Razdvajanje instrumentala i glasa u audio datoteci

Vodička, David

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

RAZDVAJANJE INSTRUMENTALA I GLASA U AUDIO DATOTECI

Završni rad

David Vodička

Osijek, 2023.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 15.09.2023.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	David Vodička
Studij, smjer:	Programsko inženjerstvo
Mat. br. Pristupnika, godina upisa:	R4586, 28.07.2020.
OIB Pristupnika:	24909584532
Mentor:	doc. dr. sc. Hrvoje Leventić
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Razdvajanje instrumentala i glasa u audio datoteci
Znanstvena grana rada:	**Obradba informacija (zn. polje računarstvo)**
Zadatak završnog rad:	Istražiti i opisati postojeće metode i aplikacije za razdvajanje instrumentala i glasa u audio datoteci. Opisati potrebne teorijske osnove iz obrade signala. Za praktični dio implementirati algoritam za razdvajanje instrumentala i glasa iz audio datoteke. Rezervirano za: David Vodička
Prijedlog ocjene završnog rada:	Dobar (3)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 1 bod/boda Jasnoća pismenog izražavanja: 1 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene od strane mentora:	15.09.2023.
Datum potvrde ocjene od strane Odbora:	24.09.2023.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 25.09.2023.

Ime i prezime studenta:	David Vodička
Studij:	Programsko inženjerstvo
Mat. br. studenta, godina upisa:	R4586, 28.07.2020.
Turnitin podudaranje [%]:	6

Ovom izjavom izjavljujem da je rad pod nazivom**: Razdvajanje instrumentala i glasa u audio datoteci**

izrađen pod vodstvom mentora doc. dr. sc. Hrvoje Leventić

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD

Odvajanje instrumenata i glasa u audio datoteci grana je obrade audio signala koja ima za cilj izdvajanje pojedinačnih izvora zvuka iz mješovite audio snimke. S napretkom tehnologije i sve većom potražnjom za visokokvalitetnim zvukom, odvajanje je postalo važno područje istraživanja i razvoja. Sposobnost izdvajanja specifičnih instrumenata ili vokala iz složene glazbene mješavine ima brojne primjene, u rasponu od poboljšavanja glazbenih zapisa do poboljšanja razumljivosti govora u prepunim okruženjima.

U području obrade zvuka razne metode razdvajanja koriste se za izdvajanje zvukova u mješinom audio signalu. Pojam izvor koristi se za signale koji tvore miješani signal, dok se zadatak naziva slijepo odvajanje izvora (BSS – *eng.* Blind Source Separation)[1]. Proces odvajanja jedno kanalnih izvora zvuka poseban je slučaj odvajanja izvora zvuka jer se odvajanje vrši pomoću jednog miješanog signala. Ovo dodaje još jedan izazov jer se različiti signali preklapaju u vremenu i frekvenciji, što otežava proces odvajanja.

U prvom poglavlju razmatramo problematiku zadatka i načine pristupa rješavanju istog, dok se u drugome poglavlju osvrćemo na postojeće rješenje i rezultate prikazane u[1]. U trećem poglavlju se prikazuju osnovni alati potrebni za razdvajanje, a u četvrtom poglavlju ćemo se osvrnuti na metode za izdvajanje i analizu značajki koje su korisne u identificiranju i razlikovanju različitih audio komponenti. U petom poglavlju su objašnjene metode strujnog učenja, a u šestom su detaljnije objašnjene duboke neuronske mreže. Sedmo poglavlje prikazuje primjere primjene razdvajanja koji se koriste u stvarnome svijetu. U osmom i devetom se prikazuju dvije metode za razdvajanje od kojih je metoda prikazana u osmom poglavlju razvijena unutar Facebook AI Research tima i postignuti su rezultati visoke kvalitete dok se u devetome poglavlju prikazuje metoda koja koristi jednostavne alate za razdvajanje. Deseto poglavlje objašnjava dobivene rezultate obje metode, te daje mogućnost da se poslušaju dobiveni rezultati.

1.1. Zadatak završnog rada

Osnovni cilj odvajanja glazbenih izvora je razdvojiti i izolirati sastavne izvore iz mješovite audio snimke, precizno razaznajući svaki instrument ili glas. Ovaj proces, koji se nekoć smatrao ogromnim računalnim izazovom, doživio je značajan napredak u posljednjih nekoliko godina potaknut napretkom metoda obrade signala, napretkom strojnog učenja i eksponencijalnim rastom računalnih resursa.

Tradicionalne metode za odvajanje oslanjale su se na ručno uređivanje ili jednostavne tehnike filtriranja, ali nedavni napredak u algoritmima strojnog učenja revolucionirao je ovo područje. Potrebno je istražiti principe, izazove i tehnike uključene u odvajanje objašnjavajući potencijalne primjene.

2. POSTOJEĆA RJEŠENJA ZA RAZDVAJANJE

U ovome poglavlju ćemo razmotriti problematiku razdvajanja vokala i instrumenata te jedan od načina kako razdvojiti vokal i instrumente i do kojih rezultata su došli u [1].

2.1. Metodologija problema

Proces odvajanja signala audio izvora od jedno kanalnog miješanog signala zahtijeva procjenu (S) izvora iz mješovitog signala, kao u jednadžbi (2-1):[1]

$$x(t) = \sum_{i=1}^{S} y_i(t) \tag{2-1}$$

Gdje je $y_i(t)$, i = 1…S, i-ti izvor koji se procjenjuje, dok je x(t) promatrani miješani signal. Pojednostavljeno, pretpostavljamo da se miješani signal sastoji od dva različita signala $s_1(t)$, $s_2(t)$ kao u jednadžbi (2-2):[1]

$$x(t) = s_1(t) + s_2(t) \tag{2-2}$$

Ovaj problem se može riješiti u domeni kratke Fourierove transformacije (SFFT – *eng.* Short Time Fourier Transform). Neka *X(n, f)* bude odgovarajući STFT od miješanog signala *x(t)*, gdje *t* označava vremensku domenu, a *n* predstavlja indeks okvira i *f* predstavlja frekvencijski indeks STFT domene signala. Ovaj problem se može formulirati na sljedeći način:[1]

$$X(n, f) = S_1(n, f) + S_2(n, f) \tag{2-3}$$

Gdje su $S_1(n, f)$ i $S_2(n, f)$ nepoznanice izvora u miješanom signalu. S obzirom na $X(n, f)$ cilj odvajanja monofonskog izvora je oporaviti jedan ili više željenih signala iz miješanog signala. Tipična pretpostavka je da su dostupni samo spektri magnitude a razlike između faznih kutova izvora zanemaruju se tijekom razdvajanja. Ovo se koristi samo kada je potrebno rekonstruirati valne oblike izvora u vremenskoj domeni. Spektrogram magnitude mjerenog audio signal se može napisati kao zbroj spektrograma magnitude signala izvora kako slijedi:[1]

$$|X_n| \approx |S_1(n, f)| + |S_2(m, f)| \tag{2-4}$$

Koristimo matrični oblik za predstavljanje spektrograma magnitude gdje *n* i *f* označavaju spektralni indeks okvira i frekvencije, kako slijedi:[1]

$$X(n,f) \approx S_1(n,f) + S_2(n,f) \tag{2-5}$$

Gdje su $S_1(n, f)$ i $S_2(n, f)$ spektrogrami nepoznate magnitude izvora koje je potrebno procijeniti. Spektrogram magnitude miješanog signala $|X(n, f)|$, zajedno sa spektralnim značajkama unose se u (DNN – *eng.* Deep Neural Network) za predviđanje vremensko-frekvencijske maske za svakog govornika. Maske se množe s miješanim signalom pomoću operacije množenja po elementima za procjenu veličine STFT željenog izvora. Odvojeni valni oblici procijenjenog izvora ponovno se sintetiziraju korištenjem inverznog STFT, procijenjene veličine izvora i informaciji o šumu.[1]

2.2. Procjena maske pomoću duboke neuronske mreže

Duboke neuronske mreže obično se koriste za predviđanje maske. Ova se maska koristi za određivanje doprinosa svakog izvora u mješovitom signalu za treniranje. Ulaz u mrežu je spektrogram magnitude mješovitog signala za obuku (X_{tr}), izvorni signali za svaki izvor (S_{tr1}) i (S_{tr2}). Nakon treniranja modela, predviđanje izlaza mreže je (S'_{tr1}) i (S'_{tr2}). Kako bi izgladili rezultate spektra, dvije vrste maski se zasebno procjenjuju modelom, prva maska je maska idealnog omjera (IRM) koja je izglađeni oblik idealne binarne maske (IBM), (IRM) maska predstavljena u jednadžbi (2-6):[1]

$$IRM_s = \left(\frac{S_1(t,f)^2}{S_1(t,f)^2 + S_2(t,f)^2}\right)^{\beta} \tag{2-6}$$

Gdje su $S_1(t, f)^2$ i $S_2(t, f)^2$ energija svakog izvora signala u mješovitom signalu. β je vrijednost kvadratnog korijena koji se koristi kao parametar koji se može podešavati, a koji se obično postavlja na 0,5. IRM maska koristi se za očuvanje energije izvora svake TF(vremensko-frekvencijske) jedinice, pod pretpostavkom da S_1 i S_2 nisu u korelaciji. Drugi maska je maska optimalnog omjera (ORM) koja se može izvesti minimiziranjem srednje kvadratne pogreške (MSE) između čistog signala izvora i ciljanog signala izvora koji je procijenjen i pomoću kojeg se može definirati jednadžba (2-7):[1]

$$M(t,f) = \frac{|S_1(t,f)|^2 + R\left(S_1(t,f) \times S_2{*}(t,f)\right)}{|S_1(t,f)|^2 + |S_2(t,f)|^2 + 2 \times R\left(S_1(t,f) \times S_2{*}(t,f)\right)} \tag{2-7}$$

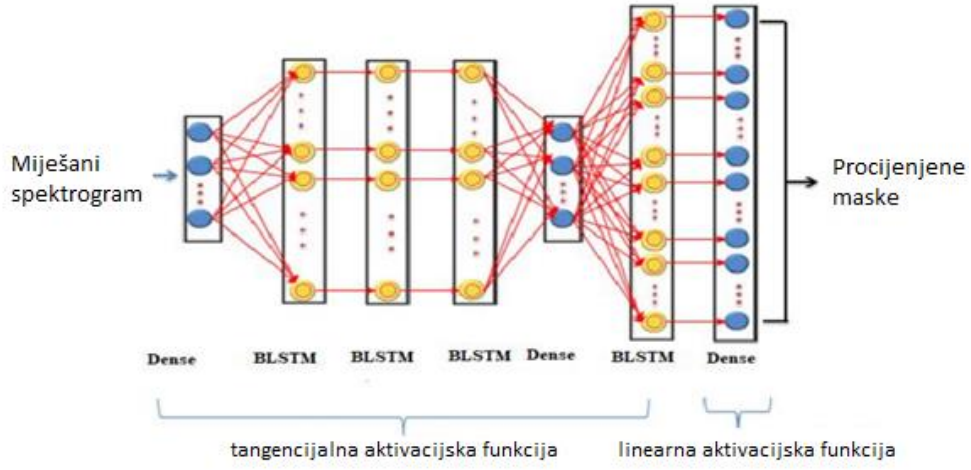
Gdje S_1 i S_2 predstavljaju spektar dva izvora signala u mješovitom signalu u okviru (t) i frekvenciji (f). Simbol (*) označava konjugiranu operaciju, dok R predstavlja realnu komponenta spektra. ORM se razlikuje od IRM-a po prisutnosti koherentnog dijela *R(S(t, f)N*(t, f))*, čija je vrijednost jednaka nuli u IRM. ORM postiže visoku učinkovitost u proces razdvajanja u slučaju visoke korelacije između izvora signala i šuma. Vrijednosti maske ORM kreću se od <-∞, +∞>, što čini proces procjene težim. Dakle, vrijednosti ORM maske određene su pomoću funkcije hiperboličkog tangensa kao u jednadžbi (2-8):[1]

$$ORM(t,f) = K\frac{1-e^{-cy(t,f)}}{1+e^{-cy(t,f)}} \tag{2-8}$$

Gdje je $c = 0.1$ strmost, dok je K jednak 10, ograničava vrijednosti ORM između [-10, +10]. Jednadžba (2-7) je osnovna jednadžba ORM maske.[1]

