Klasifikacija glazbe po žanru
Projektni prijedlog

Šime Batović Mislav Vučković
Andrija Mandić Marko Jukić

18. lipnja 2020.

Sažetak

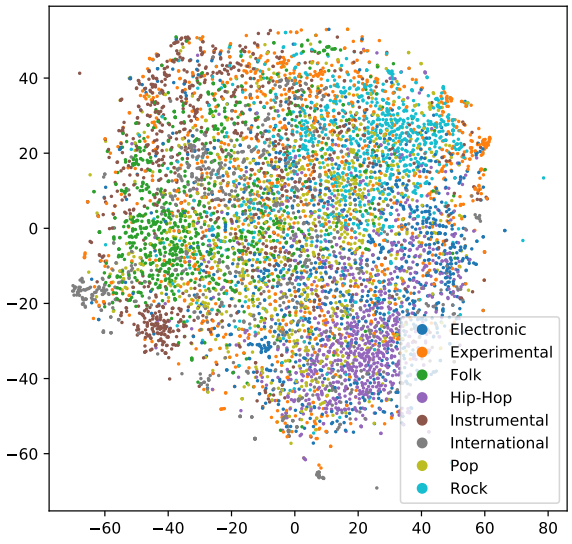
Analiza pjesama i njihovo prepoznavanje su često istraživani problemi. U ovom radu bavit ćemo se problemom klasifikacije glazbe po žanru. Usporedit ćemo klasične modele problema klasifikacije u strojnom učenju s metodom dubokog učenja koristeći konvolucijske neuronske mreže. Cilj je provjeriti već dokazane metode u radu s klasifikacijom zvuka te analizom i uvođenjem novih značajki poboljšati krajnju preciznost predviđanja. Istražili smo kakve rezultate daju modeli: stabla odlučivanja, slučajnih šuma, lagističke regresije, metode potpornih vektora, AdaBoosta i XGBoosta. Koristili smo FMA dataset koji sadrži 8000 isječaka pjesama koje su ravnomjerno raspoređene u 8 glavnih žanrova. Krajnji rezultati su u skladu s očekivanima, dobili smo točnost malo manju od 60% gdje su konvolucijske neuronske mreže nadmašile metode klasičnog strojnog učenja.

1 Uvod (motivacija i ciljevi)

Ljudi su oduvijek imali potrebu za stvaranjem i slušanjem glazbe. Glazba je sveprisutna te gotovo svi imaju neki žanr koji im najviše odgovara i koji najčešće slušaju. U današnje vrijeme postoji puno aplikacija i programa koji korisniku pružaju preporučen (recommenden) sadržaj. Tako Spotify, najpoznatija svjetska platforma za slušanje glazbe, „teška" preko 26 milijardi američkih dolara, uvelike koristi klasifikaciju glazbe po žanru. Također, kako reprezentacija glazbe na računalu nije trivijalna te je to nešto s čime se većina nas dosad nije susretala, ovaj je problem ipak nešto složeniji što nas je navelo da istražimo te odaberemo ovu temu za projekt. Cilj našeg istraživanja bio je usporediti klasične modele problema klasifikacije u strojnom učenju s metodom dubokog učenja koristeći konvolucijske neuronske mreže. Ideja je provjeriti već dokazane metode u radu s klasifikacijom zvukate analizom i promjenom/uvodenjem novih značajki poboljšati krajnju preciznost predviđanja.

2 Opis problema (skup podataka)

Za skup podataka odabrali smo FMA dataset [3], konkretnije `fma_small`, koji se sastoji od 8000 isječaka pjesama duljine 30 sekundi u mp3 formatu, po 1000 pjesama za svaki od 8 žanrova. Dataset je poprilično velik, više od $8,5$ GB, što nam je predstavljalo neke tehničke probleme. Kako je skup podataka balanciran, odlučili smo koristiti točnost za usporedbu modela. Uz svaku pjesmu dani su i brojni metapodatci, od kojih nam je potreban samo žanr jer on predstavlja fokus našeg istraživanja. Iako je žanr dan kao hijerarhijski, mi smo zbog jednostavnosti i veličine skupa podakata pokušali predvidjeti samo jedan od osam glavnih žanrova: Experimental, Hip-Hop, Rock, Pop, Folk, Electronic, Instrumental i International. Kako ovaj skup podataka za svaki žanr sadrži velik broj podžanrova, od kojih neki mogu smatrani potpuno drugim žanrom, pjesme koje pripadaju istom skupu se mogu uvelike razlikovati.



