**FIŞA LUCRĂRII DE DISERTAŢȚIE**

|  |  |
| --- | --- |
| Universitatea *Transilvania* din Braşov | Lucrare de disertaţie nr. .......... |
| Facultatea :  Matematică și Informatică |
| Departamentul :  Matematică și Informatică | Viza facultății |
| Programul de studii :  Tehnologii moderne în ingineria sistemelor  software | Anul universitar 2017-2019 |
| Candidat :  Predoiu Diana – Elena | Promoția 2019 |
| Cadrul didactic îndrumător | Gâlmeanu Honorius |

|  |
| --- |
| **DISERTAŢȚIE** |
| Titlul lucrării: Separarea surselor de semnal folosind măști de frecvență |
| Problemele principale tratate:  Introducere  Descrierea problemei  Tehnologii folosite: TODO  TODO  Descrierea soluției: Obiective, Implementare (Structura aplicației, Setul de date folosite,  TODO capitolele din descrierea solutiei)  Ghid de utilizare ??? |
| Locul şi durata practicii: |
| Bibliografie: |
| Aspecte particulare: – |

|  |
| --- |
| Primit tema la data de:  1 octombrie 2018 |

Data predării lucrării:

Director departament, Cadru didactic îndrumător,

*(nume, prenume, semnătura)............................................* Lect. Dr. Gâlmeanu Honorius

Candidat,

Predoiu Diana – Elena

**DISERTAȚŢIE – VIZE**

Data

vizei

Capitole/ problemele analizate

Semnătura cadrului didactic îndrumător

|  |  |  |
| --- | --- | --- |
| **APRECIEREA ŞI AVIZUL CADRULUI DIDACTIC ÎNDRUMĂTOR** | | |
|  | | |
| Data: | ADMIS pentru susținere/ RESPINS | CADRU DIDACTIC ÎNDRUMĂTOR  Lect. Dr. Gâlmeanu Honorius |

|  |  |  |
| --- | --- | --- |
| **AVIZUL DIRECTORULUI DE DEPARTAMENT** | | |
| Data: | ADMIS pentru susținere/ RESPINS | Director departament  (nume, prenume, semnătură) |

|  |  |
| --- | --- |
| **SUSȚINEREA DISERTAȚŢIEI** | |
| Sesiunea | |
| Rezultatul  susținerii | PROMOVAT cu media: |
| RESPINS **cu** refacerea lucrării |
| RESPINS **fără** refacerea lucrării |
| PREŞEDINTE COMISIE  (nume, prenume, semnătura) | |

Separarea surselor de semnal folosind măști de frecvență

**Contents**

[1. Introducere 6](#__RefHeading___Toc677_177693844)

[2. Motivația temei 6](#__RefHeading___Toc679_177693844)

[3. Tehnologii folosite 6](#__RefHeading___Toc681_177693844)

[3.1. Python 6](#__RefHeading___Toc683_177693844)

[3.1.1. Descriere 6](#__RefHeading___Toc685_177693844)

[3.1.2. Jupyter Notebook 6](#__RefHeading___Toc687_177693844)

[3.1.3. Pachetul “numPy” 6](#__RefHeading___Toc689_177693844)

[3.1.4. Pachetul “PyTorch” 6](#__RefHeading___Toc691_177693844)

[3.1.5. Pachetul “LibROSA” 6](#__RefHeading___Toc693_177693844)

[3.2. Transformata Fourier 7](#__RefHeading___Toc695_177693844)

[3.2.1. Semnalele și caracteristicile sunetului 7](#__RefHeading___Toc697_177693844)

[3.2.2. STFT (short-time frequency tranform) 7](#__RefHeading___Toc699_177693844)

[3.2.3. Spectrograma 8](#__RefHeading___Toc701_177693844)

[3.2.4. ISTFT (inverse short-time frequency transform) 8](#__RefHeading___Toc703_177693844)

[3.3. Articol îndrumător “Low-latency sound source separation using deep neural networks” 9](#__RefHeading___Toc705_177693844)

[3.3.1. NMF 9](#__RefHeading___Toc707_177693844)

[3.4. Deep neural networks ( Rețele neuronale profunde) 10](#__RefHeading___Toc709_177693844)

[4. Descrierea soluției 10](#__RefHeading___Toc711_177693844)

[4.1. Obiective 10](#__RefHeading___Toc713_177693844)

[4.2. Implementare 10](#__RefHeading___Toc715_177693844)

[4.2.1. Setul de date de antrenare 10](#__RefHeading___Toc717_177693844)

[4.2.2. Setul de date de validare 10](#__RefHeading___Toc719_177693844)

[4.2.3. Setul de date de testare 10](#__RefHeading___Toc721_177693844)

[4.2.4. Încărcarea semnalelor și mixarea lor 10](#__RefHeading___Toc723_177693844)