2.3. Arhitektura mreže

U radu je korištena DRNN_BLSTM arhitektura, prikazana na Slika 2.1., koja se temelji na monofonskom razdvajanju govora. Mreža je obučena za predviđanje maske koja je najbliža referentnim maskama minimiziranjem srednje kvadratne pogreške između predviđene i referentne maske, predviđena maska se zatim koristi za odvajanje. Model se sastoji od dvije pod mreže, prva se mreža koristi za procjenu maski za „muški“ signal, dok se drugi koristi za procjenu maski za „ženski“ signal. Svaka pod mreža sastoji se od tri skrivena sloja BLSTM 128 neurona, iza kojih slijedi gusti sloj 64 neurona i BLSTM sloj sa 512 neurona. Funkcija aktivacije koristi se za obuku modela. [1]

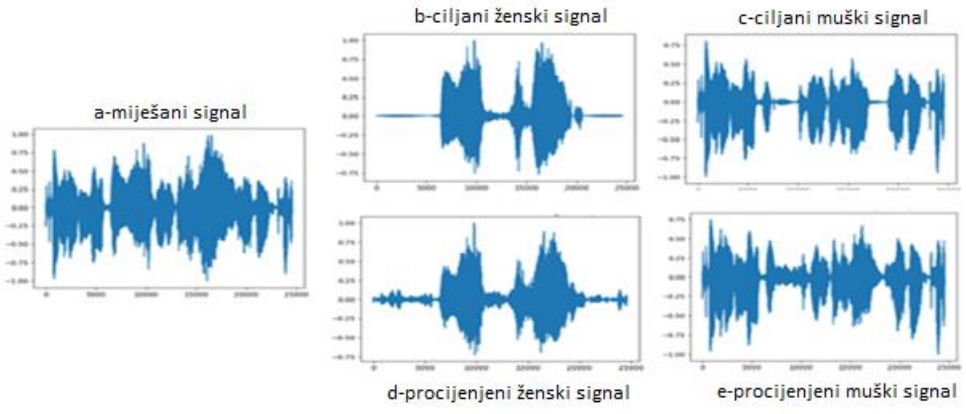


Slika 2.1. Arhitektura mreže korištenog modela.

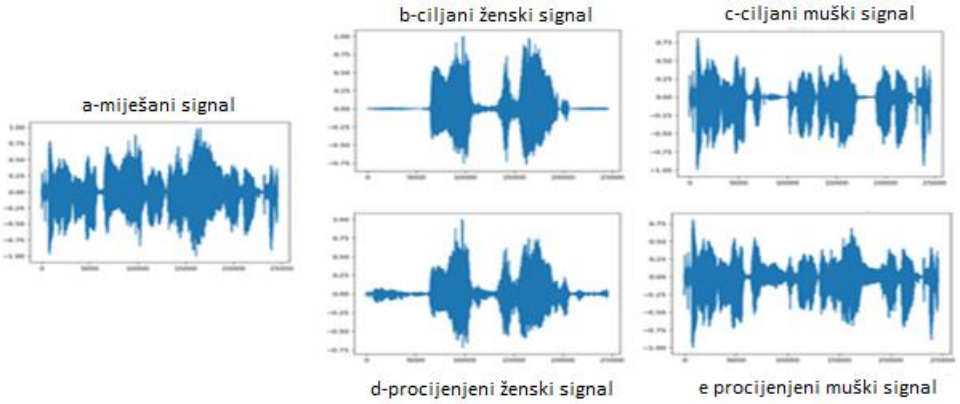
2.4. Rezultati

Prikazani rezultati su prikazani u [1]. Set podataka koji je korišten za treniranje modela sastoji se od 500 audio datoteka sa trajanjem od 3 sekunde. Audio datoteke su također rastavljene na „muški“ i „ženski“ signal, a prikupljene su na internetu i spojene u miješani signal.

Uvježbani model testiran je na 10 mješovitih signala za svaku masku, a rezultati procesa odvajanja razlikuju se ovisno o korištenoj maski. Stoga je učinkovitost mrežnih performansi mjerena korištenjem tri vrste metrika: omjer signala i izobličenja (SDR – *eng.* signal to distortion ratio), omjer signala i smetnje (SIR – *eng.* signal to interference ratio) te omjer signala i artefakta (SAR – *eng.* signal to artifact ratio) u signalu generiranom postupkom razdvajanja. Slika 2.2. prikazuje kombinirani signal i signal izvora, kao i odvojeni signal pomoću IRM maske, a Slika 2.3. prikazuje proces razdvajanja za isti signal pomoću ORM maske. Učinkovitost modela odvajanja procjenjuje se na temelju SDR-a, SIR-a i SAR-a izračunatih korištenjem alata za procjenu BSS-a, Tablica 2.1., prikazuje rezultate ove tri metrike kada se koriste IRM i ORM u zadatku odvajanja. Rezultati odvajanja pokazuju da je proces odvajanja signala korištenjem ORM maske postigao SDR malo veći (0,183dB) nego što je slučaj kada se koristi IRM. Kao i za SIR za 0,198dB je veći kada se koristi IRM, dok je SAR i u IRM i ORM visok, ali razlika između dvije maske je 0,13dB. Učinkovitost odvajanja korištenjem ORM-a je visoka u slučaju visoke korelacije između kombiniranih signala i funkcije gubitka predstavljene krivuljom učenja neuronske mreže korištenjem korijena srednjeg kvadrata. Optimizator propagacije (RMSprop) smanjuje vrijednost funkcije pogreške kako bi se postigla najbolja vrijednost u kojoj je funkcija pogreške niža. Za obje maske postignut je najbolji rezultat.



Slika 2.2. Prikaz originalnog signala i procijenjenog korištenjem IRM maske.



Slika 2.3. Prikaz originalnog signala i procijenjenog korištenjem ORM maske.

Tablica 2.1. Prikaz usporedbe performansi dviju maski.

Maska	SDR	SIR	SAR
IRM	6.164dB	7.189dB	14.020dB
ORM	6.347dB	7.387dB	14.150dB

3. TEHNIKE OBRADE SIGNALA

U ovom poglavlju istražujemo tehnike obrade signala koje se koriste za razdvajanje glasa i instrumenata. Ove tehnike pružaju temelj za mnoge algoritme razdvajanja, nudeći raznolik raspon alata za analizu i manipuliranje audio signalima. Korištenjem ovih metoda, audio inženjeri i istraživači postigli su značajan napredak u izolaciji glasa i instrumenata, olakšavajući poboljšane aplikacije za uređivanje zvuka i odvajanje izvora. Razumijevanjem ovih tehnika možemo steći uvid u temeljne mehanizme uspješnih algoritama razdvajanja. Kroz istraživanje tehnika obrade signala, omogućavamo usavršavanje audio produkcije i pružamo umjetnicima i producentima moćne alate za oblikovanje glazbenih kompozicija.

3.1. Analiza frekvencijske domene

Fourierova transformacija je matematička tehnika koja se intenzivno koristi u analizi frekvencijskog područja. Omogućuje nam da razložimo signal na njegove frekvencijske komponente, dajući nam uvid u različite frekvencijske komponente koje čine signal. Postoji nekoliko verzija Fourierove transformacije, ali najčešće su kontinuirana Fourierova transformacija, diskretna Fourierova transformacija i brza Fourierova transformacija.[2]

Fourierovu transformaciju prvi je uveo Joseph Fourier početkom 19. stoljeća. Od tada je postao temeljni alat u mnogim područjima znanosti i inženjerstva. Koristi se u područjima kao što su fizika, kemija, biologija i ekonomija.[3]

3.1.1. Kontinuirana Fourierova transformacija

Kontinuirana Fourierova transformacija koristi se za pretvaranje signala kontinuiranog vremena u njegovu reprezentaciju u frekvencijskoj domeni. Izračunava Fourierove koeficijente signala, koji predstavljaju amplitudu i fazu svake frekvencijske komponente u signalu. Kontinuirana Fourierova transformacija korisna je u aplikacijama kao što je obrada zvuka, gdje je signal kontinuiran i treba ga analizirati u frekvencijskoj domeni.

Na primjer, u audio obradi, Fourierova transformacija može se koristiti za analizu frekvencijskih komponenti glazbenog signala. To može biti korisno u zadacima kao što je razdvajanje glasa od instrumenata, otkrivanje prisutnosti određenih instrumenata u glazbi ili prepoznavanje nota koje se sviraju.[3]

3.1.2. Diskretna Fourierova transformacija

Diskretna Fourierova transformacija slična je kontinuiranoj Fourierovoj transformaciji, ali se koristi za pretvaranje diskretnog vremenskog signala u njegovu reprezentaciju frekvencijske domene. Za razliku od kontinuirane Fourierove transformacije, diskretna Fourierova transformacija izračunava konačan skup Fourierovih koeficijenata koji predstavljaju diskretne komponente frekvencije. Diskretna Fourierova transformacija obično se koristi u aplikacijama za digitalnu obradu signala.

Diskretna Fourierova transformacija naširoko se koristi u mnogim područjima digitalne obrade signala, kao što je obrada slike i prepoznavanje govora. Na primjer, u obradi slike, Fourierova transformacija može se koristiti za analizu frekvencijskih komponenti slike. To može biti korisno u zadacima kao što je kompresija slike ili poboljšanje slike.[3]

3.1.3. Short-time Fourier transform (STFT)

STFT je Fourierova transformacija koja se koristi za određivanje sinusne frekvencije i faznog sadržaja lokalnih dijelova signala kako se mijenja tijekom vremena. U praksi, postupak za izračunavanje STFT-a je dijeljenje dužeg vremenskog signala u kraće segmente jednake duljine i zatim izračunavanje Fourierove transformacije zasebno za svaki kraći segment. Ovo otkriva Fourierov spektar na svakom kraćem segmentu. Tada se obično iscrtavaju promjenjivi spektri kao funkcija vremena, poznati kao spektrogram, kao što se obično koristi u softverski definiranim radijskim (SDR) spektralnim prikazima.[4]

3.2. Vremensko-frekvencijsko maskiranje

Vremensko-frekvencijsko maskiranje jedna je od tehnika koja se koristi za odvajanje instrumenata od glasa u pjesmi. Iskorištava vremenski promjenjivi spektralni sadržaj audio signala za selektivno pojačavanje ili stišavanje određenih instrumenata dok istovremeno poboljšava željenu glasovnu komponentu.

Temeljna ideja iza vremensko-frekvencijskog maskiranja je predstavljanje miješanog audio signala u vremensko-frekvencijskoj domeni, kao što je korištenje spektrograma, a zatim primjena maske koja modificira amplitudu različitih frekvencijskih komponenti u svakom vremenskom

okviru. Manipuliranjem maske može se kontrolirati doprinos određenih instrumenata i pojačati izolacija vokalne komponente.