Slika 1: Vizualizacija značajki pomoću t-SNE

3 Opis metode i pristupa za rješavanje problema

Klasifikaciji glazbe po žanru pristupili smo na dva načina. Prvi pristup je bio izračunati razne spektralne i ritamske značajke za svaku pjesmu i pomoću tih značajki i klasičnih metoda strojnog učenja pokušati odrediti žanr pjesme. Drugi pristup je bio reprezentirati pjesme sa mel-spektogramima i iskoristiti snagu konvolucijskih neuronskih mreža na klasifikaciji slika da bismo klasificirali pjesme pomoću mel-spektogramima. Sve smo radili u Pythonu, a modele trenirali na *Kaggle*-u, koristeći grafičke procesore za ubrzanje gdje god je to bilo moguće. Kaggle nudi mogućnost rada sa Tesla P100 grafičkim procesorom sa 16GB memorije.

3.1 Klasične metode

Koristili smo dvije vrste značajki za klasifikaciju: spektralne i ritamske. Neke od spektralnih značajki su *spectral centroid*, *spectral centroid* i *spectral bandwidth*, a od ritamskih značajki smo koristili tempo. Neke od značajki su skalarne, dok ostale ovise o vremenu. Za one koje ovise o vremenu smo odredili distribucije i izračunali ukupno 7 mjera: minimum, maksimum, aritmetičku sredinu, standardnu devijaciju, medijan, koeficijent asimetrije i koeficijent zaobljenosti. Na taj način dobili smo ukupno 380 značajki za svaku pjesmu. Za računanje svih značajki koristili smo Python paket *LibROSA*. Popis svih značajki i ostali detalji nalaze se na GitHub repozitoriju.

Skup značajki podijelili smo na train (80%) i test (20%). Za svaki model na train skupu isprobali smo veliki broj parametara pomoću `GridSearchCV` i za model s najboljim parametrima odredili točnost na testnim podacima. Kako nam je u istraživanju cilj ispitati točnost različitih modela, za kriterijsku funkciju za odabir parametara smo odabrali upravo točnost. Nakon pronalaska najboljeg skupa parametara, model je ponovno istreniran na svim train podacima.

Istražili smo šest klasičnih modela strojnog učenja:

- Stablo odlučivanja
- Slučajna šuma
- Logistička regresija
- Metoda potpornih vektora (SVM)
- AdaBoost
- XGBoost

Za sve modele osim XGBoost koristili smo implementacije iz Python paketa `scikit-learn`, a za XGBoost implementaciju iz paketa `xgboost`.

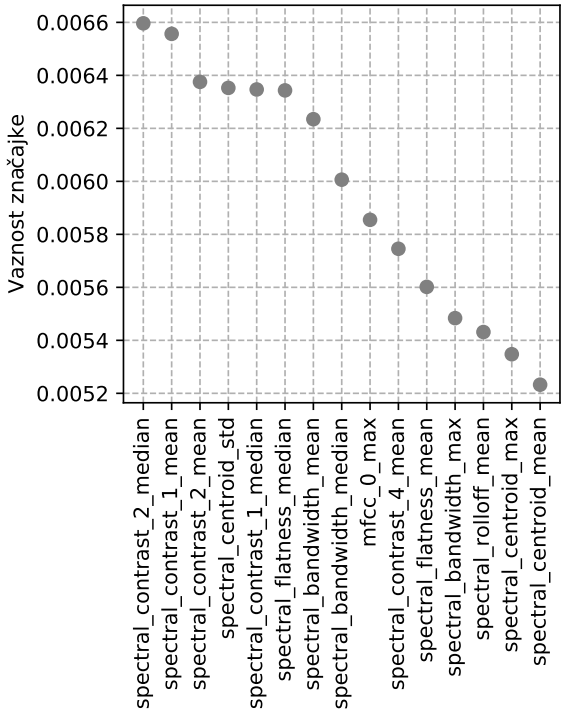
3.1.1 Stablo odlučivanja

Pomoću `GridSearchCV` pretražili smo parametre za `criterion` funkciju koja mjeru kvalitetu razdvajanja u stablima. Dva moguća parametra koja se pretražuju su `gini` koja mjeri uspješnost na temelju Gini indeksa, te `entropy` koja mjeri porast informacije. Drugi parametar koji `GridSearchCV` izabire je strategija za odabir splita na svakom čvoru. Podržane strategije su `random` i `best` koja odabire najbolji split. Najbolja kombinacija parametara daje točnost od 34.75% na testnim primjerima.