[4.2.5. Construirea unui set de date de antrenare/validare/testare 10](#__RefHeading___Toc725_177693844)

[4.2.6. Strutura rețelei neuronale 10](#__RefHeading___Toc727_177693844)

[4.2.7. Antrenarea rețelei 10](#__RefHeading___Toc729_177693844)

[4.2.8. Testarea rețelei 10](#__RefHeading___Toc731_177693844)

[4.2.9. Performanțe 10](#__RefHeading___Toc733_177693844)

[4.3. Rezultate 10](#__RefHeading___Toc735_177693844)

[5. Index figuri 10](#__RefHeading___Toc737_177693844)

[6. Concluzie 10](#__RefHeading___Toc739_177693844)

[7. Bibliografie și webografie 10](#__RefHeading___Toc741_177693844)

# Introducere

# Motivația temei

# Tehnologii folosite

## Python

### Descriere

### Jupyter Notebook

### Pachetul “numPy”

### Pachetul “PyTorch”

### Pachetul “LibROSA”

## Transformata Fourier

Din punct de vedere matematic, transformata Fourier reprezintă operația aplicată asupra unei funcții cu elemente complexe, și returnează tot o funcție cu elemente complexe. Ce este diferit la funcția returnată este că este reorganizată după frecvențele componente.

Analiza Fourier este considerată abecedarul analizei semnalelor. Aceasta tranformă semnalul nostru din domeniul timpului în domeniul frecvențelor. În timp ce domeniul timpului exprimă semnalul într-o secvență de probe/bucăți de semnal, domeniul frecvențelor reprezintă semnalul ca sinusoidale, cu diferite magnitudini, frecvențe, și decalaje de fază a semnalului.

### Semnalele și caracteristicile sunetului

Semnalul în timp este reprezentat printr-o anumită funcție, care de regulă dă forma de dependență a nivelului sau amplitudinii semnalului în raport cu timpul.

Semnalele pot fi de două feluri, periodice și neperiodice. Cele periodice sunt caracterizate printr-o perioada care se tot repetă în decursul semnalului, iar cele nepriodice nu conțin această perioadă de repetiție.

La rândul lor, semnalele perioduce sunt de două feluri, armonice și nearmonice.

Semnalul audio este un semnal neperiodic, de cele mai multe ori, fiind determinat de sunete neperiodice. Din punct de vedere electric, semnalul audio este reprezentat de variația în timp a tensiunii sau curentului electric.

Sunetul este caracterizat de patru elemente:

* amplitudinea – caracteristica undei sonore percepute ca volum
* frecvența/înălțimea sunetului – numărul de perioade ( T -> s) sau oscilații efectuate

( F -> HZ) într-un timp dat

* intensitatea sunetului/sonoră – este măsurată în decibeli
* timbrul – calitatea de bază a fiecărui sunet, determinat de componenta de armonici

În procesare semnalelor se ține foarte mult cont și sunt controlate frecvența și intensitatea

sunetelor. În tema aleasă toate înregistrările sunt modificate la 16KHz, iar analiza pentru aflarea măștilor de frecvență se bazează pe ideea de „frecvențe în timp”.

### STFT (short-time frequency tranform)

Transformata Fourier pe termen scurt (STFT) este o transformare de tip Fourier care ajută la determinarea frecvenței și fazei sinusoidalei a secțiunilor locale ale unui semnal, având în vedere schimbarea lui în timp.

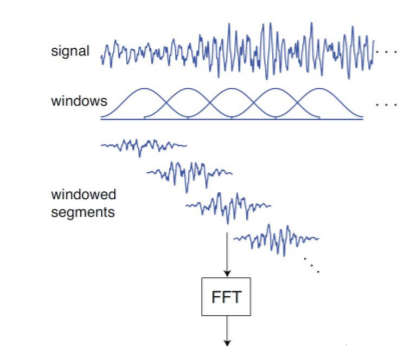
Pentru calculul STFT se împarte un semnal mai lung în segmente mai scurte, de dimensiuni egale, iar mai apoi se calculează tranformata Fourier, separat, pe fiecare segment mai scurt. La final, aceasta duce la aflarea spectrului Fourier pentru fiecare segment mai scurt. După se poate arăta spectrul, printr-un desen, ca o funcție a timpului.