Za izradu maske koriste se informacije iz miješanog audio signala i ciljane glasovne komponente. Za procjenu maske mogu se koristiti različite metode, uključujući statističku analizu, strojno učenje i tehnike dubokog učenja.

U statističkoj analizi, maska se dobiva usporedbom spektralnih karakteristika miješanog audio signala sa spektralnom karakteristikom glasovne komponente u frekvencijskoj domeni. Ova usporedba pomaže identificirati regije u kojima dominira glas i područja u kojima su drugi instrumenti jače prisutni. Na temelju ove analize, maska se može stvoriti za ublažavanje područja u kojima dominiraju instrumenti, učinkovito ih odvajajući od glasovne komponente.

Strojno učenje i tehnike dubokog učenja također su uspješno primijenjene na vremensko-frekvencijsko maskiranje. Ove metode uključuju modele trenirane na velikim skupovima podataka koji sadrže primjere miješanog zvuka i njihove odgovarajuće komponente glasa i instrumenata. Modeli uče procijeniti masku koja najbolje odvaja glas od instrumenata koristeći uzorke u spektralnim podacima.

Nakon što se dobije maska, ona se primjenjuje na mješoviti audio signal u vremensko-frekvencijskoj domeni, mijenjajući amplitudu specifičnih frekvencijskih komponenti u svakom vremenskom okviru. Pojačavanjem ili stišavanjem područja u kojima dominiraju instrumenti, a ostavljajući glasovnu komponentu relativno netaknutom.[2]

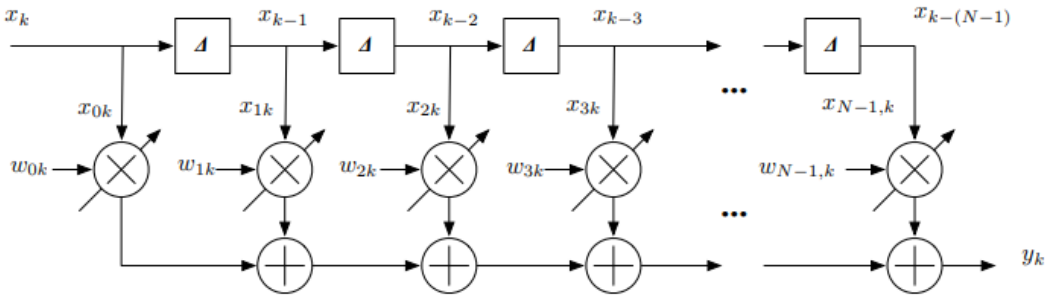
3.3. Adaptivno filtriranje

Konvencionalni filteri sa statičkim koeficijentima dizajnirani su kako bi odradili jednu zadaću poput prigušivanja svih frekvencija nakon nekog praga dok je adaptivno filtriranje dizajnirano s namjerom da se ti koeficijenti mogu dinamički mijenjati tijekom vremena. Koriste se kako bi filter mogao što bolje obaviti specifični zadatak, također se koristi kada se operirajuće okruženje mijenja tijekom vremena.

Adaptivno filtriranje je tehnika za odvajanje instrumenata od glasa u pjesmi. Kontinuiranom prilagodbom koeficijenata filtra na temelju ulaznog signala i željenog ciljnog signala, adaptivno filtriranje pojačava ili stišava komponente instrumenta dok poboljšava vokalnu komponentu. U kombinaciji s drugim tehnikama, kao što je vremensko-frekvencijsko maskiranje, adaptivno

filtriranje može dodatno poboljšati performanse razdvajanja, nudeći poboljšane mogućnosti za glazbenu produkciju i audio inženjering.

3.3.1. Adaptivni linearni kombinator



Slika 3.1. Struktura FIR (eng. finite impulse response) filtera korištena u adaptivnom filtriranju.

Izlaz adaptivnog linearnog kombinatora može se opisati jednadžbom:

$$Y_k = \sum_{l=0}^{L-1} X_{lk} W_{lk} \tag{3-1}$$

U kojoj $\{W_{lk}\}$ predstavlja varijablu težine adaptivnog filtera i $\{X_{lk}\}$ predstavlja ulazni signal. Ova jednadžba se može primijeniti na bilo koji sistem za koji su izlazi dobiveni računanjem linearne kombinacije izlaznog niza. Najčešća implementacija adaptivnog procesora je na FIR filteru sa dinamičnim koeficijentima. Ta struktura je prikazana u Slika 3.1. s notacijskim konvencijama koje se obično koriste za opisivanje adaptivnih sustava. Napominjemo da su koeficijenti filtera (ili "težine") sada varijable s nazivom varijable W_{lk}, gdje prvi indeks identificira pojedinačni koeficijent filtera koji se razmatra a drugi koeficijent označava vremenski indeks. Dijagonalne strelice kroz simbol množenja opet pokazuju da koeficijenti filtra variraju tijekom vremena.

Važna varijabla prikazana na slici je vektor promatranja, koji je po definiciji skup signalnih ulaza u varijable težine. Ako je adaptivni procesor u obliku FIR filtar, komponente vektora promatranja X_{lk} jednake su odgođenim verzijama izvorni unos gdje je $X_{lk} = X_{k-l}$.

Često je zgodno predstaviti ulaze $\{ X_{lk} \}$ i varijabilne koeficijente filtera $\{ W_{lk} \}$ kao vektori stupaca:

$$X_k = \begin{bmatrix} x_{0k} \\ x_{1k} \\ x_{2k} \\ \vdots \\ x_{L-1,k} \end{bmatrix} \text{ i } W_k = \begin{bmatrix} w_{0k} \\ w_{1k} \\ w_{2k} \\ \vdots \\ w_{L-1,k} \end{bmatrix} \tag{3-2}$$

Koristeći notaciju linearne algebre lako prikazati da

$$Y_k = \sum_{l=0}^{L-1} X_{lk} W_{lk} = X^T_k W_k = W^T_k X_k \tag{3-3}$$

Gdje T predstavlja matrični operator transpozicije.

4. ANALIZA I OBRADA ZNAČAJKI

U ovome poglavlju raspraviti ćemo o metodama za izdvajanje i analizu značajki koje su korisne u identificiranju i razlikovanju različitih audio komponenti. Također istražujemo tradicionalne i moderne pristupe, uključujući strojno učenje i tehnike dubokog učenja, koje koriste principe obrade signala za poboljšane performanse odvajanja. Izdvajanje i analiza značajki igraju ključnu ulogu u odvajanju instrumenata od vokala u pjesmi. Ove tehnike omogućuju prepoznavanje i razlikovanje audio karakteristika koje su prepoznatljive za specifične instrumente i ljudski glas.

Ekstrakcija značajki uključuje ekstrakciju važnih informacija iz audio signala koji se mogu koristiti za razlikovanje različitih izvora zvuka. Ove značajke hvataju različite aspekte kao što su visina, harmonici, frekvencija i vremenska dinamika, jer su bitni u prepoznavanju i odvajanju instrumenata od glasa.

Jedna od temeljnih značajki koja se koristi u razdvajanju instrumenta i vokala je frekvencija. Frekvencija tona usko je povezana s glazbenim notama. Instrumenti i glas imaju jedinstvene uzorke frekvencija, a analizom sadržaja tona može se razlikovati između različitih izvora u audio signalu.

Harmonici su još jedna ključna značajka koja se koristi u razdvajanju. Instrumenti kao što su gitare, klaviri i violine proizvode harmonijske prizvuke koji slijede specifične obrasce na temelju njihovih fizičkih svojstava. Analiza harmonijske strukture audio signala omogućuje prepoznavanje uzoraka specifičnih za instrumente i razlikovanje instrumentalnih komponenti iz glasa.

Pristup strojnom učenju, uključujući algoritme klasifikacije i klasteriranja, obično se koristi za analizu značajki. Uvježbavanjem modela na označenim skupovima podataka koji sadrže primjere miješanog zvuka s poznatim instrumentima i glasovnim komponentama, ti modeli mogu se istrenirati i napraviti predviđanja o izvoru svakog instrumenta ili vokala.

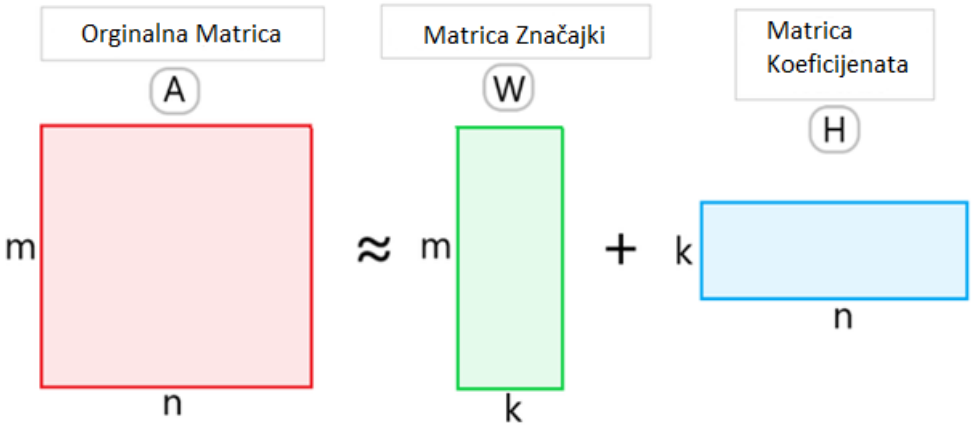
5. PRISTUPI STROJNOM UČENJU

U ovom poglavlju ćemo se više upoznati sa algoritmima i metodama strojnog učenja koje se koriste za razdvajanje instrumenata od glasa. Napretkom tehnologije i računalne snage omogućeno je korištenje algoritama za stvaranje modela koji se treniraju na skupu podataka kako bi kasnije mogli koristiti taj model za primjenu na do tada modelu ne viđenim podacima i tako nam omogućavaju samo razdvajanje, stišavanje ili pojačavanje instrumenata ili vokala.

5.1. NMF (Non-Negative Matrix Factorization)

Faktorizacija ne negativne matrice je metoda faktorizacije matrice u kojoj ograničavamo da matrice budu ne negativne. Kako bismo razumjeli NMF, trebali bismo razjasniti temeljnu intuiciju između faktorizacije matrice.

Za matricu A dimenzija m x n, gdje je svaki element ≥ 0, NMF je može rastaviti na dvije matrice W i H dimenzija m x k odnosno k x n, a ove dvije matrice sadrže samo ne negativne elemente.[5]



Slika 5.1. Prikaz rastavljanja matrica.

Cilj NMF-a je smanjenje dimenzionalnosti i ekstrakcija značajki. Dakle, kada postavimo nižu dimenziju kao k, cilj NMF-a je pronaći dvije matrice W ∈ Rm×k i H ∈ Rn×k koje imaju samo ne negativne elemente. (Kao što je prikazano na Slika 5.1.)

Stoga, korištenjem NMF-a možemo dobiti faktorizirane matrice koje imaju znatno niže dimenzije od onih matrica proizvoda. Intuitivno, NMF pretpostavlja da je izvorni unos napravljen od skupa skrivenih značajki, predstavljenih svakim stupcem W matrice, a svaki stupac u H matrici

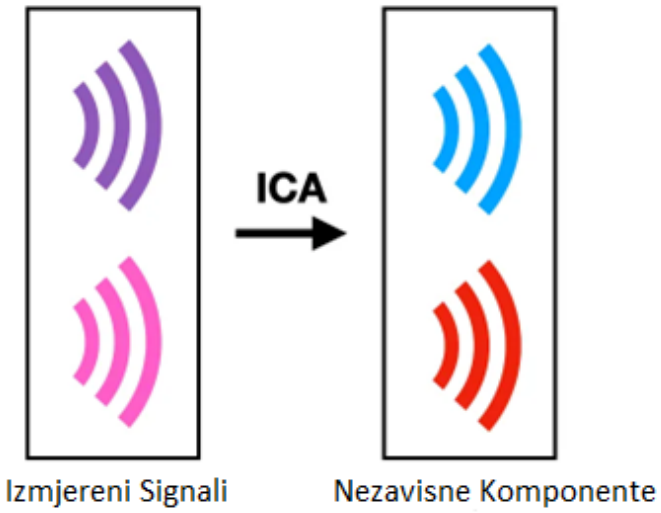
predstavlja 'koordinate podatkovne točke' u matrici W. Jednostavno rečeno, sadrži težine povezane s matricom W.