3.1.2 Slučajna šuma

Algoritam slučajne šume pokazao se puno uspješnijim nego Stabla odlučivanja. Uz neograničenu maksimalnu dubinu stabla, funkciju

razdvajanja na temelju Gini indexa te 1000 stabala u šumi, uspjeli smo postići točnost od 53.25% na test primjerima. Pomoću dobivenog modela određene su važnosti značajki:



Slika 2: Važnost najvažnijih 15 značajki uz pomoć slučajne šume

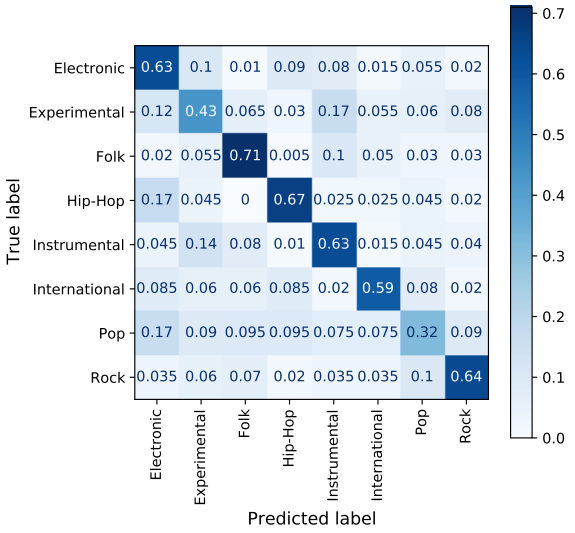
Značajke koje se nalaze bliže korijenu stabla imaju veći utjecaj na predikciju i prema tome se mogu smatrati važnijima. Na slici 2 vidimo da su najvažnije značajke nastale od *spectral contrast*, *spectral centroid* i *spectral bandwidth*.

3.1.3 Logistička regresija

Za odabir značajki koristimo prethodno izračunat model slučajne šume i funkciju `SelectFromModel` koja za *threshold* vrijednost uzima medijan vrijednosti značajki. Koristimo Pipeline koji kao transformer uzima `StandardScaler` koji od svake značajke oduzima srednju vrijednost te ju skalira, te kao estimator model logističke regresije s maksimalnim brojem iteracija postavljenim na 1000. Model je postigao točnost od 52.25% na test primjerima.

3.1.4 Metoda potpornih vektora

Za odabir značajki također koristimo slučajnu šumu. Metoda je uglavnom ista kao i kod logističke regresije, koristimo još RBF kernel za mapiranje u veće dimenzije te malo veći regularizacijski parametar koji žrtvuje veličinu margine za što bolju klasifikacijsku točnost. Dobiveni rezultati su dosad najbolji: 57.63% na testnim primjerima. Matrica konfuzije za testne primjere dana je na idućoj slici.



Slika 3: Matrica konfuzije za SVM

Vidimo da model jako loše prepoznaje žanrove Pop i Experimental, a najlakše prepoznaje žanrove Folk i Hop-Hop.

