Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Vitja Klun

Detekcija harmonike v posnetkih ljudske pesmi

diplomsko delo

visokošolski strokovni študijski program prve stopnje računalništvo in informatika

Mentor: doc. dr. Matija Marolt

Ljubljana 2012

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00328/2012
Datum: 03.09.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **VITJA KLUN**

Naslov: **DETEKCIJA HARMONIKE V POSNETKIH LJUDSKE PESMI**
ACCORDION DETECTION IN FOLK MUSIC RECORDINGS

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

V diplomskem delu razvijte algoritem, ki v posnetkih slovenske ljudske glasbe detektira igranje na harmoniko. Pri tem ustvarite učno in testno množice pesmi z in brez harmonike, preučite katere značilke so najbolj primerne in izberite algoritem strojnega učenja, ki bo na bazi značilk opravljal detekcijo igranja na harmoniko.

Mentor:

doc. dr. Matija Marolt

Fakulteta za računalništvo in informatiko
2
UNIVERZA V LJUBLJANI

Dekan:

prof. dr. Nikolaj Zimic

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Vitja Klun, z vpisno številko **63080252**, sem avtor diplomskega dela z naslovom:

Detekcija harmonike v posnetkih ljudske pesmi

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Matije Marolta,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 18. septembra 2012 Podpis avtorja:

Iskreno se zahvaljujem mentorju doc. dr. Matiji Maroltu za vso pomoč in napotke pri izdelavi diplomskega dela.

Zahvaljujem se tudi materi Danieli in očetu Dariju, ker sta me ves čas študija podpirala in verjela vame.

Kazalo

Povzetek

V diplomskem delu smo implementirali algoritem, ki v posnetkih slovenske ljudske glasbe prepozna inštrument, v našem primeru harmoniko. Algoritem kot vhodni argument sprejme poljubno dolg posnetek oblike wave in ga po 3-sekundnih odsekih klasificira v enega izmed dveh skupin – vsebuje harmoniko ali ne vsebuje harmonike.

Učno bazo za strojno učenje smo zgradili na podlagi baze terenskih posnetkov slovenske ljudske glasbe. Iz dolgih posnetkov smo izrezali 4680 posnetkov dolgih 3 sekunde, med katerimi je bilo 2340 takšnih, ki vsebujejo harmoniko in 2340 posnetkov brez harmonike. Odločili smo se, da prvo tretjino baze namenimo učenju, drugo tretjino testiranju in tretjo tretjino izračunu pomembnosti značilnosti zvoka. Na vseh posnetkih smo izračunali značilnosti zvoka in raziskali, katere značilnosti najbolj vplivajo na klasifikacijo. Odvečne značilnosti zvoka smo zanemarili in s tem povečali hitrost ter natančnost klasifikacije. Z metodo podpornih vektorjev smo s pomočjo učne baze klasifikator naučili in opravili testiranje z našo testno bazo posnetkov. Rezultat testiranja je bila 95,83% natančnost klasifikatorja, ob koncu pa smo algoritem implementirali v okolju MATLAB.

Ključne besede:

Avtomatsko prepoznavanje glasbenih inštrumentov, klasifikacija, strojno učenje, računanje značilnosti zvoka

Abstract

In the thesis, we implemented an algorithm that automatically recognizes an instrument in Slovene Folk music, in our case the accordion. The input argument of our algorithm is an arbitrarily long wave form recording. Its 3-seconds long sections are then classified into one of two groups – contains the accordion or does not contain the accordion.

We have built the learning base for machine learning on the basis of database containing field recordings of Slovene Folk music. We cut 4680 3-seconds long sections from long recordings, 2340 of those incorporated the accordion and 2340 did not. We decided to devote the first third of the base to learning, the second third of the base to testing and the last third of the base to calculating the relevance of the characteristics of sound. We have calculated the characteristics of sound in all recordings and then researched which features of sound affect the classification the most. We have neglected redundant features od sound and thus increased the speed and accuracy of classification. We have taught the classifier with the method of support vectors and the help of learning database and then tested it with our test database of recordings. The testing showed the classifier to be 95,83% accurate. At the end, we implemented the algorithm in MATLAB environment.

Keywords:

Automatic music instrument recognition, classification, machine learning, audio feature extraction

Poglavje 1

Uvod

Avtomatsko prepoznavanje glasbenih inštrumentov v glasbenih posnetkih je še vedno eden največjih problemov na področju pridobivanja informacij iz glasbe. Na tem področju lahko govorimo o prepoznavanju inštrumentov v monofoničnih ali polifoničnih glasbenih posnetkih. Monofonična glasba vsebuje samo en inštrument, sočasno je zaigrana le ena nota, polifonična glasba pa vsebuje več inštrumentov hkrati in sočasno je lahko zaigranih več not. Prepoznavanje inštrumentov v polifonični glasbi je zato veliko bolj kompleksno, ker sočasno igranje več inštrumentov povzroča prekrivanje frekvenčnih komponent. Eden izmed pristopov reševanja problema prepoznavanja posameznega inštrumenta v polifonični glasbi je ločevanje želenega inštrumenta od posnetka, katerega opisujejo Heittola, Klapuri in Virtanen [7]. Tak pristop pri reševanju problema prepoznavanja se ne izkaže vedno kot veliko boljši in lahko kvečjemu poveča kompleksnost algoritma.

V naši diplomski nalogi smo se ukvarjali s prepoznavanjem inštrumenta v polifonični glasbi brez ločevanja posnetkov, pri čemer smo se osredotočili samo na slovensko ljudsko glasbo in s tem posplošili problem. Namen dela je bil implementirati algoritem, ki bi znal za poljubno dolg posnetek slovenske ljudske glasbe ugotoviti, na katerih delih tega posnetka se nahaja inštrument harmonika. Reševanje tega problema je zelo zanimivo, ker na nek način poskušamo računalniško prepoznavanje inštrumenta približati človeškemu pre-

poznavanju.

Slišno območje človeškega ušesa je med 20 Hz in 20000 Hz, to območje pa se spreminja glede na starost ali deformacijo. Ko zvočni valovi pripotujejo v uho, zanihajo bobnič, se prenesejo na slušne koščice, prek teh pa potujejo do polža. V polžu se nahajajo slušne čutnice z dlačnicami, ki se s pomočjo pretakanja tekočine v polžu vzdražijo. Dražljaji se prenesejo na čutilna živčna vlakna in potujejo po možganskem živcu do središča za sluh v možganih.

V možganih se vrši slušno grupiranje frekvenc prispelega zvoka ter analiza lastnosti zvoka. Ta zvok se primerja z zvoki v našem ti. leksikonu, kjer se opravi končno prepoznavanje. Če je zvok v leksikonu, torej nam je že znan, je prepoznavanje končano. Če zvoka v našem leksikonu ni, mu dodelimo nov smisel in pomen. Podobno delujejo tudi računalniški algoritmi za prepoznavanje inštrumentov. Na primerih v učni bazi se izračunajo določene zvočne značilnosti, potem pa se klasifikator s pomočjo določene metode nauči, kakšne vrednosti značilnosti ima tipično nek inštrument. Ko klasifikatorju podamo nov zvočni posnetek, lahko ta z določeno verjetnostjo na podlagi izračunanih značilnosti novega zvoka napove, ali le-ta vsebuje določen inštrument.

Namen implementiranega algoritma je hitro iskanje posnetkov slovenske ljudske glasbe (in delov v njih), ki vsebujejo inštrument harmonika. Algoritem je uporaben za iskanje v bazi, ki vsebuje veliko število posnetkov slovenske ljudske glasbe in onemogoča odkrivanje posnetkov, ki vsebujejo harmoniko zgolj z poslušanjem teh posnetkov. S tem bistveno pripomoremo k časovni ekonomičnosti iskanja po obsežnih bazah.