Dakle, svaka podatkovna točka koja je predstavljena kao stupac u A, može se aproksimirati aditivnom kombinacijom ne negativnih vektora, koji su predstavljeni kao stupci u W.[5]

5.2. ICA (Independent Component Analysis)

U obradi signala, analiza nezavisnih komponenti (ICA) računalna je metoda za odvajanje miješanog signala u dodatne pod komponente. To se postiže pretpostavkom da je najviše jedna pod komponenta Gaussova i da su pod komponente statistički neovisne jedna o drugoj. ICA je poseban slučaj slijepog odvajanja. Uobičajeni primjer primjene je "problem koktel zabave" slušanja govora jedne osobe u bučnoj prostoriji.[6]

Ovaj problem se lako rješava analizom nezavisnih komponenti (ICA) koja transformira skup vektora u maksimalno nezavisan skup. Vraćajući se na naš “Problem s koktel zabavama”, ICA će pretvoriti dvije miješane audio snimke (predstavljene ljubičastim i ružičastim valnim oblicima ispod) u dvije ne miješane snimke svakog pojedinačnog govornika (predstavljene plavim i crvenim valnim oblicima ispod). Primijetite da je broj ulaza i izlaza isti, a budući da su izlazi međusobno neovisni, ne postoji očit način za ispuštanje komponenti.[6]



Slika 5.2. Jednostavan prikaz signala prije i nakon primjene ICA algoritma.

U ICA postoje dvije ključne pretpostavke. Skrivene neovisne komponente koje pokušavamo otkriti moraju biti jedna, statistički neovisna, i druga, koje nije Gaussova. Semantički, pod neovisnim mislim na informacije o x ne daju vam informacije o y i obrnuto. Matematički, ovo znači

$$p(x, y) = p(x)p(y) \tag{5-1}$$

Gdje p(x) predstavlja distribuciju vjerojatnosti x. p(x, y) predstavlja zajedničku distribuciju x i y. Ne-Gaussova pretpostavka jednostavno znači da neovisne komponente imaju distribucije koje nisu Gaussove, što znači da ne izgleda kao zvonasta krivulja.[6]

5.2.1. Prednosti ICA analize

Sposobnost odvajanja mješovitih signala: ICA je moćan alat za odvajanje mješovitih signala u njihove neovisne komponente. Ovo je korisno u raznim aplikacijama, kao što je obrada signala, analiza slike i kompresija podataka.

Ne parametarski pristup: ICA je ne parametarski pristup, što znači da ne zahtijeva pretpostavke o temeljnoj distribuciji vjerojatnosti podataka.

Učenje bez nadzora: ICA je tehnika učenja bez nadzora, što znači da se može primijeniti na podatke bez potrebe za označenim primjerima. To ga čini korisnim u situacijama kada označeni podaci nisu dostupni.

Ekstrakcija značajki: ICA se može koristiti za ekstrakciju značajki, što znači da može identificirati važne značajke u podacima koje se mogu koristiti za druge zadatke, kao što je klasifikacija.[7]

5.2.2. Nedostaci ICA analize

Ne-Gaussova pretpostavka: ICA pretpostavlja da temeljni izvori nisu Gaussovi, što ne mora uvijek biti točno. Ako su temeljni izvori Gaussovi, ICA možda neće biti učinkovit.

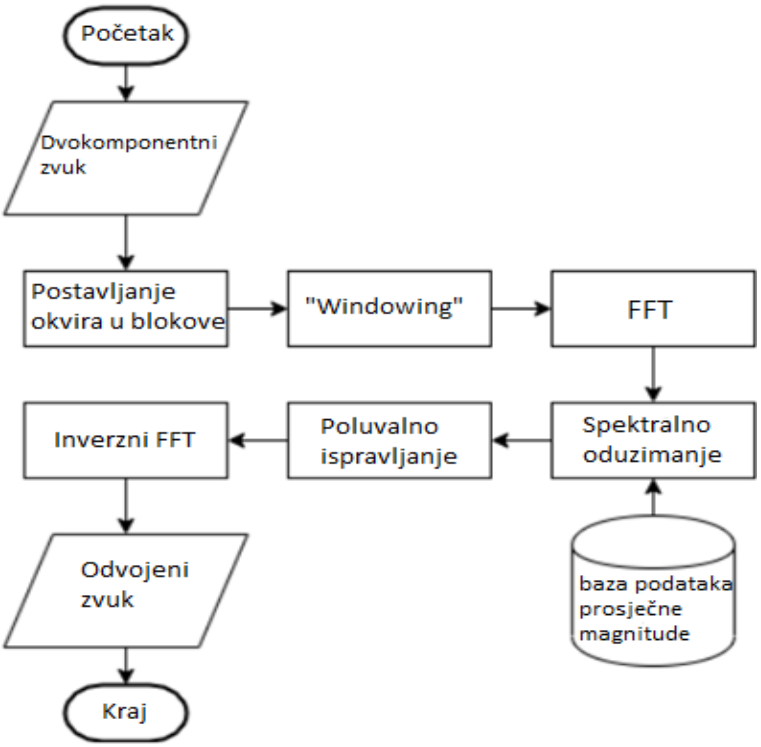
Pretpostavka linearnog miješanja: ICA pretpostavlja da se izvori miješaju linearno, što ne mora uvijek biti slučaj. Ako se izvori miješaju nelinearno, ICA možda neće biti učinkovit.

Računalno skupo: ICA može biti računalno skupo, posebno za velike skupove podataka. To može otežati primjenu ICA-e na probleme iz stvarnog svijeta.

Problemi s konvergencijom: ICA može imati problema s konvergencijom, što znači da možda neće uvijek moći pronaći rješenje. To može biti problem za složene skupove podataka s mnogo izvora.[7]

5.3. Spektralno oduzimanje

Postoji nekoliko koraka za odvajanje metodom spektralnog oduzimanja. Koraci potrebni za odvajanje započinje izradom modela prosječne magnitude spektra za svaki pojedinačni zvuk i pohranom u bazu podataka. Zatim su okviri zvučnog signala pretvoreni iz vremenske domene u frekvencijsku domenu. Spektar svakog okvira oduzet je s model prosječne magnitude spektra šuma, a audio signal u frekvencijskoj domeni bio je rekonstruiran u vremenskoj domeni. Slika 5.3. prikazuje korake metode.[8]



Slika 5.3. Koraci spektralnog oduzimanja.

5.4. Wienerov filter

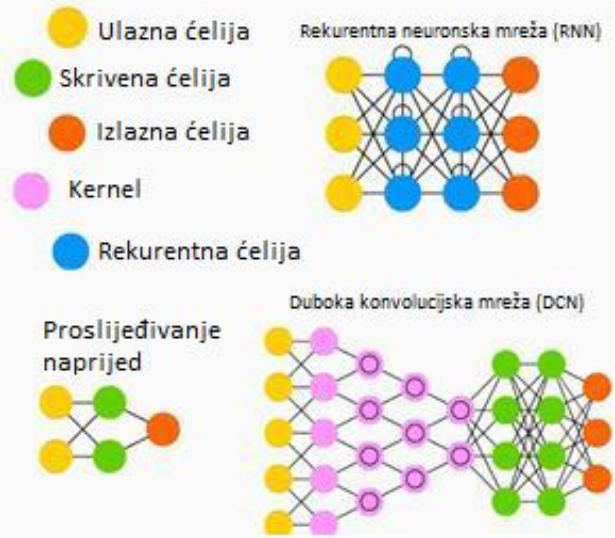
Wienerovo filtriranje jedan je od najčešće korištenih alata u obradi signala, posebno za uklanjanje šuma i odvajanje komponenata. U kontekstu zvuka, gdje signali nisu stacionarni, ali kratkoročno jesu, obično se primjenjuje u vremensko-frekvencijskoj domeni putem kratkotrajne Fourierove transformacije (STFT). Koeficijenti filtera, koji nisu negativni, izračunavaju se neovisno u svakom vremensko-frekvencijskom pretincu od varijanci signala koji će biti odvojeni bez vođenja računa o dosljednosti dobivenog skupa STFT koeficijenata, tj. činjenicu da oni zapravo odgovaraju na STFT signala u vremenskoj domeni. Zajedno s netočnim procjenama varijanci signala, ova neovisna obrada je jedan od razloga iza prisutnosti preostalih smetnji ili glazbenih šumova u odvojenim signalima.

6. DUBOKO STROJNO UČENJE

Duboka neuronska mreža (DNN) je ANN (eng. Artificial neural network) s više skrivenih slojeva između ulaznog i izlaznog sloja. Slično plitkim ANN-ovima, DNN-ovi mogu modelirati složene nelinearne odnose.

Glavna svrha neuronske mreže je primanje skupa ulaza, izvođenje progresivno složenih izračuna na njima i davanje rezultata za rješavanje problema stvarnog svijeta poput klasifikacije.

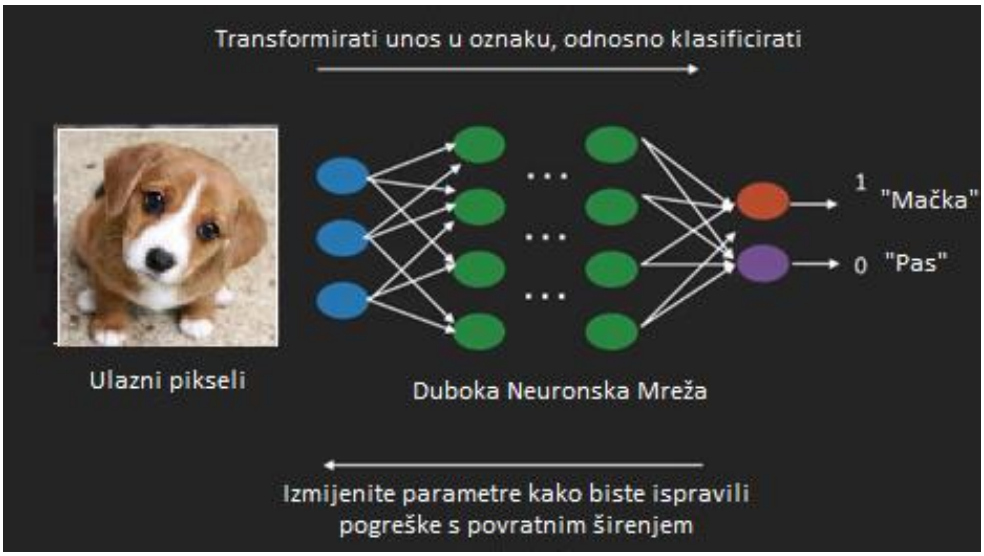
Imamo ulaz, izlaz i tok sekvencijalnih podataka u dubokoj mreži.[9]



Slika 6.1. Pojednostavljeni prikaz DNN-a.