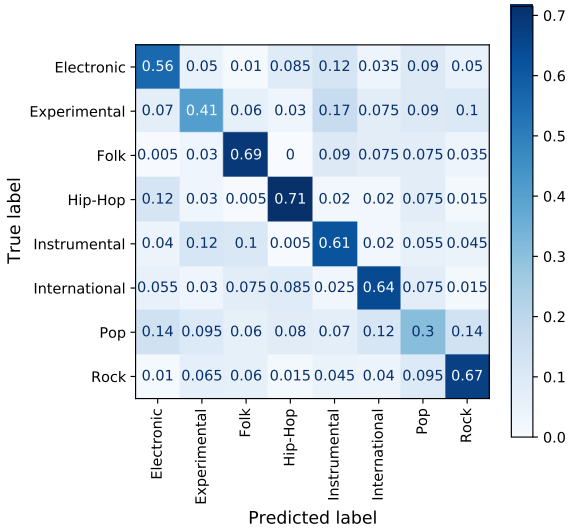
3.1.5 AdaBoost

Prva boosting tehnika koju smo primjenili je AdaBoost. Boosting algoritmi primarno smanjuju pristranost (bias), ali su također dobri i za smanjivanje varijance. AdaBoost algoritmu smo povećali broj estimatora na 180, ali smanjili `learning_rate` koji određuje koliko svaki novi model pridonosi postojećem. Rezultati su loši u usporedbi s drugim metodama poput SVM-a, 46.31% na testnim primjerima.

3.1.6 XGBoost

XGBoost pokazao se boljim nego AdaBoost, prva boosting tehnika koju smo koristili. To

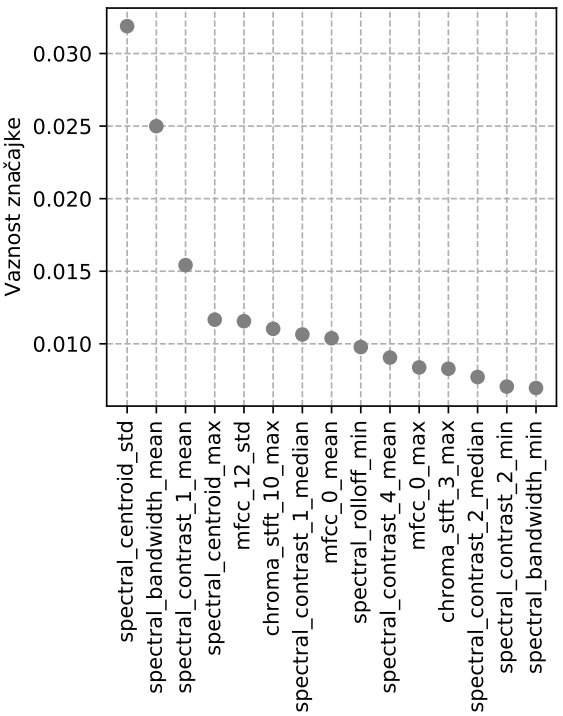
u velikoj mjeri možemo zahvaliti njegovoj robustnosti na outliere, s obzirom da je naš skup podataka poprilično neuredan s mnogo outliera. XGBoost je tako i kod nas pokazao najbolje rezultate kao i u većini stvarnih slučajeva. Za konstrukciju stabala koristili smo metodu `gpu_hist` koja koristi grafički procesor za ubrzano treniranje. Ispostavilo se da postavljanje broja stabala na 180 i `learning_rate` na 0.25 daje najbolju točnost: 57.50% na testnim primjerima. Dobivena matrica konfuzije dana je na slici 4.



Slika 4: Matrica konfuzije za XGBoost

Vidimo da je matrica konfuzije za XGBoost jako slična matrici konfuzije za SVM. Možemo primijetiti da XGBoost bolje prepoznaje žanrove Electronic i International od SVM.

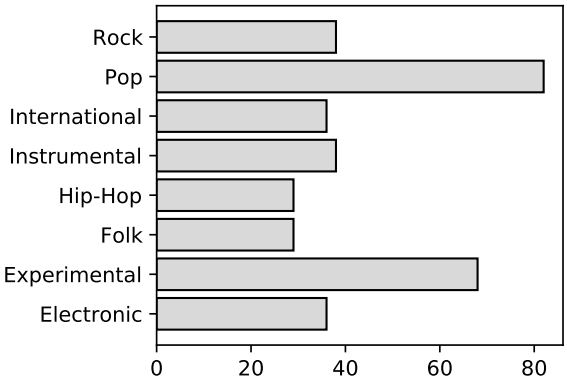
Dobili smo točnost od 57.50% na testnim podacima, što je slično rezultatima SVM-a. Kao i za slučajnu šumu, izračunali smo najvažnije značajke (na slici 5). Vidimo da su, kao kod slučajne šume, *spectral contrast*, *spectral centroid* i *spectral bandwidth* među najvažnijima. Primjećujemo da su i MFC koeficijenti među najvažnijim značajkama.