Poglavje 2

Orodja in metode

Za reševanje našega problema in implementacijo algoritma smo večinoma uporabljali štiri orodja : EtnoMuza, MATLAB, Timbre Toolbox, ki je dodatek k orodju MATLAB, ter orodje Orange.

Z orodjem EtnoMuza [2] smo si pomagali pri pripravi baze posnetkov, saj je bilo treba poslušati terenske posnetke slovenske ljudske glasbe in v njih postavljati oznake začetnih časov pojavitve harmonike ter oznake začetnih časov, kjer harmonike ni. Čase smo kasneje uporabili za avtomatizirano rezanje posnetkov.

Orodje Timbre Toolbox je dodatno orodje za programsko okolje MATLAB, s pomočjo katerega smo izračunali značilnosti zvoka.

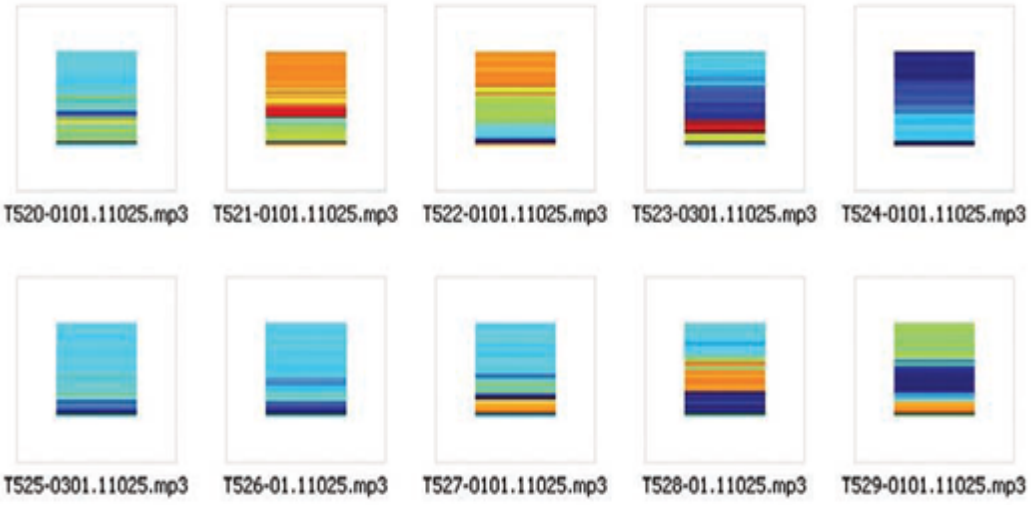
Orodje Orange smo uporabili za izračun pomembnosti atributov (značilnosti zvoka), za izgradnjo klasifikacijskega modela, za testiranje klasifikatorja ter na koncu za vrednotenje našega modela.

Programsko okolje MATLAB smo na začetku uporabili za izračun značilnosti zvoka z orodjem Timbre Toolbox, po izgradnji klasifikatorja v orodju Orange in testiranju le-tega pa smo celoten model tudi implementirali v okolju MATLAB.

2.1 EtnoMuza

EtnoMuza je digitalna multimedijska shramba slovenske ljudske glasbene in plesne kulture [9]. Gre za namizno aplikacijo, v katero so integrirana orodja za hrambo, anotacijo in upravljanje digitalnih vsebin in metapodatkov. Razvita je tudi spletna aplikacija, ki je namenjena širšemu krogu uporabnikov in predstavitvi zbirk EtnoMuze. Projekt EtnoMuza je bil razvit v Laboratoriju za grafiko in multimedije na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Za potrebe našega dela smo koristili zgolj urejevalnik glasbenih zvočnih posnetkov.

Urejevalnik glasbenih zvočnih posnetkov avtomatsko prepozna vsebino posnetka in na podlagi tega ustrezno vizualizira posamične segmente, segmentacijo pa lahko opravimo tudi ročno. Algoritem klasificira 3 sekundne odseke posnetka v pet razredov: govor, solo petje, večglasno petje, viža in pritrkavanje. Rezultat vizualizacije je obarvanje razredov z različnimi barvami. Barva vsakega dela posnetka je izračunana z interpolacijo med barvami razredov in upoštevanjem verjetnostne porazdelitve po razredih.



Slika 2.1: Predogled posameznega posnetka. Po deležu barv hitro razberemo, katera kategorija v posnetku prevladuje [9].

Avdio urejevalnik omogoča tudi predogled posameznega posnetka, ki predstavlja povzetek njegove vsebine. Predogled izračunamo z grupiranjem pripadnosti odsekov skladbe posamičnim kategorijam z algoritmom k-means, pri čemer pri izrisu predogleda uporabimo središče in velikost posamične skupine. Rezultat so generirane sličice, na podlagi katerih lahko hitro razločimo vsebino posnetka (slika 2.1).

2.2 Orange

Orange [1] je brezplačen in odprtokodni projekt, ki je bil razvit v Laboratoriju za bioinformatiko na Fakulteti za računalništvo in informatiko v Ljubljani. Uporablja se za procesiranje podatkov, vizualizacijo podatkov na več različnih načinov, strojno učenje, podatkovno rudarjenje, modeliranje, vrednotenje modelov, odkrivanje zakonitosti iz podatkov in za statistične raziskave. Orange je zgrajen in deluje na podlagi programskega jezika Python.

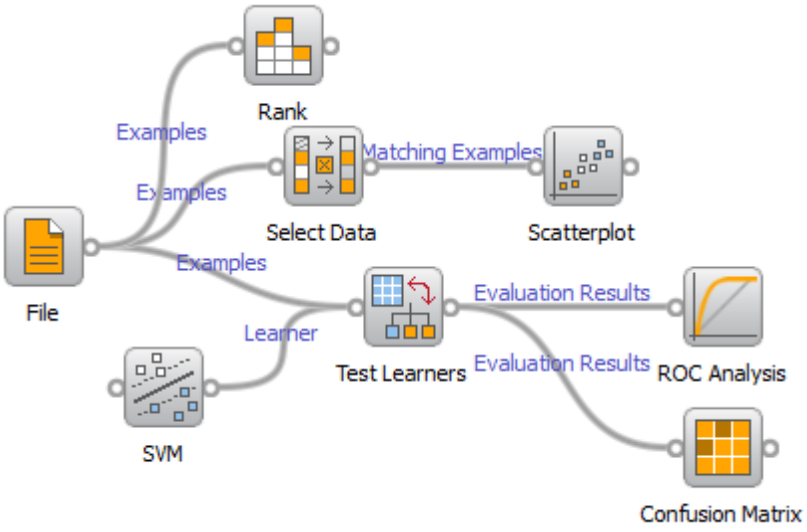
Orodje deluje na principu povezovanja gradnikov med seboj, kot lahko vidimo na sliki 2.2.

Ko odpremo poljubni gradnik, se odpre novo okno, kjer lahko spreminjamo nastavitve funkcij gradnika in opazujemo rezultate.

2.3 MATLAB

MATLAB [3] je programsko okolje podjetja MathWorks za razvoj algoritmov, analizo podatkov, vizualizacijo, numerično analizo, procesiranje slik, obdelavo digitalnih signalov ter meritve in testiranja. Uporaben je na področjih, kot so strojništvo, fizika, finančna matematika, računalništvo in ekonomija. Z orodjem MATLAB je reševanje problemov hitrejše kot s standardiziranimi programskimi jeziki, kot so C, C++ in Fortran.