Astfel, tranformata Fourier pe termen scurt este o secvență de transformări Fourier a unui semnal ferestruit. Pentru această transformare trebuie setați mai mulți parametrii, și anume:

* fereastra de analiză ( de exemplu 10/20/30 ms)
* valoarea de „overlap”/„hopping”, cât de mult se dorește suprapunerea ferestrelor în mutarea către următoarea fereastră ( de exemplu overlap de 50%, pentru dimensiunea ferestrelor de 10 ms se vor lua ferestre în felul următor:

0 ms → 10 ms, 5 ms → 15 ms, 10 ms → 20 ms, și așa mai departe)

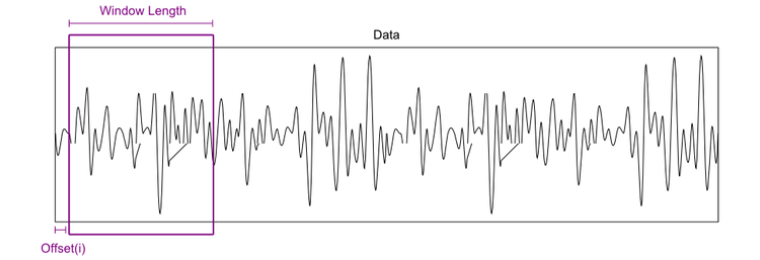
* funcția de ferestruire ( de exemplu Hann, Gaussian, etc. )
* generarea segmentelor ferestruite
* aplicarea tranformatei Fourier pe fiecare segment ferestruit

  
Figure 1: Pașii aplicării transformatei Fourier pe un semnal

Aplicarea algoritmului de STFT constă în respectarea unor pași, și anume:

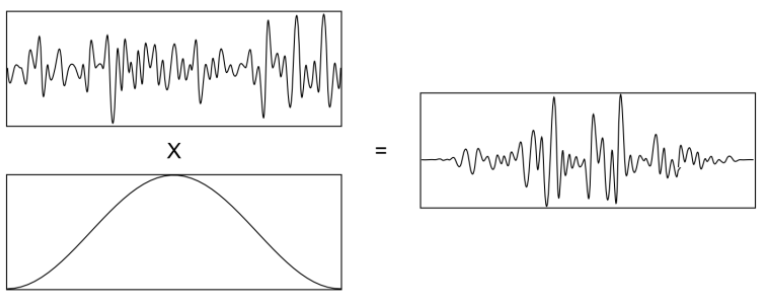
1. Se alege un segment de dimensiune mică din întregul semnal:

* se folosește conceptului de „sliding window” (fereastră care se plimbă) pentru a fi mai ușor de vizualizat ce se întâmplă
* datele dinăuntrul ferestrei reprezintă segmentul curent care va fi procesat
* dimensiunea ferestrei rămâne aceeași pe tot parcursul procesării datelor, dar punctul de pornire („offsetul”) se va schimba după fiecare pas din algoritm
* de obicei, schimbarea acestui „offset” constă într-o valoare mai mică decât dimensiunea ferestrei, pentru a face posibil fenomenul de „overlap”, astfel încât fereastra precedentă și cea curentă se vor suprapune

  
Figure 2: Fereastră care se plimbă peste un semnal

1. Se înmulțește segmentul cu o funcție de jumătate de cosinusoidă:

* folosită pentru a evita problemele create de segmentarea unui semnal
* în momentul în care tăiem un semnal în bucățele mai mici, marginile creează o tranziție ascuțită care nu exista înainte în semnal
* înmulțind cu această funcție „jumătate cosinusoidală”, ajută la atenuarea semnalului, astfel încât tranzițiile la margini să nu afecteze transformata Fourier a datelor

  
Figure 3: Datele unui semnal după aplicarea funcției "half-cosine"

* alt atuu al acestei funcții, este că având un overlap de 50%, suma tuturor ferestrelor este o constată egală cu 1; acest lucru înseamnă că înmulțind datele noastre cu o anumită funcție, nu se vor adăuga diferențe în amplitudinea semnalului original

1. se adaugă zerouri la finalul segmentului:
2. aplicarea transformatei Fourier asupra segmentului și normalizarea în frecvențe pozitive și negative:

* numărul the bucăți de date care intră este egal cu numărul de bucăți de frecvențe care ies
* datele date de transformarea Fourier trebuie să fie scalate cu numărul de bucăți din transformare pentru a menține egala energia semnalului ( în concordanță cu teorema lui Parseval)
* transformata Fourier returnează atât frecvențele pozitive, cât și cele negative

1. combinarea energiilor frecvențelor pozitive și negative la un loc și crearea unui grafic
2. scalarea spectrului rezultat în dB pentru a fi mai ușor de văzut:

* convetirea datelor în dB are efectul de a lungi vârfurile frecvențelor în jos, spre nivelul mediu al sunetului, și de a aduce valorile foarte joase în sus.
* acest lucru permite compararea conținutului la toate nivelele de amplitudine

1. înlăturarea zgomotului care se află peste o anumită limită:

* tot ce este sub -40 dB este sub limita minimă a frecvențelor, astfel încât se consideră ca este frecvențe sunt erori numerice din algoritm

### Spectrograma

„Spectrograma este reprezentarea vizuală a unui spectru (a densității spectrale) de frecvențe a unui sunet în raport cu timpul sau cu alte variabile [[1]](#footnote-2).”