Slika 5: Važnost najvažnijih 15 značajki uz pomoć XGBoost

3.1.7 Analiza modela

Klasičnim metodama strojnog učenja dobili smo uglavnom očekivane razultate od kojih su mnogi bili relativno slični. Jedan od glavnih razloga za to je dataset s kojim radimo. Mnoge pjesme su toliko različite od ostalih u svom žanru da ih niti jedan klasifikator nije uspio prepoznati.



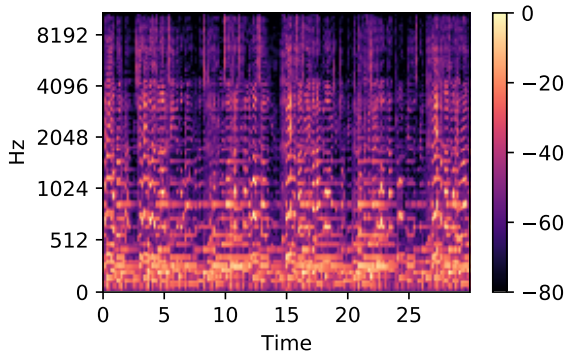
Slika 6: Broj pjesama po žanrovima koje su svi klasični modeli krivo klasificirali

Na slici 6 vidimo da je broj pjesama koje

su svi modeli krivo klasificirali jako velik, što znači da su svi modeli imali problema s klasifikacijom određenog skupa podataka. Najviše je pjesma iz žanra „pop" i „experimental", što je bilo vidljivo i iz matrica konfuzija koje smo dobivali. Ostali žanrovi su uglavnom poravnati što je možda bilo i za očekivati jer su „pop" i „experimental" žanrovi koji u današnje vrijeme obuhvaćaju jako širok spektar pjesama.

3.2 Konvolucijske neuronske mreže - CNN

Za prikaz pjesama u Pythonu koristili smo biblioteku *LibROSA* koja služi za reprezentaciju i analizu zvuka. Za opisivanje svake pjesme koristili smo mel-spektogram, dvodimenzionalni graf koji prikazuje jačinu frekvencija u ovisnosti o vremenu.

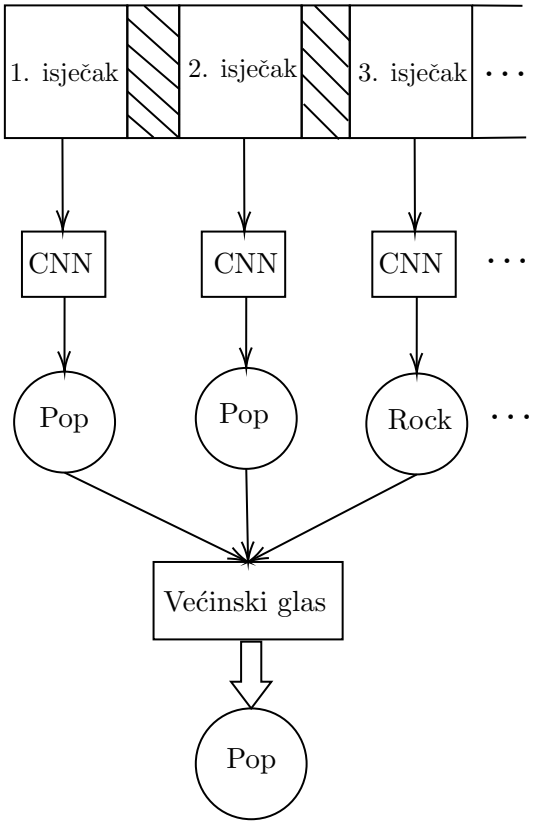


Slika 7: Mel-spektogram pjesme žanra Folk

Mel-spektogram sadrži mel-skalu na y-osi koja se dobije nelinearnim transformacijama frekvencije zvuka. Prikaz jednog spektograma se može vidjeti na slici 7. Takav mel-spektogram, dobiven koristeći funkcije biblioteke *LibROSA*, u Pythonu je reprezentiran kao dvodimenzionalno polje realnih brojeva. Zbog veličine dataseta i uštede vremena, generiranje spektograma napravljeno je odvojeno i rezultati su spremljeni.