Neuronske mreže naširoko se koriste u nadziranom učenju. Te se mreže temelje na skupu međusobno povezanih slojeva. U dubokom učenju broj skrivenih slojeva, uglavnom nelinearnih, može biti velik; recimo oko 1000 slojeva. Duboko učeni modeli daju mnogo bolje rezultate od normalnih strojno učenih mreža. Uglavnom koristimo metodu gradijentnog spuštanja za optimizaciju mreže i minimiziranje funkcije gubitaka. Možemo koristiti Imagenet, spremište milijuna digitalnih slika za klasificiranje skupa podataka u kategorije kao što su mačke i psi. Duboko učene mreže se sve više koriste za dinamičke slike osim statičkih te za vremenske serije i analizu teksta. Obuka skupova podataka čini važan dio modela dubokog učenja. Osim toga, povratna propagacija je glavni algoritam u obuci duboko učenih modela. Duboko učenje se bavi obučavanjem velikih neuronskih mreža sa složenim transformacijama ulaza i izlaza. Jedan primjer dubokog učenja je mapiranje fotografije s imenom osobe(a) na fotografiji kao što to rade na društvenim mrežama, a opisivanje slike frazom još je jedna nedavna primjena dubokog učenja.[9]

Neuronske mreže su funkcije koje imaju ulaze kao što su x1, x2, x3... koji se transformiraju u izlaze kao što su z1, z2, z3 i tako dalje u dvije (plitke mreže) ili nekoliko posrednih operacija koje se također nazivaju slojevima (duboke mreže). Težine i pristranosti mijenjaju se od sloja do sloja. 'w' i 'v' su težine ili sinapse slojeva neuronskih mreža. Najbolji slučaj korištenja dubinskog učenja je problem nadziranog učenja. Ovdje imamo veliki skup ulaznih podataka sa željenim skupom izlaza.[9]



Slika 6.2. Prikaz arhitekture mreže za klasifikaciju psa ili mačke.

Ovdje primjenjujemo algoritam povratne propagacije kako bismo dobili ispravno predviđanje izlaza. Najosnovniji skup podataka dubinskog učenja je MNIST (eng. Modified National Institute of Standards and Technology), skup rukom pisanih znamenki. Možemo duboko uvježbati konvolucijsku neuronsku mrežu s kerasom kako bismo klasificirali slike rukom pisanih znamenki iz ovog skupa podataka. Okidanje ili aktivacija klasifikatora neuronske mreže daje rezultat. Na primjer, da bismo klasificirali pacijente kao bolesne i zdrave, uzimamo u obzir parametre kao što su visina, težina i tjelesna temperatura, krvni tlak itd. Visok rezultat znači da je pacijent bolestan, a nizak rezultat znači da je zdrav. [9]

Svaki čvor u izlaznim i skrivenim slojevima ima svoje vlastite klasifikatore. Ulazni sloj prima ulaze i prosljeđuje svoje rezultate sljedećem skrivenom sloju za daljnju aktivaciju i to se nastavlja sve dok se ne postigne izlaz. Ovaj napredak od ulaza do izlaza slijeva nadesno u smjeru naprijed naziva se širenje naprijed. CAP (eng. Credit assignment path) u neuronskoj mreži je niz transformacija počevši od ulaza do izlaza. CAP-ovi razrađuju vjerojatne uzročne veze između ulaza i izlaza. Dubina CAP-a za danu neuronsku mrežu s naprednim prijenosom ili CAP dubina je broj skrivenih slojeva plus jedan jer je uključen izlazni sloj. Za rekurentne neuronske mreže, gdje se signal može širiti kroz sloj nekoliko puta, CAP dubina može biti potencijalno neograničena.[9]

6.1. CNN (Convolutional Neural Networks)

Ako povećamo broj slojeva u neuronskoj mreži kako bismo je učinili dubljom, to povećava složenost mreže i omogućuje nam modeliranje funkcija koje su kompliciranije. Međutim, broj težina i pristranosti eksponencijalno će rasti. Zapravo, učenje tako teških problema može postati nemoguće za normalne neuronske mreže. To dovodi do rješenja, konvolucijske neuronske mreže (CNN). CNN-ovi se intenzivno koriste u računalnom vidu; primijenjeni su i u akustičkom modeliranju za automatsko prepoznavanje govora.

Ideja iza konvolucijskih neuronskih mreža je ideja "pokretnog filtra" koji prolazi kroz sliku. Ovaj pokretni filtar, ili konvolucija, primjenjuje se na određeno susjedstvo čvorova koji na primjer mogu biti pikseli, gdje je primijenjeni filtar 0,5 x vrijednost čvora.

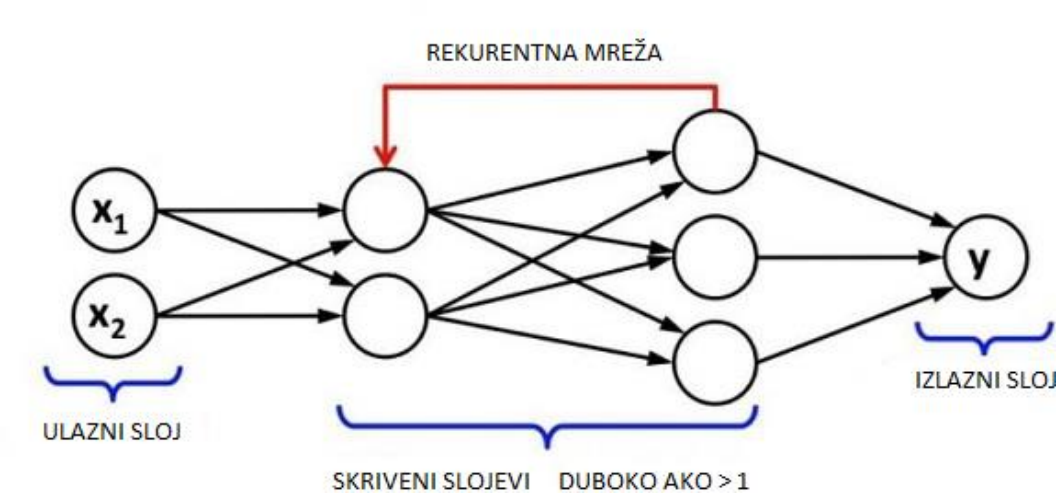
Poznati istraživač Yann LeCun bio je pionir konvolucijskih neuronskih mreža. Facebook kao softver za prepoznavanje lica koristi ove mreže. CNN je bio pravo rješenje za projekte strojnog vida. Konvolucijska mreža ima mnogo slojeva. U Imagenet izazovu, stroj je uspio pobijediti čovjeka u prepoznavanju predmeta 2015.

Ukratko, konvolucijske neuronske mreže (CNN) su višeslojne neuronske mreže. Ponekad imaju do 17 ili više slojeva i pretpostavljaju da su ulazni podaci slike.[9]

6.2. RNN (Recurrent Neural Networks)

Rekurentne neuronske mreže (RNN) su neuronske mreže u kojima podaci mogu teći u bilo kojem smjeru. Te se mreže koriste za aplikacije poput modeliranja jezika ili obrade prirodnog jezika. Osnovni koncept koji leži u RNN-u je korištenje sekvencijalnih informacija. U normalnoj neuronskoj mreži pretpostavlja se da su svi ulazi i izlazi neovisni jedni o drugima. Ako želimo predvidjeti sljedeću riječ u rečenici, moramo znati koje su riječi bile prije nje.

RNN-ovi se nazivaju rekurentnim jer ponavljaju isti zadatak za svaki element niza, a izlaz se temelji na prethodnim izračunima. Stoga se može reći da RNN-ovi imaju "memoriju" koja bilježi informacije o onome što je prethodno izračunato. U teoriji, RNN-ovi mogu koristiti informacije u vrlo dugim sekvencama, ali u stvarnosti mogu pogledati unatrag samo nekoliko koraka.[9]



Slika 6.3. Prikaz jednostavne Rekurentne neuronske mreže.

Mreže dugog kratkoročnog pamćenja (LSTM) najčešće su korištene RNN mreže.

Zajedno s konvolucijskim neuronskim mrežama, RNN-ovi su korišteni kao dio modela za generiranje opisa za neoznačene slike. Prilično je nevjerojatno koliko dobro ovo djeluje.[9]

6.2.1. LSTM (Long short-term memory)

LSTM je RNN mreža, čiji je cilj riješiti problem nestajanja gradijenta koji je prisutan u tradicionalnim RNN-ovima. Njegova relativna neosjetljivost na duljinu praznine njegova je prednost u odnosu na druge RNN-ove, skrivene Markovljeve modele i druge metode učenja nizova. Cilj mu je osigurati kratkoročno pamćenje za RNN koje može trajati tisuće vremenskih koraka, dakle "dugo kratkoročno pamćenje". Primjenjiv je na klasifikaciju, obradu i predviđanje podataka na temelju vremenskih nizova, kao što su rukopis, prepoznavanje govora, strojno prevođenje, otkrivanje govorne aktivnosti, kontrola robota , video igre, i zdravstvena skrb.[10]

Uobičajena LSTM jedinica sastoji se od ćelije, ulaznih vrata, izlaznih vrata i zaboravljenih vrata. Stanica pamti vrijednosti u proizvoljnim vremenskim intervalima, a tri vrata reguliraju protok informacija u ćeliju i iz nje. Vrata zaborava odlučuju koje će informacije odbaciti iz prethodnog stanja dodjeljivanjem prethodnom stanju, u usporedbi s trenutnim unosom, vrijednosti između 0 i 1; vrijednost 1 znači zadržati informacije, a vrijednost 0 znači odbaci ga. Koristeći isti sustav kao

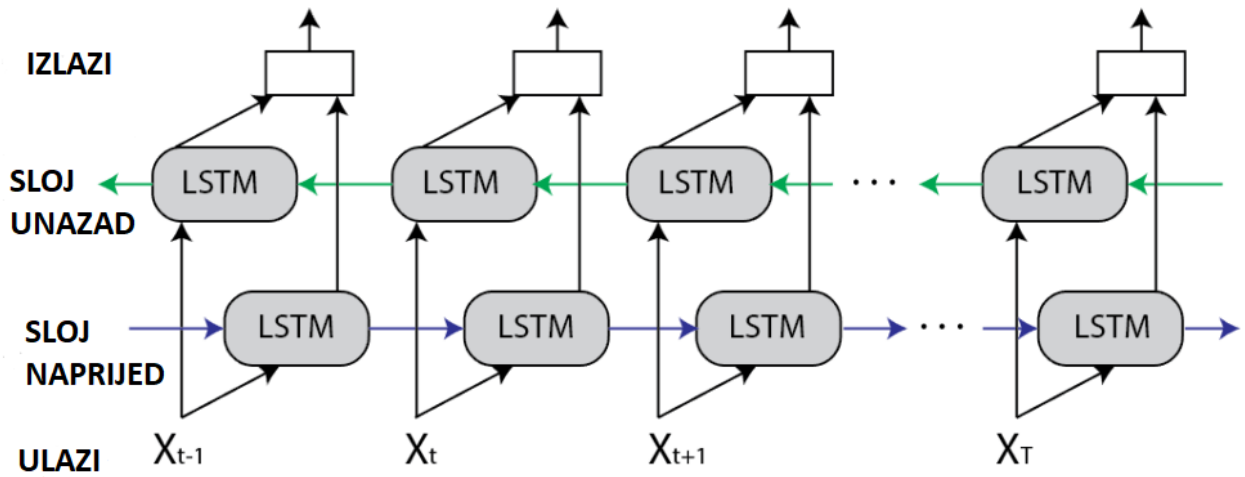
i zaboravljena vrata, ulazna vrata odlučuju koje dijelove novih informacija pohraniti u trenutno stanje. Izlazna vrata kontroliraju koje dijelove informacija u trenutnom stanju treba ispisati dodjeljivanjem vrijednosti od 0 do 1, uzimajući u obzir prethodna i trenutna stanja. Selektivno ispisivanje relevantnih informacija iz trenutnog stanja omogućuje LSTM mreži održavanje korisnih, dugoročnih ovisnosti za izradu predviđanja, kako u sadašnjim tako i u budućim vremenskim koracima.[10]

6.2.2. Bi-LSTM (Bi-directional long short term memory)

Dvosmjerne rekurentne neuronske mreže zapravo samo spajaju dva neovisna RNN-a. Ova struktura omogućuje mrežama da imaju informacije unatrag i unaprijed o nizu u svakom vremenskom koraku.[11]

Dvosmjerni LSTM (BiLSTM) je rekurentna neuronska mreža koja se prvenstveno koristi za obradu prirodnog jezika. Za razliku od standardnog LSTM-a, unos teče u oba smjera i sposoban je koristiti informacije s obje strane. To je također moćan alat za modeliranje sekvencijalnih ovisnosti između riječi i fraza u oba smjera niza.[12]

Ukratko, BiLSTM dodaje još jedan LSTM sloj, koji mijenja smjer protoka informacija. Ukratko, to znači da ulazna sekvenca teče unatrag u dodatnom LSTM sloju. Zatim kombiniramo izlaze iz oba LSTM sloja na nekoliko načina, kao što su prosjek, zbroj, množenje ili ulančavanje.[12]



Slika 6.4. Prikaz Bi-LSTM mreže.