Začetek razvoja orodja MATLAB sega v leto 1970, ko je Cleve Moler, predsednik oddelka za računalniško znanost na Univerzi v Novi Mehiki, svojim študentov želel omogočiti uporabo knjižnic LINPACK za linearno algebro



Slika 2.2: Primer sheme v orodju Orange. Na delovno površino lahko dodajamo različne gradnike, ki jih med seboj povezujemo.

in EISPACK za numerično računanje, ne da bi se študenti morali naučiti programski jezik Fortran. Inženir elektrotehnike Jack Little je na obisku Molerja leta 1983 opazil poslovni potencial in se mu pridružil skupaj z Stevom Bangertom. Leta 1984 so MATLAB predelali v programskem jeziku C, ustanovili podjetje MathWorks in nadaljevali z razvojem.

2.4 Timbre Toolbox

Timbre Toolbox [4] je orodje za meritev akustične strukture kompleksnih zvočnih signalov. Orodje je zmožno izračunati obsežen nabor zvočnih značilnosti, ki se uporabljajo na področju pridobivanja informacij iz glasbe ter prepoznavanja glasbenih inštrumentov s pomočjo strojnega učenja.

Zvočni dogodki so najprej analizirani z vidika različnih vhodnih predstavitev (kratko-časovna Fourierjeva transformacija, hitra Fourierjeva transformacija, harmonske sinusne komponente, ADSR ovojnica). Veliko število zvočnih značilnosti je nato pridobljenih iz vsake izmed teh predstavitev za zajem

časovnih, spektralnih, spektralno-časovnih in energijskih lastnosti zvočnih dogodkov. Nekaj izmed značilnosti je globalnih, za njih je izračunana samo ena vrednost za cel dogodek, ostale pa so časovno spreminjajoče. Za časovno spreminjajoče značilnosti je izračunanih veliko vrednosti, za vsak časovni okvir ena vrednost, zato so statistično obdelani. Na vrednostih se izračuna minimalna in maksimalna vrednost, aritmetična sredina, standardni odklon, mediana in interkvartilna razdalja med 25. in 75. percentilom.

Slika 2.3 prikazuje seznam vseh zvočnih značilnosti, izračunanih s pomočjo orodja Timbre Toolbox.

	Audio descriptor	Units	Abbreviation	Input representation
Global descriptors	Attack	s	Att	Temporal Energy Envelope
	Decay	s	Dec	
	Release	s	Rel	
	Log-Attack Time	log(s)	LAT	
	Attack Slope	a/s	AttSlope	
	Decrease Slope	log(a)/s	DecSlope	
	Temporal Centroid	s	TempCent	
	Effective Duration	s	EffDur	
	Frequency of Energy Modulation	Hz	FreqMod	
	Amplitude of Energy Modulation	a	AmpMod	
Time-varying descriptors	Autocorrelation (12 coefficients)	-	AutoCorr	Audio Signal
	Zero Crossing Rate	s^{-1}	ZcrRate	
	RMS-Energy Envelope	a	RMSEnv	Temporal Energy Envelope
	Spectral Centroid	F	SpecCent	STFTmagnitude (STFTmag) STFTpower (STFTpow) ERBfft (ERBfft) ERBgammatone (ERBgam) Harmonic
	Spectral Spread	F	SpecSpread	
	Spectral Skewness	-	SpecSkew	
	Spectral Kurtosis	-	SpecKurt	
	Spectral Slope	F^{-1}	SpecSlope	
	Spectral Decrease	-	SpecDecr	
	Spectral Rolloff	F	SpecRollOff	
	Spectro-temporal variation	-	SpecVar	
	Frame Energy	I	FrameErg	
	Spectral Flatness	-	SpecFlat	STFTmag, STFTpow, ERBfft, ERBgam
	Spectral Crest	-	SpecCrest	
	Harmonic Energy	a^2	HarmErg	Harmonic
	Noise Energy	a^2	NoiseErg	
	Noisiness	-	Noisiness	
	Fundamental Frequency	Hz	F0	
	Inharmonicity	-	InHarm	
	Tristimulus (3 coefficients)	-	TriStim	
	Harmonic Spectral Deviation	a	HarmDev	
	Odd to even harmonic ratio	-	OddEveRatio	

Slika 2.3: Seznam vseh značilnosti zvoka, ki jih izračunamo z dodatkom Timbre Toolbox za orodje MATLAB [5].

Poglavje 3

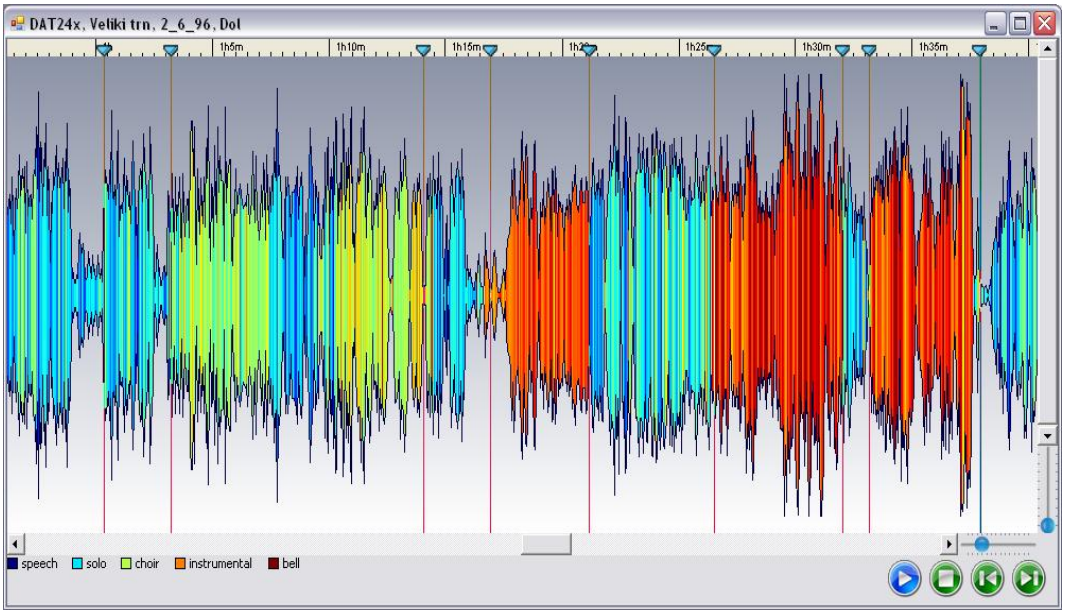
Priprava podatkov

3.1 Priprava baze posnetkov

Naš algoritem smo implementirali z metodo strojnega učenja in naredili klasifikator, ki se s pomočjo učne množice nauči razlikovati med dvema diskretnima razredoma. Ko klasifikator naučimo, ga moramo testirati s pomočjo testne množice in ugotoviti njegovo natančnost klasifikacije. V naši diplomski nalogi smo se ukvarjali s prepoznavanjem harmonike v posnetkih slovenske ljudske glasbe, zato smo celotno bazo zgradili iz terenskih posnetkov slovenske ljudske glasbe.

Za orodje EtnoMuza smo se odločili predvsem zato, ker se ob nalaganju posnetka v omenjenem orodju valovna oblika posnetka vizualizira tako, da se obarva glede na to, kaj vsebuje posnetek (pritrkavanje, govor, večglasno petje, solo petje, viža), kot lahko vidimo na sliki 3.1. Tako smo lahko zelo hitro ugotovili, kje v posnetku se nahaja igranje inštrumentov in med njimi iskali harmoniko.