3.2.1 Metoda

Originalni mel-spektogrami većinom su bili dimenzija 128×1291, gdje ih je par bilo malo dužih po vremenu (pjesme su se možda razlikovale u mili sekundama). Tako učitane spektograme podijelili smo na više dijelova te tako dobili manje slike dimenzija 128×16 na kojima smo učili našu neuronsku mrežu. Na taj način dobili smo veći skup podataka za učenje neuronske mreže, s time da smo dobivene isječke iz određene pjesme nekog žanra označili tako da pripada tom žanru.



Slika 8: Metoda većinskog glasa

Konačni model će biti *metoda većinskog glasa*. Klasifikacija se sastoji u tome da pjesmu koja je dimenzija 128×1291 podijelimo na dijelove dimenzija 128×16, svaki od tih dijelova „glasa" za koji je žanr te tu pjesmu svrstamo u neki žanr na način da gledamo koji se od tih „glasova" najviše pojavljivao. Ideja je prikazana lijevo na slici 8.

Kako u treniranju neuronske mreže moramo prilagoditi velik skup parametara, opisano povećanje broja primjeraka za učenje može pridonijeti kvalitetnijem odrađivanju te zadaće.

Ideju za arhitekturu konvolucijske neuronske mreže dobili smo proučavanjem radova koji se bave sličnim problemima analize zvučnih za-

pisa. Kod prepoznavanja uzoraka na slikama dobrim se pokazalo višestruko nizanje konvolucijskog sloja zajedno s *Max-Pooling* slojem. Konvolucijski sloj zadužen je za prepoznavanje raznih uzoraka, a *Max-Pooling* služi za redukciju dimenzije te pridonosi detekciji prostorne invarijantnosti. Na kraju slijedi potpuno povezani sloj (*fully connected layer*) koji je zaslužan za klasifikaciju.

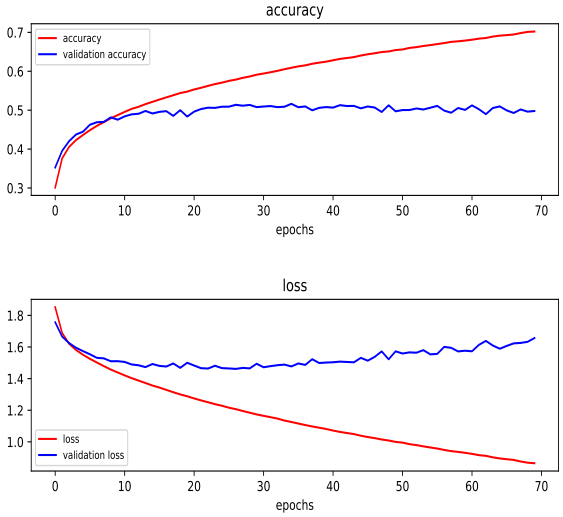
Ulazni sloj dobiven dijeljenjem spektograma predstavlja slika dimenzija 128×16. Kod konvolucijskih slojeva varirali smo broj filtera čiji se koeficijenti uče treniranjem. Potrebno je odrediti još hiperparametre pomaka filtera i veličinu proširenja nulama. Pomoću njih kontroliramo i veličinu izlaza, za koju želimo da ostane nepromijenjena nakon izlaza iz ovog sloja. Koristit ćemo u svim modelima filtere veličine 5×5 s pomakom filtera 1 te jediničnо dopunjavanje nulama sa svake strane slike. Veličinu filtera u sloju sažimanja izborom maksimalnog elementa postavljamo na 2×2, s pomakom 2. Pri izboru najbolje arhitekture mreže variramo još i broj potpuno povezanih slojeva zajedno s brojem neurona u njima. U takvim slojevima korištena je *ReLU* aktivacijska funkcija. Na kraju uvijek slijedi potpuno povezani sloj sa *Softmax* aktivacijskom funkcijom koji predstavlja konačnu klasifikaciju po žanrovima.

broj modela	validation accuracy
6	0.51625
11	0.5144922
5	0.5137109
3	0.51310545
2	0.511875

Tablica 1: Točnosti pet najboljih modela

Naše podatke podijelili smo u tri skupine: *train* 72%, *validation* 8% i test 20%. Prilikom učenja smo sveukupno iskoristili 11 CNN arhitektura koje se mogu vidjeti ovdje [1]. Prilikom učenja koristili smo *Stohastic gradient descent* s parametrima $learning\_rate = 0.001$ i $momentum = 0.9$ te je tokom učenja korišten $batch\_size = 256$. Od tih 11 arhitektura odabrali smo onu koja ima najbolji *validation accuracy* tokom učenja i na kraju procijenili koliko dobro taj model klasificira, pomoću metode većinskog glasa, na testnom skupu.