7. PRIMJERI IZ STVARNOG SVIJETA

U ovome poglavlju biti će predstavljen primjer primjene razdvajanja vokala i instrumenata koji se koristi u stvarnom svijetu.

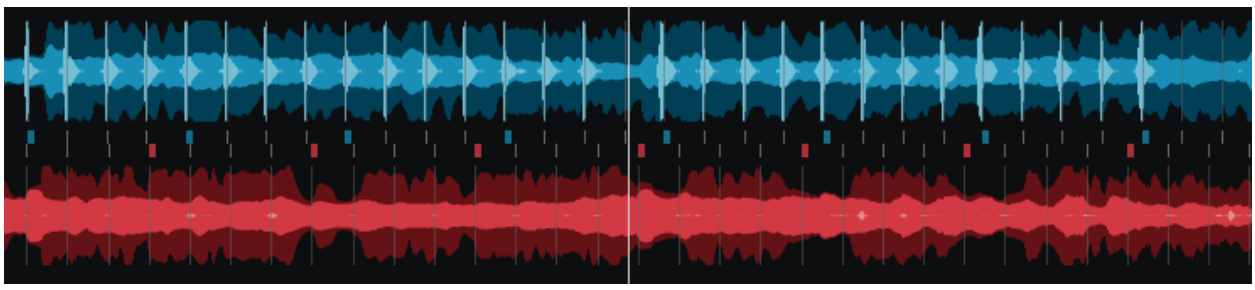
7.1. VirtualDJ

Program VirtualDJ koji koriste DJ-evi prilikom nastupa je jedan od niza programa sa implementiranom funkcijom razdvajanja. Jedna od prednosti ovog programa je svakako mogućnost razdvajanja ne samo instrumenata i vokala nego i mogućnost razdvajanja određenih segmenata pjesme poput basa i kick-a.

Slika 7.1. Prikaz „Stems“ izbornika unutar VirtualDJ programa.

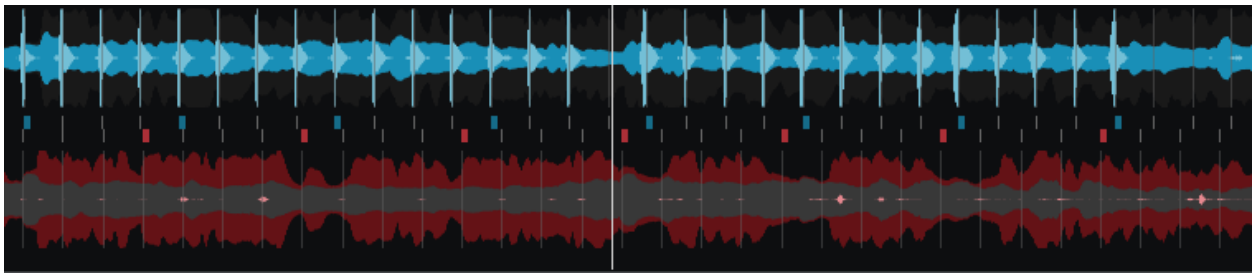
Slika 7.1. Prikazuje kako izgleda „Stems“ izbornik te se može vidjeti kako je dostupno razdvajanje ne samo vokala i instrumenata nego se još dodatno mogu izdvojiti bas, kick i HiHat.

Program VirtualDJ koristi naprednu matematiku u kombinaciji sa naprednom umjetnom inteligencijom kako bi postigli što bolje rezultate razdvajanja. [13]
Usprkos korištenju naprednih algoritama koji zahtijevaju veliku računalnu snagu funkcija Stems će raditi i na računalu slabije računalne snage ali sa lošijom kvalitetom ako se prethodno nisu pripremili, odnosno, na slabijim računalima je potrebna priprema kako bi Stems funkcionalnost radila u punoj kvaliteti prilikom nastupa uživo.[14]



Slika 7.2. Prikaz valnog oblika dvije pjesme unutar VirtualDJ programa

Slika 7.2. Prikazuje valni oblik dvije pjesme koje nisu iste i kojima su „uključene“ sve komponente.



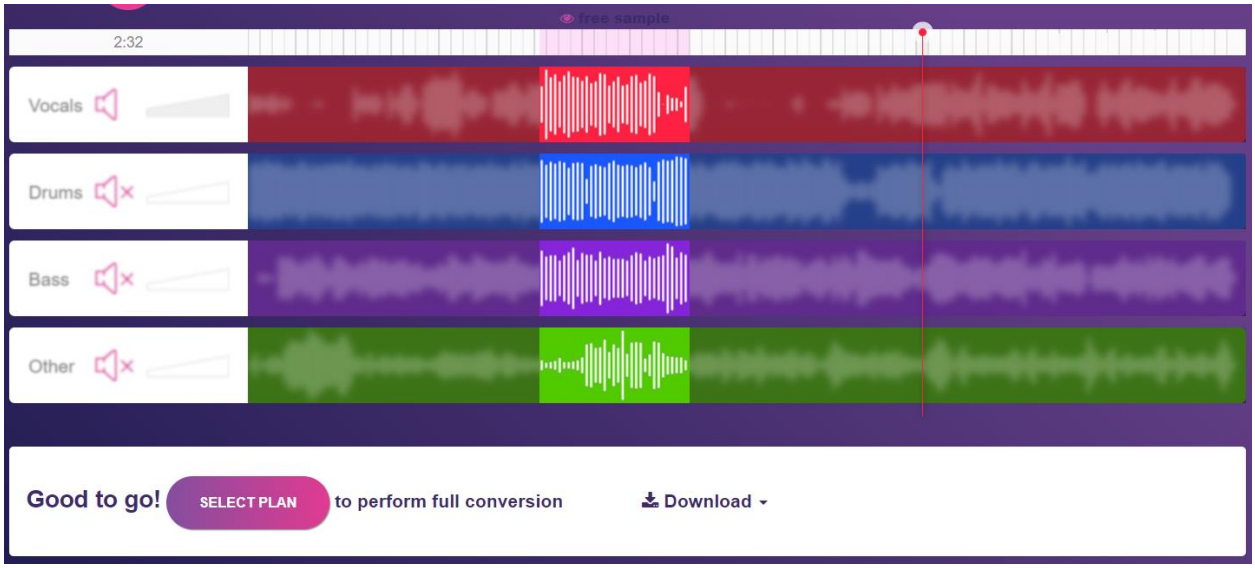
Slika 7.3. Prikaz valnog oblika dvije pjesme unutar VirtualDJ programa.

Na Slika 7.3. možemo vidjeti iste dvije pjesme prikazane na Slika 7.2. ali pjesmi sa plavim valnim oblikom je „isključen“ vokal a pjesmi sa crvenim valnim oblikom je isključen instrumental.

7.2. PhonicMind

PhonicMind je online stranica za obradu pjesme te je specijalizirana za izdvajanje vokala i instrumenata iz pjesme. Koristi umjetnu inteligenciju i tehnike strojnog učenja za odvajanje vokala od instrumenata.

U usporedbi sa VirtualDj-om PhonicMind je više prilagođen za korisnika i lakša je za korištenje, ali glavni nedostatak PhnonicMind stranice je što su plaćena usluga, odnosno kako bi mogli pristupiti izdvojenim komponentama u cijelosti mora se platiti, dok je VirtualDJ besplatan program ako ne koristite DJ kontroler.

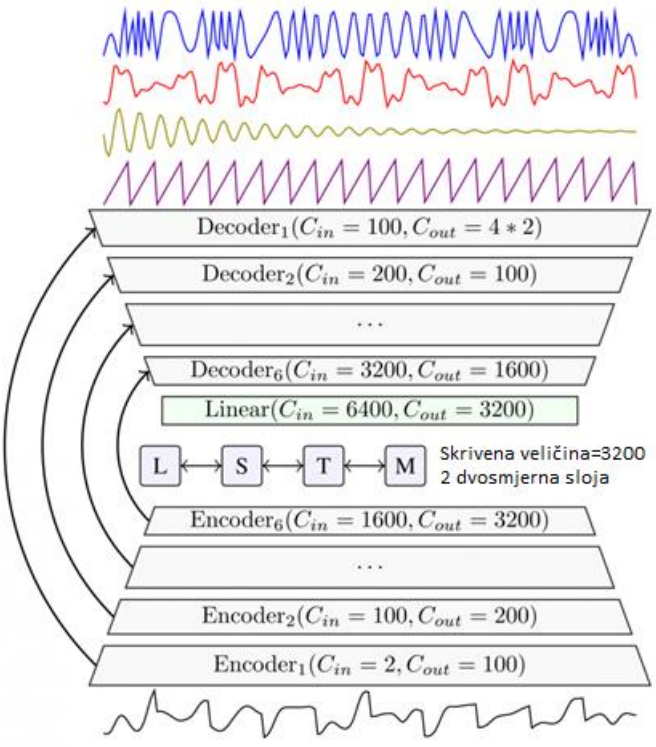


Slika 7.4. Prikaz rezultata nakon ubacivanja audio datoteke na PhonicMind stranici.

8. IMPLEMENTACIJA POSTOJEĆEG RJEŠENJA

Za implementaciju razdvajanja vokala i instrumenata korištena je python biblioteka demucs.

Demucs (Deep Extractor of Music Sources) je python biblioteka koja omogućuje razdvajanje vokala i instrumenata pomoću dubokog učenja i softvera koji je razvio Facebook AI Research (FAIR). Dizajniran je za razdvajanje pojedinačnih audio izvora u miješanim audio datotekama. Demucs je posebno poznat po svojoj sposobnosti razdvajanja glazbe u više kategorija, uključujući vokal, vokalnu pratnju, bas, bubnjeve itd. s rezultatima visoke kvalitete.[15] Demucs koristi mrežnu arhitekturu sličnu U-Netu za učenje izdvajanja uzoraka valnog oblika. Predložena mreža je konvolucijska, što znači da uzima Mel-spektrograme (umjesto amplitude korištena je decibel skala) kao ulazne slike. Demucs je enkoder-dekoder arhitektura koja koristi dvosmjerne LSTM ove (*eng.* Long short-term memory) kako bi uzela u obzir vremensku dimenziju ulaza. Izlaz mreže je stereo procjena za svaki izvor zvuka.[16]



Slika 8.1. Prikaz arhitekture Demucs-a.

