZAT RMSPROP OPTIMALIZÁLÓVAL RENDELKEZŐ MODELL TESZTEREDMÉNYEI (TANÍTÁS KÖZBENI, CHECKPOINT ÁLLAPOT).....	69
22. TÁBLÁZAT RMSPROP ALAPÚ MODELL TESZTEREDMÉNYEIRE VONATKOZÓ TOP1 EREDMÉNYEK KONFÚZIÓS MÁTRIXA	70

Irodalomjegyzék

- [1] R. B. Dannenberg, B. Thom, D. Watson: A Machine Learning Approach to Musical Style Recognition, 1997.
- [2] S. E. Fahlman, C. Lebiere: The cascade-correlation learning architecture. *Advances in neural information processing systems*, pp. 524-532., 1990..
- [3] H. Soltau, T. Schultz, M. Westphal, A. Waibel: Recognition of Music Types. *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, 1998., pp. 1137-1140.
- [4] H. Bahuleyan: Music genre classification using machine learning techniques. *arXiv preprint arXiv: 1804.01149*, 2018.
- [5] fogalomtar.hte.hu, Hírközlési és Informatikai Tudományos Egyesület.
(<https://www.fogalomtar.hte.hu/wiki/-/wiki/HTE+Infokommunikacios+Fogalomtar/Jel-zaj+viszony>), Hozzáférés dátuma: 2020. 05. 16.
- [6] Aszalós T.: Stílusismeret és technikaképzés. *Parlando: zenepedagógiai folyóirat*, 2009., pp. 13-21.
- [7] Zenei ENCIklopédia - Popzene (<https://www.zenci.hu/szocikk/popzene>), Hozzáférés dátuma: 2019. 05. 16.
- [8] Zenei ENCIklopédia - Rockzene (<https://www.zenci.hu/szocikk/rock-and-roll>), Hozzáférés dátuma: 2019. 05. 16.
- [9] C. Dahlhaus, H. H. Eggebrecht: Brockhaus Riemann zenei lexikon I-III., Budapest: Zeneműkiadó, 1983-1985.
- [10] Zenei ENCIklopédia - Rapzene (<https://www.zenci.hu/szocikk/rap>), Hozzáférés dátuma: 2019. 05. 16.
- [11] pulzar.hu (<https://www.pulzar.hu/zeneistilusok/house>), Hozzáférés dátuma: 2020. 05. 17.
- [12] R. Snoman: The Dance Music Manual: Tools, Toys, Techniques - Second Edition. Oxford, UK, Elsevier Press, 2009., pp. 251-253., 266.
- [13] Zenei ENCIklopédia - Jazz-zene (<https://www.zenci.hu/szocikk/jazz>), Hozzáférés dátuma: 2019. 05. 16.
- [14] Zenei ENCIklopédia - Elektronikus zene (<https://www.zenci.hu/szocikk/elektronikus-zene>), Hozzáférés dátuma: 2019. 05. 16.
- [15] R. J. Burgess: The History of Music Production. Oxford, 2014., pp. 136-146..
- [16] Zenei ENCIklopédia - Latin zene (<https://www.zenci.hu/szocikk/latin-amerikai>), Hozzáférés dátuma: 2019. 05. 16.

-
- [17] Hanula Zs., Ma Is Tanultam Valamit, Index.hu Zrt.
(https://index.hu/tudomany/til/2019/02/01/henry_ford_gyulolte_a_jazzt_ezert_lett_a_merika_nepzeneje_a_country/), Hozzáférés dátuma: 2020. 05. 14.
- [18] countrymusic.hu (<http://countrymusic.hu/country-tortenete>), Hozzáférés dátuma: 2020. 05. 14.
- [19] Horváth T., tancolo.pte.hu. (http://tancolo.pte.hu/hu/hip_hop_elmelet), Hozzáférés dátuma: 2020. 05. 14.
- [20] D. Coffey, encyclopedia.com. (<https://www.encyclopedia.com/media/encyclopedias-almanacs-transcripts-and-maps/new-age-music>), Hozzáférés dátuma: 2020. 05. 14.
- [21] recorder.blog.hu
(https://recorder.blog.hu/2014/08/13/minden_egyforman_tanczene_kis_edmologia), Hozzáférés dátuma: 2020. 05. 14.
- [22] rap.hu (<https://www.rap.hu/eletstilus/rnb/>), Hozzáférés dátuma: 2020. 05. 14.
- [23] techno.hu (<https://techno.hu/techno/>), Hozzáférés dátuma: 2020. 05. 14.
- [24] W. Drevo, willdrevo.com. (<https://willdrevo.com/fingerprinting-and-audio-recognition-with-python/>), Hozzáférés dátuma: 2019. 05. 16.
- [25] Tasnádi T., math.bme.hu. (<https://math.bme.hu/~reffyj/Fourier-transzformacio.pdf>), Hozzáférés dátuma: 2020. 05. 16.
- [26] Dr. Lipovszki Gy., regi.tankonyvtar.hu, Oktatási Hivatal.
(https://regi.tankonyvtar.hu/hu/tartalom/tamop412A/2010-0017_36_jelfeldolgozas_es_szamitogepes_iranyitas/ch04s05.html), Hozzáférés dátuma: 2020. 05. 17.
- [27] Kislexikon, Lapoda Multimédia. (http://www.kislexikon.hu/amplitudo_a.html), Hozzáférés dátuma: 2020. 05. 17.
- [28] A. L.-C. Wang: An Industrial Strength Audio Search Algorithm. *Ismir*, 2003., pp. 7-13.
- [29] C. Nicholson, pathmind.com. (<https://pathmind.com/wiki/neural-network>), Hozzáférés dátuma: 2020. 05. 15.
- [30] MathWorks, The MathWorks Inc.
(<https://www.mathworks.com/discovery/convolutional-neural-network.html>), Hozzáférés dátuma: 2020. 05. 10.
- [31] S. Saha, towardsdatascience.com. (<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>), Hozzáférés dátuma: 2020. 05. 10.
- [32] S. Amidi, A. Amidi, stanford.edu. (<https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks>), Hozzáférés dátuma: 2020. 05. 10.