Iz priloženog vidimo da je najbolji model na temelju *validation accuracy*ja bio model broj 6. U modelu broj 6 imali smo jedan konvolucijski sloj s *ReLU* aktivacijom sa 128 filtera i *Max-Poolingom* te jedan *Dense layer* koji je imao 32 neurona s *ReLU* aktivacijom. Na kraju slijedi output *Dense layer* s 8 neurona sa *Softmax* aktivacijom jer smo imali 8 žanrova za klasificirati. Opisani prolazak isječka kroz neuronsku mrežu prikazan je na slici 9.



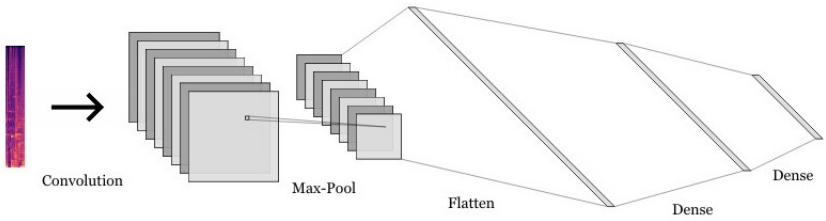
Slika 10: Krivulja učenja šestog modela

Iz slike 10 vidimo da se točnost na *training* skupu povećava s brojem epoha, dok validacija stagnira tokom epoha i dobivamo overfitting, koji može biti i veći ako povećamo broj epoha.

Na kraju, s tako dobivenim modelom, kada se ide procijeniti točnost testnog skupa metodom većinskog glasa dobili smo sveukupnu točnost od 58.69%, što je značajno poboljšanje od rezultata dobivenih samo na isječcima pjesama.

4 Prikaz rezultata

Svi dobiveni rezultati su zapisani u Tablici 2. Točnost je mjerena na testnim primjerima.



Slika 9: Arhitektura mreže devetog modela

Model	Točnost
Stablo odlučivanja	34.75%
Slučajna šuma	53.25%
Logistička regresija	52.25%
Metoda potpornih vektora	57.63%
AdaBoost	46.31%
XGBoost	57.50%
CNN	58.69%

Tablica 2: Točnosti modela na testnim podacima

5 Osvrt na druge pristupe

Klasifikacija glazbe po žanru je već duboko istražena tema pa postoji velik broj provedenih istraživanja. Rezultati uvelike ovise o datasetu s kojim se radi, u kojem su formatu pjesme, kolike su duljine te koliko su slične pjesme koje pripadaju istom žanru. S obzirom na to, najviše smisla ima gledati ona istraživanja koja su koristila isti `fma` dataset.

U istraživanju [5] korištena je proširena verzija našeg dataseta, koja sadrži ukupno 106,574 pjesama iz 16 žanrova. Također su koristili biblioteku `libROSA` za izvlačenje značajki. Koristili su slične modele kao mi. Koristeći softmax regresiju, koja je generlizacija logističke regresije, postigli su najlošiju točnost od 54,61%. Logističkom regresijom postigli su točnost od 64.75%. Za SVM model su koristili dvije verzije, s linearnim te RBG kernelom kakav smo i mi koristili. U verziji s linearnim dobili su točnost od 64.55% dok su s RBF kernelom postigli najbolju točnost u svom istraživanju, 68.07%. Na kraju su koristili CNN s 2 skrivena sloja, s 320 te 32 čvora, oba sa sigmoidalnom aktivacijskom funkcijom. *Output layer-i* koristili su *Softmax* aktivacijsku funkciju. Konačna postignuta točnost bila je 66.03%, što je lošije nego kod SVM-a s RBF kernelom. S obzirom na veličinu njihovog dataseta, vidimo da su naši rezultati dosta slični, možda čak i bolji kad usporedimo pristup sa CNN.