Prema istraživačima, Demucs značajno nadmašuje trenutne najsuvremenije metode na skupu podataka MusDB (*eng.* Music Source Separation Database) – što je standardna referentna vrijednost na području razdvajanje izvora zvuka. Također, ljudske procjene Demucsa pokazuju veću percipiranu kvalitetu rezultata u usporedbi s postojećim metodama.[16]

Set podataka korišten za treniranje Demucs mreže je MusDB HQ (*eng.* MusDB High Quality) i dodatni set podataka od 800 pjesama. MusDB HQ je set podataka koji je temeljen na MusDB18, ali sa povećanom kvalitetom audio datoteka te sadržava 150 pjesama koje su spremljene u raw .wav formatu za razliku od MusDB18 u kojemu su datoteke spremljene u .mp4 formatu.[17]

8.1. Kôd

Za instalaciju *Demucs* biblioteke korištena je naredba:

python.exe -m pip install -U demucs[18]

Za razdvajanje vokala od instrumenata korištenjem *Demucs* biblioteke koristi se *demucs.separate.main()* funkcija kojoj se predaju parametri navedeni nakon Slika 8.2.

Linija ***Kôd***

```
1:      import demucs.separate
        demucs.separate.main(["--mp3", "--two-stems", "vocals", "-n",
2:      "mdx_extra", "audio/1001Noc.mp3"])
```

Slika 8.2. Kôd primjer za razdvajanje vokala od ostatka audio izvora pomoću demucs biblioteke.

demucs.separate.main – naredba demucs biblioteke za razdvajanje vokala i instrumenata

--mp3 – označava ekstenziju audio datoteka koje su kreirane razdvajanjem

--two-stems – oznaka koja govori algoritmu da razdvoji audio datoteku na dvije; vokal i instrumente

vocals – oznaka kojom određujemo točno šta želimo izdvojiti od ostatka audio datoteke, možemo razdvojiti vokal, bas, bubnjeve i drugo (izdvajanje svega što ne spada u prethodno navedene kategorije)

-n – zastavica koja se navodi prije imena unaprijed naučenog modela

mdx_extra – ime unaprijed treniranog modela koji želimo koristiti

Popis unaprijed obučenih modela:[18]

- htdemucs: prva verzija Hybrid Transformer Demucs. Trenirano na MusDB + 800 pjesama. Zadani model.
- htdemucs_ft: fino podešena verzija htdemucs, odvajanje će trajati 4 puta više vremena, ali bi moglo biti malo bolje. Isti set pjesama za obuku kao htdemucs.
- htdemucs_6s: verzija htdemucsa sa 6 izvora, s klavirom i gitarom koji su dodani kao izvori. Imajte na umu da klavir trenutno ne radi u punoj kvaliteti.
- hdemucs_mmi: hibridni Demucs v3, preoblikovan na MusDB + 800 pjesama.
- mdx: trenirao samo u MusDB HQ, pobjednički model na ljestvici A MDX (*eng.* Music Demixing) izazova.
- mdx_extra: treniran s dodatnim podacima za treniranje (uključujući MusDB skup testova), rangiran kao 2. na ljestvici B MDX izazova.
- mdx_q, mdx_extra_q: kvantizirana verzija prethodnih modela. Manje preuzimanje, ali kvaliteta može biti nešto lošija.

Nakon pokretanja naredbe stvaraju se dvije .mp3 datoteke, jedna koja sadržava samo vokale (vocals.mp3) i jedna koja sadrži instrumente (no_vocals.mp3). Datoteke su spremljene u mape .../separated/ime_korištenog_modela/ime_pjesme/, mape su automatski stvorene u mapi koja sadržava Python skriptu za razdvajanje.

9. IMPLEMANTACIJA ALGORITMA ZA RAZDVAJANJE

Pristup razvijanju algoritma za razdvajanje je korištenjem Fourierove transformacije za prelazak u frekvencijsku domenu, filtriranje prema izvorima i primjena maske za dodatno pojačavanje pojedinog izvora te povratak u vremensku domenu inverznom Fourierovom transformacijom.

Za razvoj algoritma razdvajanja vokala od instrumenata korišten je programski jezik Python. Python biblioteke koje su bile potrebne su *numpy* za operacije računanja jednostavnih stvari poput minimuma i maksimuma, *matplotlib* za prikaz spektrograma, *librosa* za obradu podataka pomoću filtera, STFT (3.1.3) itd. i *soundfile* za pohranu rezultata u obliku .wav datoteke.

Linija ***Kôd***

```
1:      import numpy as np
2:      import matplotlib.pyplot as plt
3:      import librosa
4:      import librosa.display
5:      import soundfile as sf
```

Slika 9.1. Kôd primjer pytboh biblioteka za implementaciju razdvajanja vokala.

Linija ***Kôd***

```
1:      y, sr = librosa.load('audio/pjesma.mp3', duration = 120)
```

Slika 9.2. Kôd primjer za učitavanje audio datoteke.

Za učitavanje željene audio datoteke korištena je funkcija *librosa.load()* koja nam kao rezultat daje *y* koji predstavlja audio signal kroz jednodimenzionalan *NumPy* niz, a varijabla *sr* je brzina uzorkovanja u herc-ima. Prvi parametar funkcije je lokacija audio datoteke, a drugi predstavlja željenu duljinu uzorka u sekundama.[19]

Linija ***Kôd***

```
1:      magnitude_matrix, phase_matrix = librosa.magphase(librosa.sftf(y))
```

Slika 9.3. Kôd primjer za računanje matrice magnituda i fazne matrice.

Za prelazak u frekvencijsku domenu koristimo funkciju *librosa.sftf()* koja računa STFT signala *y* te daje izlaz u obliku matrice sa koeficijentima kratke Fourierove transformacije.[20] Kako bi rezultat Fourierove transformacije bio pogodan za daljnju obradu pomoću funkcije *librosa.magphase()* rastavljamo matricu Fourierovih koeficijenata na dvije, matricu magnitude i faznu matricu.[21]

Linija ***Kôd***

```
1:      idx = slice(*librosa.time_to_frames([30, 35], sr=sr))
2:      plt.figure(figsize=(12, 4))
        librosa.display.specshow(librosa.amplitude_to_db(magnitude_matrix
3:      [:, idx], ref=np.max), y_axis='log', x_axis='time', sr=sr)
4:      plt.colorbar()
5:      plt.tight_layout()
```

Slika 9.4. Kôd primjer za prikaz spektrograma isječka audio datoteke.

Za prikaz spektrograma određenog dijela pjesme koristimo *librosa.time_to_frames()* funkciju koja omogućava „izvlačenje“ dijela pjesme, preciznije govoreći stvara okvir između 30-e i 35-e[22] sekunde u pjesmi kako bi mogli vidjeti spektrogram tog dijela. No prije nego možemo prikazati spektrogram moramo pretvoriti matricu magnitude u decibele, to radimo sa funkcijom *librosa.amplitude_to_db()* te joj predajemo matricu magnitude i *np.max* što označava da je vrijednost maksimalne amplitude referentna vrijednost za decibel skalu.[23]

Linija ***Kôd***

```
        NN_filter = librosa.decompose.nn_filter(magnitude_matrix,
        aggregate=np.median, metric='cosine',
1:      width=int(librosa.time_to_frames(2, sr=sr)))
2:      NN_filter = np.minimum(magnitude_matrix, NN_filter)
```

Slika 9.5. Kôd primjer izračuna i primjene ne-negattivnog filtera.

Funkcijom *librosa.decompose.nn_filter()* računamo nearest-neighbor filter za razdvajanje vokala. Filtriranje prema nearest-neighbor radi na način da se svaka podatkovna točka zamjenjuje agregacijom(u ovom slučaju medijan) svojih najbližih susjeda u prostoru značajki. Prvi parametar koji predajemo funkciji *nn_filter()* je matrica magnitude, drugi parametar *aggregate=np.median* govori da je medijan funkcija tokom agregacije, odnosno da se odvajanje temelji na medijan vrijednostima izvora (vokal i instrumenti). *metric* parametar omogućava mjerenje sličnosti vremensko-frekvencijskih intervala u određenoj mjernoj jedinici, u ovome slučaju je to kosinus. *width* parametar postavlja širinu filtera u okviru, računa se na temelju vremenskog trajanja 2 sekunde i danoj brzini uzorkovanja.[24]

Linija ***Kôd***

```
1:      margin_i = 2
2:      margin_v = 10
3:      power = 2
        mask_i = librosa.util.softmask(NN_filter, margin_i *
4:      (magnitude_matrix - NN_filter), power=power)
        mask_v = librosa.util.softmask(magnitude_matrix - NN_filter,
5:      margin_v * NN_filter, power=power)
6:      vocals = mask_v * magnitude_matrix
7:      instruments = mask_i * magnitude_matrix
```

Slika 9.6.. Kôd primjer za izračun i primjenu maski za naglašavanje vokala.

Nakon računanja nearest-neighbor filtera funkcijom *np.minimum()* primjenjujemo taj isti filter na matricu magnitude. Primjenom tog filtera dobili smo razdvajanje na vokal i instrumente. *NN_filter* sada predstavlja procijenjenu magnitudu vokala.

Varijable *margin_i* i *margin_v* su parametri koji određuju granice *librosa.util.softmask()* funkcije, a varijabla *power* predstavlja eksponent u računanju te kontrolira oblik maske. Korištenje *librosa.util.softmask()* funkcije omogućava naglašavanje vokala (*mask_v*), odnosno instrumenata (*mask_i*). Kako bi *mask_i* i *mask_v* bile primijenjene potrebno ih je pomnožiti sa matricom magnitude.[25]

Linija ***Kôd***

```
1:      plt.figure(figsize=(12, 8))
2:      plt.subplot(3, 1, 1)
        librosa.display.specshow(librosa.amplitude_to_db(magnitude_matrix[:,
3:      idx], ref=np.max), y_axis='log', sr=sr)
4:      plt.title('Full spectrum')
5:      plt.colorbar()

6:      plt.subplot(3, 1, 2)
        librosa.display.specshow(librosa.amplitude_to_db(instruments[:,
7:      idx], ref=np.max), y_axis='log', sr=sr)
8:      plt.title('Instruments')
9:      plt.colorbar()

10:     plt.subplot(3, 1, 3)
        librosa.display.specshow(librosa.amplitude_to_db(vocals[:, idx],
11:     ref=np.max), y_axis='log', x_axis='time', sr=sr)
12:     plt.title('Vocal')
13:     plt.colorbar()
14:     plt.tight_layout()
15:     plt.show()
```

Slika 9.7.. Kôd primjer za prikaz spektrograma.

Slika 9.7. prikazuje kôd za prikaz spektrograma miješanog signala (pjesme), spektrogram vokala i spektrogram instrumenata.

Linija ***Kôd***

```
1:      vocals_audio = librosa.istft(vocals * phase_matrix)
2:      sf.write('vocals.wav', vocals_audio, sr)
```

Slika 9.8. Kôd primjer za spremanje u .wav datoteku.

Kako bi razdvajanje imala smisla potrebno je spremiti rezultat primjene filtera i maski. To radimo pomoću funkcije *write()* iz *soundfile* biblioteke. No prije nego predamo niz (*vocals_audio*) potrebno ga je vratiti u vremensku domenu inverznom kratkom Fourierovom transformacijom pomoću funkcije *librosa.istft()* kojom predajemo umnožak fazne matrice i varijable *vocals*.