V orodju EtnoMuza smo vsakega izmed izbranih terenskih posnetkov odprli ter ob poslušanju le-tega postavljali oznake na določenih mestih. Vsaka oznaka je pomenila začetni čas pojavitve harmonike, ali začetni čas, kjer harmonike ni. Ker orodje EtnoMuza omogoča izvoz vseh postavljenih oznak, smo kot rezultat izvoza dobili razpredelnico, v kateri je za vsako oznako za-



Slika 3.1: Orodje EtnoMuza - ob nalaganju posnetka slovenske ljudske glasbe se valovna oblika posnetka ustrezno obarva [9].

pisana pot do posnetka na trdem disku ter pozicija oznake v milisekundah.

Ko smo pridobili vse začetne čase, smo izračunali še končne čase tako, da smo začetnim prišteli 5 sekund. S pomočjo vseh oznak smo iz dolgih terenskih posnetkov z avtomatiziranim postopkom pridobili 936 posnetkov, dolgih 5 sekund. Med temi posnetki je bilo 468 posnetkov, ki vsebujejo harmoniko in 468 posnetkov, ki ne vsebujejo harmonike. Ker je za strojno učenje dobro, da imamo čim več podatkov, tako za učenje, kot za testiranje, smo se odločili za povečanje baze posnetkov. V orodju MATLAB smo napisali funkcijo, ki je vsakega izmed 936 posnetkov razrezal na 5 novih posnetkov, dolgih 3 sekunde. Vsak posnetek dolg 5 sekund je razrezal tako, da je iz njega zajel zvok od 0 do 3000 milisekund, od 500 do 3500 milisekund, od 1000 do 4000 milisekund, od 1500 do 4500 milisekund in od 2000 do 5000 milisekund. Tako smo na koncu dobili bazo, ki je sestavljena iz 4680 posnetkov, 2340 izmed njih s harmoniko in 2340 brez harmonike. Bazo smo razdelili na 3 enako velike skupine – učna in testna baza ter baza za izračun pomembnosti

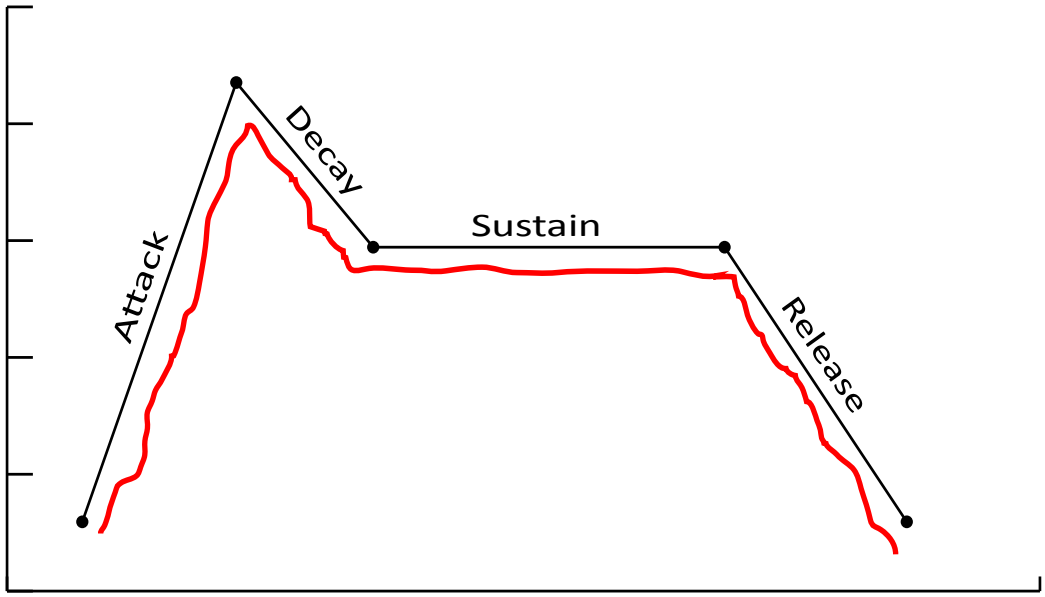
značilnosti zvoka.

3.2 Izračun značilnosti zvoka

Da bi lahko naučili algoritem, kako prepoznati nek inštrument v posnetku, kjer sočasno igra tudi eden ali več drugih inštrumentov, moramo zvok inštrumenta nekako opisati. Z opisom zvoka se ukvarjajo na področju pridobivanja informacij iz glasbe (*Music Information Retrival*). Na podlagi raziskav o zvoku je bilo zgrajeno orodje Timre Toolbox, ki je dodatek za orodje MATLAB. Orodje je za poljubno dolg zvočni posnetek zmožno izračunati veliko število atributov oziroma lastnosti zvoka, ki jim pravimo tudi značilnosti zvoka. Tako lahko s temi atributi opišemo zvok in z njihovo pomočjo naučimo algoritem, kakšne so vrednosti le-teh za določen inštrument oziroma kateri drugi vir zvoka.

Orodje Timbre Toolbox izračuna dve glavni skupini značilnosti, to so globalne značilnosti (angl. global descriptors) in časovno spreminjajoče značilnosti (angl. time-varying descriptors). Globalne značilnosti se izračuna za celoten signal, torej za vsako globalno značilnost izračunamo samo eno vrednost. Primer globalnih značilnosti so vrednosti ADSR ovojnice (slika 2.3). ADSR ovojnica je sestavljena iz 4 vrednosti. Prva je vzpon signala (Attack), druga je spust signala (Decay), tretja je trajanje signala (Sustain) in četrta sprostitev signala (Release).

Primer globalne značilnosti je vzpon signala, ki ima samo eno vrednost za celoten posnetek. Časovno spreminjajoče značilnosti so sestavljene iz sekvence vrednosti, ki so izračunane za vsak časovni okvir, dolg tipično 60 milisekund. Celoten posnetek je torej razdeljen na okvirje in za vsak okvir je izračunana ena vrednost. Ker je sekvenca teh vrednosti lahko zelo velika, smo za vsako značilnost izračunali mediano vseh vrednost ter interkvartilno razdaljo med 25. in 75. percentilom. Interkvartilna razdalja je pomnožena s koeficientom 0,7413 in s tem postane robustni ocenjevalec standardne deviacije normalno porazdeljenih podatkov. Enake časovno spreminjajoče značilnosti



Slika 3.2: Primer ADSR ovojnice.

so izračunane večkrat, vendar z vidika različnih vhodnih predstavitev signala. Signal je lahko predstavljen kot kratko-časovna Fourierjeva transformacija (STFTmagnitude in STFTpower), izhod zvočnega modela (ERBfft in ERBgammatone), harmonske sinusne komponente (Harmonic) ali kot navaden zvočni signal (Audio signal). Slika 2.3 prikazuje z vidika katerih vhodnih predstavitev signala so izračunane določene značilnosti zvoka.

Z orodjem Timbre Toolbox smo na koncu dobili 77 časovno spreminjajočih značilnosti ter 10 globalnih značilnosti, ker pa smo za vsako časovno spreminjajočo značilnost izračunali še mediano ter interkvartilno razdaljo, smo jih dobili 154. Končno število vseh značilnosti zvoka (v nadaljevanju atributov) je bilo 164.