U zadnjih nekoliko godina, pojavilo se mnogo CNN modela koji poboljšavaju točnost kod prepoznavanja slika. Tim s pekinškog sveučilišta je u svom istraživanju [6] koristio također metodu većinskog glasa kao što smo mi u našem istraživanju. Oni su koristili također FMA-small dataset, s time da su oni svoje slike smanjili na dimenziju 128×128 te još su proveli *data augmentation*. *Data augmentation* je tehnika izbjegavanja *overfittinga* povečavanjem volumena dataseta, u ovom slučaju tehnikama *Time overlappinga*, gdje se svaki isječak preklapa 50% sa svojim susjednim isječkom, i *Pitch shiftinga* na izračunatim slikama spektograma. Nadalje, njihova arhitektura CNN-a se znatno razlikovala od naše. Mi smo radili s *Conv2D layerima* dok su oni radili s *Conv1D layerima*, gdje je glavna razlika da se koristi filter na poljima, a ne na dvodimenzionalnim matricama slika. Također, koristili su takvih 5 *Conv1D layera*, što je više nego u našem istraživanju. Na kraju, njihova metoda većinskog glasa je imala točnost od 59.4%, što pokazuje da smo s nešto jednostavnijim modelom dobili skoru istu točnost kao u tom radu. Također su kao model korisitili rezidualne neuronske mreže odnosno *ResNet*, razvijen 2015. godine od strane Microsoftovog istraživačkog tima, koji koristi veliki broj skrivenih slojeva koji koriste takozvane „prečice". Ulaz rezidualnog bloka se grana gdje se jedna grana sastoji od nekoliko konvolucijskih slojeva dok je druga grana prečica do kraja rezidualnog bloka gdje se dvije grane spajaju prije primjene nelinearnosti. Izlazi dviju

grana se jednostavno zbroje i na takvu sumu se primjenjuje nelinearnost, čime se dobije izlaz rezidualnog bloka. Primarni cilj ove mreže je rješavanje problema semantičke segragacije. Koristeći takve rezidualne mreže, skupa s kernelom veličine 4 i SVM-om kao *stacking classifier* umjesto metode većinskog glasa, postigli su točnost od 66.3%.

6 Mogući budući nastavak istraživanja

Prostor za daljnji napredak korištenjem konvolucijskih neuronskih mreža je velik. Vjerujemo da istraživanje novih arhitektura mreže u budućnosti može značajno pridonijeti analizi zvučnih zapisa. U našem slučaju, zbog hardverskih ograničenja koristili smo najmanju verziju *dataseta* te smo bili ograničeni kod složenosti konvolucijske mreže i njenog treniranja. Idući korak k većoj preciznosti bio bi treniranje na većem skupu podataka, što bi moglo značiti i bolje rezultate jer se povećanjem složenosti mreže uvelike povećava i broj parametara koje treba naučiti. Korigiranjem *overlapa* kod rezanja slika opisanog na slici 8 ili korištenje *data augmentation* radi povećanja volumena dataseta i smanjivanja overfitting-a kao u [6] bi također moglo utjecati na krajnji rezultat. Napomenimo još da korišteni skup podataka nudi i dodatne informacije o pjesmama koje nismo koristili. Konkretno, mogli bismo koristiti podatke o podžanrovima pjesma, učiti mrežu takvu kvalifikaciju koja bi možda onda donijela veću točnost u predviđanju glavnog žanra.

Literatura

[1] Odabir CNN-a. `https://github.com/mik1997/smam_strojno/blob/master/cnn_modeli.ipynb`.

[2] Strojno učenje. `https://web.math.pmf.unizg.hr/nastava/su/`.

[3] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. FMA: A dataset for music analysis. In *18th International Society for Music Information Retrieval Conference*, 2017.

[4] Daniel Grzywczak and Grzegorz Gwardys. Deep image features in music information retrieval. volume 60, pages 187–199, 08 2014.

[5] Tianchi Liu Li Guo, Zhiwei Gu. Music genre classification via machine learning.

[6] Bojin Zhuang Jiankui Yang Shaojun Wang Jing Xiao Wenhao Bian, Jie Wang. Audio-based music classification with densenet and data augmentation.