10. REZULTATI RAZDVAJANJA

Svi rezultati razdvajanja dostupni su na linku:
https://drive.google.com/drive/folders/1XQVzbfnrfS22_5wcizvb_ybY_HpuCV9h?usp=drive_link

Ili skeniranjem QR koda:

10.1. Demucs

Rezultati razdvajanja pomoću Demucs-a su jako visoke kvalitete zbog korištenja neuronske mreže u kombinaciji sa U-Net arhitekturom. Ako se poslušaju dobiveni rezultati može se samo sluhom primijetiti visoka kvaliteta obrade, iako rijetko postoje iznimke kod kojih u datoteci sa samo vokalom dolazi do „curenja“ instrumenata kada se tek kreće čuti glas ili između kratkih pauza tokom pjevanja, također do curenja može doći kada se miješa više glasova.

Kvaliteta razdvajanja se najbolje može poslušati u pjesmi „SLAVONSKE LOLE – Duša bećarska“ zbog jednostavnosti pjesme, odnosno nema kompliciranih digitalnih zvukova poput kick-a i basa nego su prisutni samo akustični instrumenti.

10.2. Algoritam za razdvajanje

Rezultati razdvajanja pomoću algoritma nisu najveće kvalitete jer primjenjuju neke od najjednostavnijih alata za razdvajanje kao što su filteri i maske. Kada se poslušaju sve pjesme curenje instrumenata je nešto što je konstantno u većoj ili manjoj mjeri ali usprkos tome i dalje se vokal čuje izraženije. Također je u svim pjesmama prisutno izobličenje glasova i instrumenata.

Pjesma „SLAVONSKE LOLE – Duša bećarska“ je i ovom slučaju bolje kvalitete u odnosu na druge pjesme upravo zbog toga što su prisutni samo akustični instrumenti koji su lakši za izdvojiti.

U pjesmi „SARS – Lutka“ se primjećuje veće curenje instrumenata, ali to su uglavnom dijelovi u kojima do izražaja dolazi truba ili bubnjevi. Isto se može primijetiti u pjesmi „Dino Merlin - Kremen“ kod koje najviše do izražaja, osim vokala, dolaze trube i bas gitara koja zbog načina sviranja proizvodi zvuk koji pravi veliku stršeću vrijednost u miješanom signalu.

Kod pjesama „Bijelo Dugme – Hajde u planine“ i „David Guetta -Shot Me Down“ dolazi do najvećeg curenja instrumenata koje je prisutno i u dijelovima koji uopće ne sadržavaju vokal. Također je kod pjesme „Bijelo Dugme – Hajde u planine“ prisutno veće izobličenje vokala.

11. ZAKLJUČAK

Razdvajanje vokala i instrumenata je jedna od grana koja spaja svijet računala i glazbe i omogućava razvoj sistema koji se primjenjuju u svakodnevici.

Cilj ovog rada bio je riješiti problem razdvajanja vokala i instrumenata u audio datoteci koja sadržava miješani signal sa svim izvorima, odnosno pronaći najbolje rješenje koje omogućava razdvajanje najveće kvalitete.

Odabira pristupa rješavanja problema sveden je na dvije mogućnosti, razdvajanje pomoću gotovog rješenja, Demucs, koje koristi duboke neuronske mreže kao temelj za razvijanje modela za razdvajanje te na razvoj algoritma koji koristi jednostavne alate poput filtera i maski.

Pristup gotovom rješenju je prilično jednostavan i efikasan jer se koristi gotovi model koji je istreniran na velikom setu podatka koji su visoke kvalitete te ispunjava očekivanje isto tako visoko kvalitetnog izlaza, odnosno razdvajanje na vokal i instrumente je veoma lako, a dobivaju se rezultati koji su visoke kvalitete, bez puno grešaka tokom procesa razdvajanja. Međutim kako bi se postigli takvi rezultati potrebni su veliki timovi ljudi i velika računalna snaga koju Facebook-ov AI Research (FAIR) tim svakako posjeduje.

Sa druge strane pristup u kojemu koristimo jednostavne metode za razdvajanje koje uključuju korištenje Short-time Fourier Transform (STFT), ne-negativnog filtera i maski daje rezultate manje kvalitete. Rezultati dobiveni korištenjem tih alata nisu na razini rezultata dobivenih korištenjem Demucs-a, međutim korištenjem te metode za razdvajanje može se dobiti osjećaj koliko je razdvajanje vokala i instrumenata kompleksan zadatak koji zahtjeva veliku količinu podataka i računalnih resursa kako bi se dobili rezultati visoke kvalitete.

LITERATURA

[1] R. J. Issa i Y. F. Al-Irhaym, „Audio source separation using supervised deep neural network“, *J. Phys. Conf. Ser.*, sv. 1879, izd. 2, str. 022077, svi. 2021, doi: 10.1088/1742-6596/1879/2/022077.

[2] D. Wang, „Time-Frequency Masking for Speech Separation and Its Potential for Hearing Aid Design“, *Trends Amplif.*, sv. 12, izd. 4, str. 332–353, pros. 2008, doi: 10.1177/1084713808326455.

[3] „What is frequency domain analysis?“ https://www.collimator.ai/reference-guides/what-is-frequency-domain-analysis (pristupljeno 29. lipanj 2023.).

[4] „Short-time Fourier transform“, *Wikipedia*. 11. listopad 2022. Pristupljeno: 15. rujan 2023. [Na internetu]. Dostupno na: https://en.wikipedia.org/w/index.php?title=Short-time_Fourier_transform&oldid=1115436538

[5] „Non-Negative Matrix Factorization“, *GeeksforGeeks*, 18. srpanj 2021. https://www.geeksforgeeks.org/non-negative-matrix-factorization/ (pristupljeno 11. rujan 2023.).

[6] S. Talebi, „Independent Component Analysis (ICA)“, *Medium*, 01. travanj 2023. https://towardsdatascience.com/independent-component-analysis-ica-a3eba0ccec35 (pristupljeno 11. rujan 2023.).

[7] „ML | Independent Component Analysis“, *GeeksforGeeks*, 21. svibanj 2019. https://www.geeksforgeeks.org/ml-independent-component-analysis/ (pristupljeno 11. rujan 2023.).

[8] „Rindik rod sound separation with spectral subtraction method“. [Na internetu]. Dostupno na: https://iopscience.iop.org/article/10.1088/1742-6596/1810/1/012018/pdf

[9] „Deep Neural Networks“. https://www.tutorialspoint.com/python_deep_learning/python_deep_learning_deep_neural_networks.htm (pristupljeno 30. lipanj 2023.).

[10] „Long short-term memory“, *Wikipedia*. 31. kolovoz 2023. Pristupljeno: 08. rujan 2023. [Na internetu]. Dostupno na: https://en.wikipedia.org/w/index.php?title=Long_short-term_memory&oldid=1173133233

[11] R. Aggarwal, „Bi-LSTM“, *Medium*, 04. srpanj 2019. https://medium.com/@raghavaggarwal0089/bi-lstm-bc3d68da8bd0 (pristupljeno 08. rujan 2023.).

[12] E. Zvornicanin, „Differences Between Bidirectional and Unidirectional LSTM | Baeldung on Computer Science“, 25. siječanj 2022. https://www.baeldung.com/cs/bidirectional-vs-unidirectional-lstm (pristupljeno 08. rujan 2023.).

[13] „VirtualDJ - Real-Time Stems Separation“, *VirtualDJ Website*. https://www.virtualdj.com/stems/ (pristupljeno 08. rujan 2023.).

[14] „VirtualDJ - The #1 Most Popular DJ Software“, *VirtualDJ Website*. https://www.virtualdj.com/help/stems.html (pristupljeno 08. rujan 2023.).

[15] „One-track minds: Using AI for music source separation“, *Tech at Meta*, 06. ožujak 2020. https://tech.facebook.com/artificial-intelligence/2020/3/one-track-minds-using-ai-for-music-source-separation/ (pristupljeno 08. rujan 2023.).

[16] D. Mitrev, „Demucs: New Sound Source Separation Method by Facebook AI“, 08. ožujak 2020. https://neurohive.io/en/news/demucs-new-sound-source-separation-method-by-facebook-ai/ (pristupljeno 08. rujan 2023.).

[17] „Papers with Code - MUSDB18-HQ Dataset“. https://paperswithcode.com/dataset/musdb18-hq (pristupljeno 15. rujan 2023.).

[18] „Demucs Music Source Separation“. Meta Research, 08. rujan 2023. Pristupljeno: 08. rujan 2023. [Na internetu]. Dostupno na: https://github.com/facebookresearch/demucs

[19] „librosa.load — librosa 0.10.1 documentation“. https://librosa.org/doc/main/generated/librosa.load.html (pristupljeno 11. rujan 2023.).
[20] „librosa.stft — librosa 0.10.1 documentation“. https://librosa.org/doc/main/generated/librosa.stft.html (pristupljeno 11. rujan 2023.).
[21] „librosa.magphase — librosa 0.10.1 documentation“. https://librosa.org/doc/main/generated/librosa.magphase.html (pristupljeno 11. rujan 2023.).
[22] „librosa.time_to_frames — librosa 0.10.1 documentation“. https://librosa.org/doc/main/generated/librosa.time_to_frames.html (pristupljeno 11. rujan 2023.).
[23] „librosa.amplitude_to_db — librosa 0.10.1 documentation“. https://librosa.org/doc/main/generated/librosa.amplitude_to_db.html (pristupljeno 11. rujan 2023.).
[24] „librosa.decompose.nn_filter — librosa 0.10.1 documentation“. https://librosa.org/doc/main/generated/librosa.decompose.nn_filter.html (pristupljeno 11. rujan 2023.).
[25] „librosa.util.softmask — librosa 0.10.1 documentation“. https://librosa.org/doc/main/generated/librosa.util.softmask.html (pristupljeno 11. rujan 2023.).

SAŽETAK

Razdvajanje vokala i instrumenata je proces u kojemu uzimamo audio datoteku sa pomiješanim instrumentima i vokalom te pomoću raznih metoda i alata pokušavamo „izvući“ vokal od ostatka pjesme. Jednostavne metode koje i nisu tako učinkovite podrazumijevaju upotrebu Fourierove transformacije u kombinaciji sa filterima i maskama da bi dobili rezultat sa puno preklapanja instrumenata u kanalu koji bi trebao imati samo vokal. Sa druge strane imamo modele koji se temelje na neuronskim mrežama i učenju istih na velikom skupu podataka koji ima dostupno studijski snimljenu pjesmu i samo vokal pa je moguće jako dobro istrenirati model i dobiti rezultate visoke kvalitete. Modeli poput Demucs-a se jako dobro snalaze i u pjesmama sa vokalom koji nije toliko glasan ili je prikriven sa puno pozadinske buke, odnosno u pjesmama koje imaju izražen bas, bubnjeve ili ostale instrumente.

Ključne riječi: razdvajanje vokala i instrumenata, strojno učenje, duboke neuronske mreže, Demucs, Fourierova transformacija

ABSTRACT

Separating vocals and instruments is a process in which we take an audio file with mixed instruments and vocals and try to "extract" the vocals from the rest of the song using various methods and tools. Simple methods that aren't that effective involve using a Fourier transform in combination with filters and masks to get a result with a lot of overlapping instruments in a channel that should only have vocals. On the other hand, we have models that are based on neural networks and learning them on a large data set that has a studio-recorded song and only vocals available, so it is possible to train the model very well and get high-quality results. Models like Demucs do very well in songs with vocals that are not that loud or are hidden with a lot of background noise, that is, in songs that have pronounced bass, drums or other instruments.

Key words: audio source separation, machine learning, deep neural networks, Demucs, Fourier Transformation