Po izdelavi klasifikacijskega modela, ki je opisan v poglavju 4.1, smo se odločili zmanjšati število atributov in predvsem ugotoviti, katere skupine atributov najbolj vplivajo na točnost klasifikacijskega modela. Veliko število atributov točnost klasifikacije namreč zmanjšuje. Pri izračunu značilnosti zvoka so značilnosti razvrščene v 7 skupin, in sicer glede na vhodno predsta-

vitev signala. Kot je opisano v poglavju 4.1.1, smo na podlagi testiranj ugotovili, da je pri klasifikaciji posnetkov najbolj pomembna skupina značilnosti **ERBfft** (Equivalent Rectangular Bandwidth), ki so izračunane s hitro Fourierjevo transformacijo. Model ERB se uporablja na področju psihoakustike in je približek pasovnim širinam filtrov pri človeškem slušnem zaznavanju, uporablja namreč pravokotne pasovne filtre. Sledi opis vseh značilnosti, ki so kot vhodni signal predstavljene z ERBfft modelom. Opis značilnosti je povzet po poročilu *A large set of audio features for sound description* [6].

3.2.1 Spektralni centroid

Spektralni centroid (angl. spectral centroid) je težišče spektra zvoka. Pri izračunu se spekter smatra kot porazdelitev, katere vrednosti so frekvence in verjetnosti, da gre za normalizirane amplitude:

$$\mu = \int x \cdot p(x)\,\mathrm{d}x$$

kjer je

- $x = freq\_v(x)$
- $p(x) = \dfrac{ampl\_v(x)}{\sum\limits_{x} ampl\_v(x)}$

3.2.2 Spektralni odklon

Spektralni odklon (angl. spectral spread) izračunamo na podlagi spektralnega centroida. Spektralni odklon predstavlja odklon spektra od srednje vrednosti:

$$\sigma^2 = \int (x-\mu)^2 \cdot p(x)\,\mathrm{d}x$$

3.2.3 Spektralna asimetrija

Pri računanju spektralne asimetrije (angl. spectral skewness) merimo asimetrijo porazdelitve spektra okoli srednje vrednosti. Izračunana je s pomočjo momenta tretjega reda:

$$m_3 = \int (x - \mu)^3 \cdot p(x)\,\mathrm{d}x$$

Spektralna asimetrija je:

$$\gamma_1 = \frac{m_3}{\sigma^3}$$

Asimetrija SK opisuje stopnjo asimetrije porazdelitve:

- $SK = 0$; porazdelitev je simetrična
- $SK < 0$; več energije pri frekvencah z nižjo vrednostjo od srednje vrednosti
- $SK > 0$; več energije pri frekvencah z višjo vrednostjo od srednje vrednosti

3.2.4 Spektralna sploščenost

Pri računanju spektralne sploščenosti (angl. spectral kurtosis) merimo ploskost porazdelitve okoli srednje vrednosti. Izračunana je s pomočjo momenta četrtega reda:

$$m_4 = \int (x - \mu)^4 \cdot p(x)\,\mathrm{d}x$$

Spektralna sploščenost je:

$$\gamma_2 = \frac{m_4}{\sigma^4}$$

Spektralna sploščenost K označuje koničastost porazdelitve:

- $K = 3$; normalna porazdelitev
- $K < 3$; ploska porazdelitev
- $K > 3$; koničasta porazdelitev

3.2.5 Spektralni naklon

Spektralni naklon (angl. spectral slope) predstavlja velikost upada spektralne amplitude. Izračunan je z linearno regresijo spektralne amplitude:

$$\hat{a}(f) = slope \cdot f + const$$

kjer je

$$slope = \frac{1}{\sum\limits_{k} a(k)} \frac{N \sum\limits_{k} f(k) * a(k) - \sum\limits_{k} f(k) * \sum\limits_{k} a(k)}{N \sum\limits_{k} f^2(k) - \Big(\sum\limits_{k} f(k) \Big)^2}$$

3.2.6 Spektralna ravnost

Z izračunom spektralne ravnosti (angl. spectral flatness) oziroma koeficienta tonalnosti ugotavljamo, ali je zvok podoben tonu ali šumu. Na tem mestu si lahko tonskost razlagamo kot število vrhov v spektru moči signala. Če je vrednost spektralne ravnosti blizu 1, gre za šum. Pri šumu je moč spektra približno enaka na vseh frekvenčnih pasovih spektra. Če je vrednost spektralne ravnosti 0, potem gre za tonski signal. Spektralna ravnost je izračunana kot razmerje med geometrijsko sredino in aritmetično sredino energijske vrednosti spektra:

$$SFM(num\_band) = \frac{\Big(\prod\limits_{k \in num\_band} a(k) \Big)^{1/K}}{\frac{1}{K} \sum\limits_{k \in num\_band} a(k)}$$

kjer je a(k) amplituda v frekvenčnem pasu k.

Koeficient tonalnosti lahko izračunamo na podlagi spektralne ravnosti:

$$SFM_{db} = 10 \cdot log_{10}(SFM)$$

$$Tonality = min\Big(\frac{SFM_{db}}{-60}, 1\Big)$$

3.2.7 Spektralni vrh

Spektralni vrh (angl. spectral crest) je značilnost, povezana s spektralno ravnostjo. Spektralni vrh je izračunan kot razmerje med maksimalno vrednostjo v pasu in aritmetično sredino energijske vrednosti spektra:

$$SCM(num\_band) = \frac{max(a(k \in num\_band))}{\frac{1}{K}\sum\limits_{k \in num\_band} a(k)}$$

3.2.8 Spektralni padec

Tako kot spektralni naklon tudi spektralni padec (angl. spectral decrease) predstavlja velikost upada spektralne amplitude. Ta formula je narejena na podlagi študij slušnega zaznavanja, zato bi morala biti v korelaciji s človeškim slušnim zaznavanjem:

$$decrease = \frac{1}{\sum\limits_{k=2:K} a} \sum_{k=2:K} \frac{a(k) - a(1)}{k-1}$$

3.2.9 Spektralni upad

Spektralni upad (angl. spectral roll-off) je način za oceno količine energije pri visokih frekvencah signala. Točka spektralnega upada je frekvenca, pod

katero se nahaja 95% vse energije signala. Spektralni upad je povezan z mejno frekvenco med šumom in harmoničnim delom spektra. Določen je s formulo:

$$\sum_{0}^{fc} a^2(f) = 0.95 \sum_{0}^{sr/2} a^2(f)$$

3.2.10 Spektralni pretok

Spektralni pretok (angl. spectral variation) predstavlja količino variacije spektra skozi čas. Izračunan je s pomočjo normalizirane navzkrižne korelacije (standardna metoda za ocenjevanje stopnje korelacije dveh vrst) zaporednih amplitud spektrov *a(t-1)* in *a(t)*:

$$variation = 1 - \frac{\sum_{k} a(t-1,k) \cdot a(t,k)}{\sqrt{\sum_{k} a(t-1,k)^2} \sqrt{\sum_{k} a(t,k)^2}}$$

Vrednost pretoka je blizu števila 1, če sta spektra zelo podobna, in bližje številu 0, če sta si spektra različna.