-
- [33] image-net.org, UNC Vision Lab. (<http://image-net.org/challenges/LSVRC/2016/index>), Hozzáférés dátuma: 2020. 05. 10.
- [34] R. Thakur, towardsdatascience.com. (<https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c>), Hozzáférés dátuma: 2020. 05. 17.
- [35] developers.google.com, Google Developers. (https://developers.google.com/machine-learning/glossary#gradient_descent), Hozzáférés dátuma: 2020. 05. 17.
- [36] developers.google.com, Google Developers. (<https://developers.google.com/machine-learning/crash-course/reducing-loss/gradient-descent>), Hozzáférés dátuma: 2020. 05. 17.
- [37] developers.google.com, Google Developers. (<https://developers.google.com/machine-learning/crash-course/reducing-loss/stochastic-gradient-descent>), Hozzáférés dátuma: 2020. 05. 17.
- [38] D. P. Kingma, J. Ba: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [39] kaggle.com, Kaggle Inc. (<https://www.kaggle.com/c/human-protein-atlas-image-classification/discussion/70253>), Hozzáférés dátuma: 2020. 05. 10.
- [40] keras.io. (<https://keras.io/api/applications/>), Hozzáférés dátuma: 2020. 05. 10.
- [41] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, C. Liang-Chieh: Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018., pp. 4510-4520.
- [42] M. Sandler, ai.googleblog.com. (<https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html>), Hozzáférés dátuma: 2020. 05. 10.
- [43] B. Zoph, V. Vasudevan, J. Shlens, L. Quoc, ai.googleblog.com. (<https://ai.googleblog.com/2017/11/automl-for-large-scale-image.html>), Hozzáférés dátuma: 2020. 05. 17.
- [44] Ableton.com (<https://learningmusic.ableton.com/make-beats/beat-and-tempo.html>), Hozzáférés dátuma: 2020. 05. 16.
- [45] developer.spotify.com, Spotify AB. (<https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/>), Hozzáférés dátuma: 2019. 05. 16.
- [46] openmusictheory.com, Hybrid Pedagogy Publishing. ([http://openmusictheory.com/pitch\(Class\).html](http://openmusictheory.com/pitch(Class).html)), Hozzáférés dátuma: 2020. 05. 17.
- [47] Tóthné Parázsó L., regi.tankonyvtar.hu, Oktatási Hivatal. (https://regi.tankonyvtar.hu/hu/tartalom/tamop425/0005_31_kutatasmodszertan_sco_rm_05/536_korrelcianalzis.html), Hozzáférés dátuma: 2020. 05. 17.

-
- [48] Dr. Kóródi M., regi.tankonyvtar.hu, Oktatási Hivatal.
(https://regi.tankonyvtar.hu/hu/tartalom/tamop425/0051_Turizmus_kutatasok_modszertana/ch04s04.html), Hozzáférés dátuma: 2020. 05. 17.
- [49] ID3.org (<https://id3.org/id3v2.4.0-structure>), Hozzáférés dátuma: 2020. 05. 10.
- [50] mutagen.readthedocs.io (<https://mutagen.readthedocs.io/en/latest/user/id3.html>),
Hozzáférés dátuma: 2020. 05. 17.
- [51] [tensorflow.org](https://www.tensorflow.org). (https://www.tensorflow.org/tutorials/keras/overfit_and_underfit),
Hozzáférés dátuma: 2020. 05. 16.
- [52] J. Max, [soundmaximum.com](https://www.soundmaximum.com), SoundMaximum.
(<https://www.soundmaximum.com/music-genre-statistics/>), Hozzáférés dátuma: 2020. 05. 17.
- [53] F. Richter, [statista.com](https://www.statista.com), Statista Inc. (<https://www.statista.com/chart/15763/most-popular-music-genres-worldwide/>), Hozzáférés dátuma: 2020. 05. 17.
- [54] D. Silver, M. Lee, C. C. Childress: Genre complexes in popular music. *PloS one*, 11(5), e0155471, 2016.
- [55] L. Lu, Y. Shin, Y. Su, G. E. Karniadakis: Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733 (2019)*, 2019.
- [56] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov: Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, Vol. 15., 2014., pp. 1929-1958.