3.2.11 Energija okvirja

Energija okvirja je izračunana kot vsota kvadratov vseh amplitud v času t_m:

$$E_T(t_m) = \sum_{k} {a_k}^2(t_m)$$

3.3 Priprava tabele podatkov

Za izdelavo in vrednotenje klasifikacijskega modela smo uporabili orodje Orange, ki zna prebrati določeno obliko podatkov. Podatki morajo biti v obliki 2D tabele, kjer stolpci predstavljajo atribute, vrstice pa primere. Kot

je opisano v poglavju 3.2, smo izračunali vrednosti 164 atributov za vsak posnetek, katerih skupno število v naši bazi je znašalo 4680. Za vsak posnetek smo dodali še en stolpec, ki je predstavljal diskretni razred z imenom harmonika. Za vsak posnetek smo kot vrednost razreda pripisali eno izmed naslednjih dveh - vrednost 0 (posnetek ne vsebuje harmonike) ali vrednost 1 (posnetek vsebuje harmoniko). Tako smo vsak posnetek opisali s 164 atributi in razredom. Ko smo vse podatke sestavili skupaj, smo dobili tabelo z 772200 številskimi vrednostmi (4680 vrstic in 165 stolpcev).

Poglavje 4

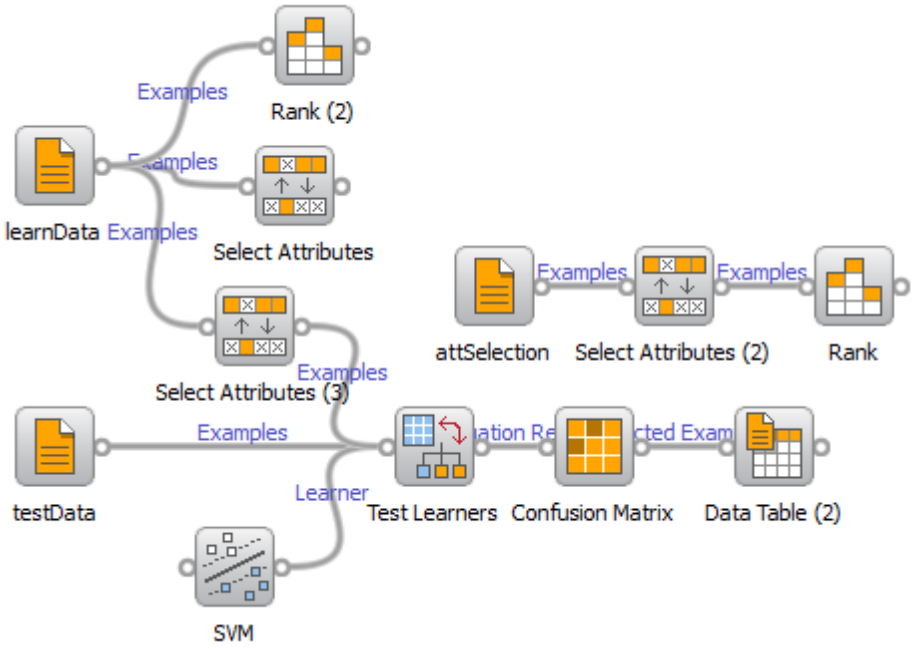
Razvoj algoritma

4.1 Izdelava klasifikacijskega modela

Klasifikacijski model, ki je prikazan na sliki 4.1, smo izdelali v orodju Orange. Zanimalo nas je, katere skupine atributov najbolj vplivajo na klasifikacijo posnetkov in kakšna je klasifikacijska točnost.

Podatke za učenje klasifikatorja smo odprli z gradnikom File (v naši shemi poimenovan learnData) in ga povezali z gradnikom Select Attributes, kateri omogoča izločitev določenih atributov. Slednji na izhodu vrne skrčeno tabelo, ki vsebuje samo tiste argumente, katere izberemo v gradniku. S tem gradnikom smo kasneje izbirali samo določene skupine atributov in z vsako skupino posebej opravljali testiranje klasifikatorja ter opazovali klasifikacijsko točnost. Tudi podatke za testiranje klasifikatorja smo odprli z gradnikom File (v naši shemi poimenovan testData). Učne in testne podatke smo nato povezali z gradnikom Test Learners, s pomočjo katerega izračunamo klasifikacijsko točnost. Gradnik Test Learners lahko povežemo z različnimi metodami za klasifikacijo, v našem primeru pa se je kot najboljša izkazala metoda podpornih vektorjev (opisana v nadaljevanju) z radialnim jedrom. V gradniku Test Learners lahko tudi izberemo na kakšen način bomo testirali klasifikator. Klasifikator lahko učimo in testiramo z enako skupino primerov, pri čemer uporabimo metodo prečnega preverjanja s K-pregibi, kjer se učna

množica naključno razdeli na K podmnožic. Ena od podmnožic se uporabi za testiranje, preostalih K-1 podmnožic pa za učne podatke. Postopek se ponovi K-krat in pri vsaki ponovitvi je vsaka od K podmnožic samo enkrat uporabljena za testne podatke. V našem primeru smo se odločili za posebno testno množico, ki je bila enako velika kot učna množica. Izhod gradnika Test Learners lahko povežemo z gradnikom Confusion Matrix, kjer lahko opazujemo, koliko primerov je klasifikator pravilno ali napačno klasificiral. Če želimo videti, za katere primere konkretno je pravilno ali napačno napovedal razred, povežemo izhod gradnika z gradnikom Data Table, kjer si v obliki tabele ogledamo vse izbrane primere in njihove atribute. Pomembnosti atributov smo izračunali s pomočjo gradnika Rank, kjer lahko z različnimi algoritmi računamo pomembnost atributov. Za ocenjevanje atributov smo uporabili algoritem ReliefF (Kononenko, 1994).

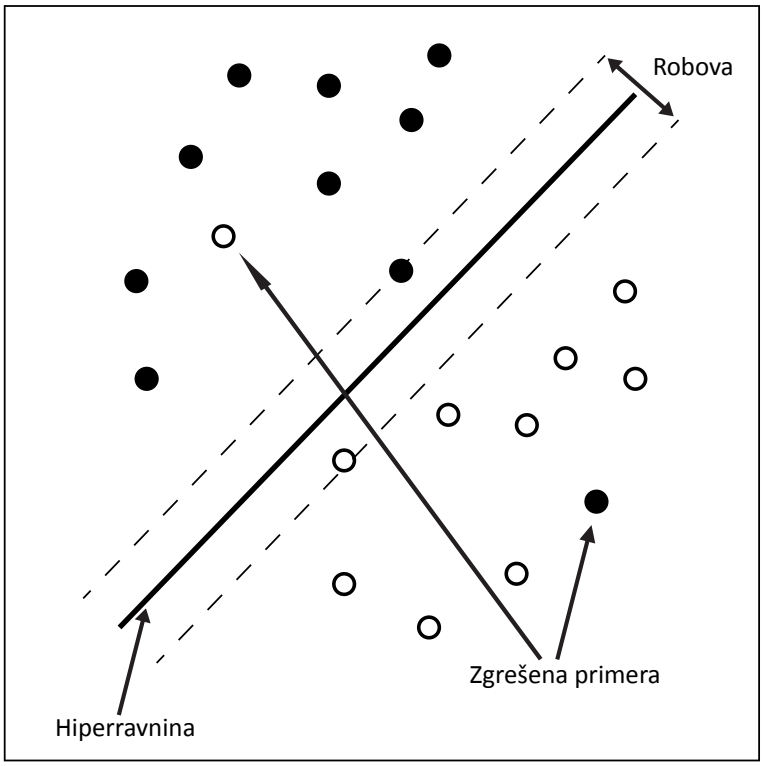


Slika 4.1: Shema klasifikacijskega modela v orodju Orange. Klasifikator deluje na podlagi metode podpornih vektorjev (SVM).

Metoda podpornih vektorjev

Metoda podpornih vektorjev (angl. Support Vector Machine s kratico SVM) je metoda razvrščanja [8]. SVM razdeli množico predmetov v razrede tako, da je meja med razredi čim večja.

Na začetku imamo množico učnih primerov, za katere vemo tudi, kateremu razredu pripadajo. Vsak učni primer je predstavljen z vektorjem v vektorskem prostoru (z n-dimenzijami). Cilj metode podpornih vektorjev je dobiti klasifikator, ki bi razločeval med razredi. SVM sprva poišče optimalno lego hiperravnine, s katero razmeji oba razreda in maksimizira razdalje vektorjev, ki ležijo najbližje hiperravnini (slika 4.2).



Slika 4.2: Delovanje SVM. Razreda sta razdeljena s hiperravnino, ob kateri je prazno območje, ki predstavlja rob obeh razredov.

Zaradi maksimizacije razdalj se med hiperravnino in razredoma ustvari

prazno območje, ki kasneje omogoča razvrščanje tudi tistih primerov, za katere ni gotovo trditi, v kateri razred spadajo (niso podobni učnim primerom). Vektorji, ki ležijo daleč od hiperravnine, ne vplivajo na lego hiperravnine. Najbolj na lego hiperravnine vplivajo vektorji najbližje njej, te imenujemo podporni vektorji. Ko vektorji niso linearno ločljivi, jih lahko transformiramo tako, da jim povečamo dimenzijo. Če dimenzijo povečamo dovolj, postanejo vsi razredi vektorjev linearno ločljivi. Za transformacijo lahko uporabimo različna jedra. Poznamo linearno, polinomsko, radialno in sigmoidno jedro.

4.1.1 Vrednotenje klasifikacijskega modela

Po izdelavi klasifikacijskega modela sledi še izbor atributov. V naši tabeli podatkov smo imeli 4680 primerov s 164 atributi in 1 diskretnim razredom, kjer vsak atribut predstavlja eno izračunano značilnost zvoka, razred pa prisotnost harmonike. Kot smo že omenili, smo tabelo razdelili na tri enake dele (testna množica, učna množica in množica za izračun pomembnosti atributov). Predpostavili smo, da klasifikacijska točnost pri vseh uporabljenih atributih ne bo največja možna. Zanimalo nas je, kakšna je klasifikacijska točnost z uporabo vseh 164 atributov ter točnost z uporabo določenih skupin atributov. Značilnosti zvoka so, kot je opisano v poglavju 3.2, razporejene v 7 skupin, glede na način obravnavanja vhodnega signala pri izračunu značilnosti. Z gradnikom Select Attributes smo izbirali le posamezne skupine značilnosti in jih kot atribute uporabili pri učenju klasifikatorja. Za vsako testiranje z ločeno testno množico smo opazovali klasifikacijsko točnost in na podlagi ocen pomembnosti atributov odstranjevali najmanj pomemben atribut. Atribute smo odstranjevali, dokler se je klasifikacijska točnost izboljševala. Ko se je z odstranitvijo nekega atributa iz skupine klasifikacijska točnost poslabšala, smo zaključili z odstranjevanjem, ker smo prišli do največje točnosti klasifikatorja pri tej skupini atributov. Tabela 4.1 prikazuje rezultate učenja in testiranja klasifikatorja z vsemi skupinami atributov. Ugotovili smo, da je pri klasifikaciji najpomembnejša skupina atributov ERBfft z vsemi atributi, ki jih vsebuje. Skupina je sestavljena iz 22 atributov, med katerimi sta po

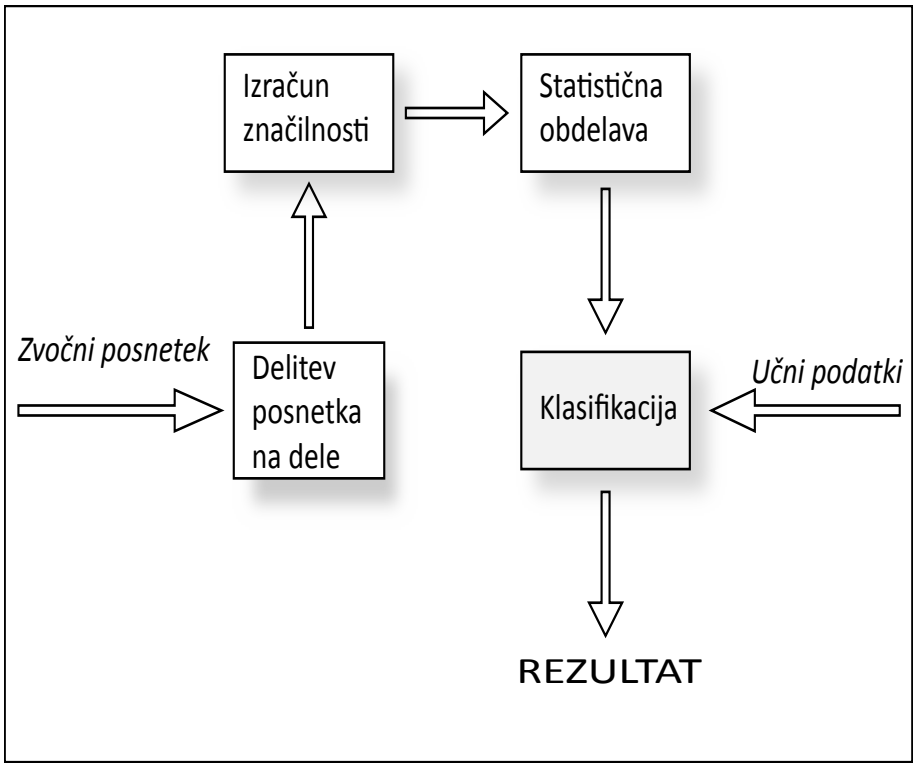
	št. znač.	klas. točnost
TEE	1/22	63,14%
AS	12/26	81,92%
STFTpow	22/22	90,77%
STFTmag	21/22	93,40%
Harmonic	30/38	93,91%
ERBgam	22/22	95,25%
ERBfft	22/22	**95,83%**
vse skupine	164	88,14%

Tabela 4.1: Tabela prikazuje rezultate učenja in testiranja klasifikatorja z različnimi skupinami atributov. Prvi stolpec prikazuje skupine atributov, drugi število pomembnih značilnosti za posamezno skupino in tretji stolpec klasifikacijsko točnost ob upoštevanju samo določene skupine atributov. V zadnji vrstici so rezultati klasifikacije ob upoštevanju vseh 164 atributov.

dve statistični vrednosti (mediana in interkvartilna razdalja) za vsako izmed 11 značilnost zvoka.

Klasifikator, ki deluje z metodo podpornih vektorjev, smo torej naučili na 1560 učnih primerih, pri katerih smo upoštevali samo atribute skupine ERBfft in na koncu klasifikator še testirali s 1560 testnimi primeri. Točnost klasifikacijskega modela ob upoštevanju vseh 164 atributov je 88,14%, ob upoštevanju atributov zgolj iz skupine ERBfft pa 95,83%.

Poskušali smo ugotoviti tudi, kateri posnetki so bili napačno klasificirani in zakaj. Na podlagi poslušanja teh posnetkov smo ugotovili, da so bili med posnetki, ki vsebujejo harmoniko, napačno klasificirani tisti, kjer kakšen drug inštrument očitno izstopa, kjer je večglasno petje v ospredju in harmonika igra zelo potiho, kjer je poleg harmonike glasno vriskanje, ter v posnetkih, ki vsebujejo zaključek pesmi in harmonika tiho izzveni. Med posnetki, ki ne vsebujejo harmonike, pa so bili večinoma napačno klasificirani tisti, ki vsebujejo večglasno petje ali kjer igra violina in več tamburic.



Slika 4.3: Shema delovanja MATLAB algoritma za klasifikacijo posnetkov.

4.2 Implementacija algoritma v orodju MATLAB

V orodju Orange smo izdelali klasifikacijski model, raziskali pomembnost atributov, ga testirali in ovrednotili. V orodju MATLAB smo implementirali algoritem, ki deluje na principu izdelanega klasifikacijskega modela. Cilj algoritma je bil prebrati poljubno dolg posnetek slovenske ljudske glasbe in ga po 3-sekundnih odsekih klasificirati v eno izmed dveh skupin - vsebuje harmoniko ali ne vsebuje harmonike. Rezultati klasifikacije se po izračunu značilnosti in klasifikaciji zapišejo v tekstovno datoteko. Slika 4.3 prikazuje shemo delovanja algoritma, implementiranega v okolju MATLAB.

Algoritem uporabljamo tako, da v ukazni vrstici MATLAB kličemo funkcijo *classifyWav* z dvema argumentoma. Kot prvi argument moramo podati

pot do posnetka, nad katerim želimo izvesti klasifikacijo, kot drugi argument pa navedemo, kam naj se rezultat klasifikacije zapiše. Ko algoritem prebere posnetek, ga razdeli na več 3-sekundnih posnetkov tako, da na posnetku postavlja dve oznaki (začetek in konec) in 3-sekundni zvočni zapis med njima izloči in shrani, obe oznaki pa pomakne za 0,5 sekunde naprej. Koraki se izvajajo tako dolgo, dokler oznaka konec ne pristane na koncu podanega posnetka. Sledi izračun značilnosti skupine ERBfft na vseh 3-sekundnih odsekih, nad temi podatki pa še izračun statistik. Za vsako značilnost sta izračunani dve statistični vrednosti, mediana in interkvartilna razdalja. Na tem mestu ima algoritem pripravljeno tabelo vseh značilnosti za vsak 3-sekundni odsek, ki je primerna za klasifikacijo. Na podlagi teh podatkov se vsak odsek nato klasificira v eno izmed dveh skupin (harmonika ali ni harmonika), rezultat pa se zapiše v tekstovno datoteko.

Poglavje 5

Sklepne ugotovitve

V diplomskem delu smo implementirali algoritem za avtomatsko prepoznavanje glasbenega inštrumenta harmonika v posnetkih slovenske ljudske glasbe. Pripravili smo bazo posnetkov za učno in testno množico, izrezanih iz dolgih terenskih posnetkov slovenske ljudske glasbe, za klasifikacijo pa smo uporabili metodo podpornih vektorjev, ki se je izkazala za zelo učinkovito. Po izračunu značilnosti zvoka s pomočjo orodja Timbre Toolbox smo ugotovili, da je pri klasifikaciji v našem algoritmu najbolj pomembna skupina značilnosti ERB-fft, v kateri je skupaj 22 značilnosti.

Algoritem kot vhodni argument sprejme poljubno dolg zvočni posnetek formata .wav, zanj izračuna zvočne značilnosti in ga po 3 sekundnih odsekih klasificira v eno izmed dveh skupin – vsebuje harmoniko ali ne vsebuje harmonike. Rezultat klasifikacije je tekstovna datoteka, v kateri je zapisan rezultat klasifikacije za vsak časovni interval, dolg 3 sekunde. Klasifikacijska točnost algoritma je zelo visoka, dobrih 95%, delovanje algoritma pa je omejeno. Algoritem dobro deluje izključno pri klasifikaciji posnetkov slovenske ljudske glasbe in prepoznava samo en inštrument, to je harmonika.

Poleg omejenosti algoritma bi navedel še problem časovne kompleksnosti algoritma. Glavni vzrok za veliko časovno kompleksnost algoritma je računanje značilnosti zvoka, saj pri tem procesu algoritem porabi približno eno tretjino časa trajanja obdelovanega posnetka na računalniku z dvoje-

dernim procesorjem Intel Core i3 M350 pri frekvenci 2.27GHz in delovnim pomnilnikom velikosti 4GB. Če je torej posnetek, ki ga želimo klasificirati, dolg 60 minut, računanje značilnosti traja približno 20 minut. Kljub temu, da smo pri izračunu značilnosti računali vrednosti samo ene izmed sedmih skupin značilnosti, sam izračun še vedno traja precej dolgo. Če torej ne bi naredili izbora značilnosti, bi bil algoritem ne samo manj natančen, ampak tudi veliko bolj časovno kompleksen. Sama klasifikacija 3-sekundnih odsekov se izvrši zelo hitro, saj pri zagonu algoritma ni potrebno znova graditi klasifikatorja, ker že naučeni klasifikator v obliki strukture v okolju MATLAB preprosto naložimo v pomnilnik, kar se zgodi v zanemarljivem času.

Ideja za nadaljnje delo bi bila izdelava vizualizacije rezultata klasifikacije po zgledu vizualizacije v avdio urejevalniku aplikacije EtnoMuza. Izdelali bi preprost avdio predvajalnik, ki bi kot rezultat klasifikacije ustrezno pobarval segmente posnetka. Segmente, v katerih je harmonika, bi obarvali z eno, in segmente, kjer je vse ostalo, z drugo barvo. Tako bi lahko uporabnik hitro preskočil dele, ki ga ne zanimajo in našel segmente s harmoniko ter jih poslušal.

Slike

Tabele

Literatura

[1] ORANGE. Dostopno na:

`http://orange.biolab.si/`

[2] Spletna stran projekta Etnomuza. Dostopno na:

`http://lgm.fri.uni-lj.si/matic/ethnomuse/`

[3] Uradna spletna stran orodja MATLAB. Dostopno na:

`http://www.mathworks.com/products/matlab/`

[4] Timbre Toolbox. Dostopno na:

`http://www.cirmmt.mcgill.ca/l/research-tools/timbretoolbox/`

[5] Peeters G., Giordano B.L., Susini P., Misdariis N, McAdams S. *The Timbre Toolbox: Extracting audio descriptors from musical signals*, 2011. Dostopno na:

`http://mt.music.mcgill.ca/mpcl/publications/peeters-giordano-susini-misdariis-mcadams-2011`

[6] G. Peeters. *A large set of audio features for sound description*, 2004. Dostopno na:

`http://recherche.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf`

[7] T. Heittola, A. Klapuri, T. Virtanen. *Musical instrument recognition in polyphonic audio using source-filter model for sound separation*, ISMIR 2009. Dostopno na:

http://ismir2009.ismir.net/proceedings/OS3-2.pdf

[8] Metoda podpornih vektorjev. Dostopno na:

http://sl.wikipedia.org/wiki/Metoda_podpornih_vektorjev

[9] Marolt, Matija, Strle, Gregor (2010). Etnomuza. Traditiones, letnik 39, številka 2, str. 149-166. URN:NBN:SI:DOC-ILNNUYL7 from http://www.dlib.si

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00328/2012
Datum: 03.09.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **VITJA KLUN**

Naslov: **DETEKCIJA HARMONIKE V POSNETKIH LJUDSKE PESMI**
ACCORDION DETECTION IN FOLK MUSIC RECORDINGS

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

V diplomskem delu razvijte algoritem, ki v posnetkih slovenske ljudske glasbe detektira igranje na harmoniko. Pri tem ustvarite učno in testno množice pesmi z in brez harmonike, preučite katere značilke so najbolj primerne in izberite algoritem strojnega učenja, ki bo na bazi značilk opravljal detekcijo igranja na harmoniko.

Mentor:

doc. dr. Matija Marolt

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Nikolaj Zimic