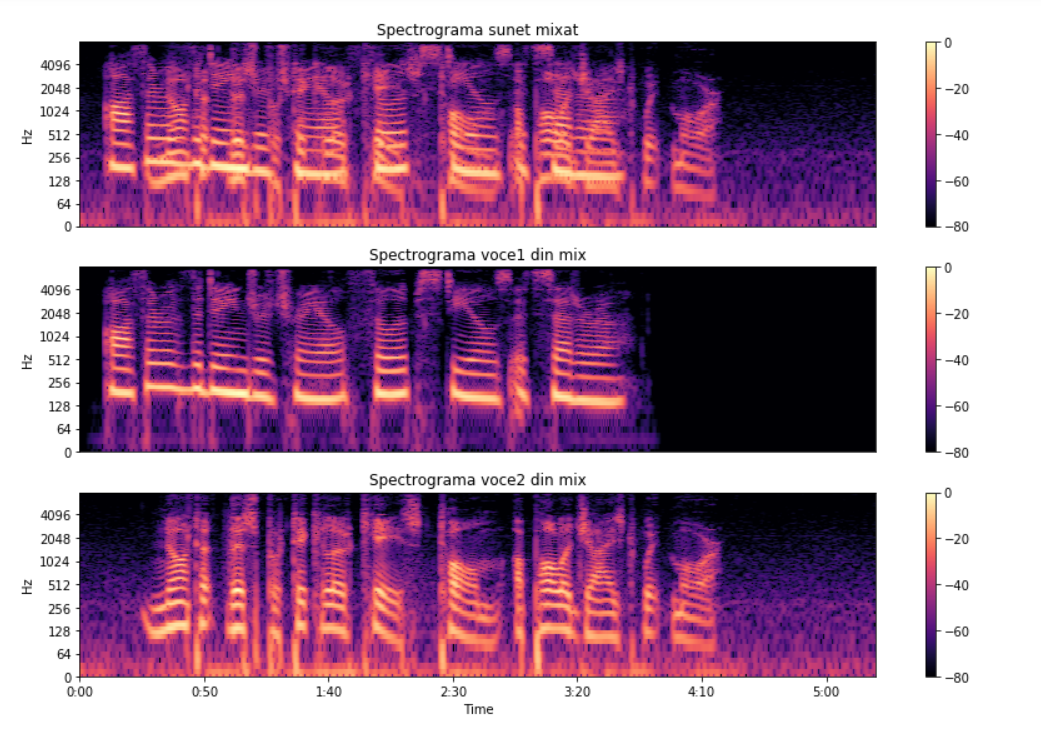
Aceasta este de obicei asociată cu un „heat map”, o imagine cu insitatea dată de varietatea culorilor și a luminozității.

Formatul general este un grafic cu 2 dimensiuni plus una separată pentru amplitudine:

* o axă reprezintă timpului, cea orizontală
* o axă reprezintă frecvențele, cea verticală
* o a treia dimensine ar fi cea a amplitudinilor frecvențelor la un anumit timp, care sunt reprezentate de intensitatea de culoare a acelui punct

Există multe moduri pentru desenarea graficul spectrogramei, uneori axele orizontale și verticale sunt inversate, astfel încât timpul se desfășoară în sus și în jos, alteori graficul este de tip „waterfall” unde amplitudinea este reprezentată de înălțimea unei suprafețe 3D în locul intensității sau culorii. Axele de frecvență și amplitudine pot fi ori lineare, ori logaritmice, în funcție de scopul spectrogramei. Semnalele audio sunt reprezentate de cele mai multe ori cu o axă logaritmică a amplitudinii (decibeli, dB) și frecvențele ar putea fi liniare pentru a menține relația armonică, sau logaritmică pentru a menține relațiile muzicale/tonurile.

În continuare sunt prezentate trei spectrograme, folosite în aplicație. Prima spectrogramă este mixul dintre următoarele două. Spectrograma a doua și a treia sunt două voci diferite, una de femeie și una de bărbat. Cum este specificat mai sus, pe axa orizontală avem timpul, pe cea verticală sunt frecvențele, iar în dreapta avem o scară pentru măsurarea amplitudinii în dB.

  
Figure 4: Spectrograme folosite în aplicație pentru două voci și mixul lor

Pentru calculul spectrogramei se va calcula magnitudinea la pătrat, unde magnitudinea este valoarea absolută: spectrogram(f, t) = | STFT(t, f) |2.

### ISTFT (inverse short-time frequency transform)

STFT este o operație inversabilă, însemnând că după aplicarea transformării STFT asupra semnalului audio, acesta poate fi recuperat folosind operația de transformare inversă Fourier, dar doar atâta timp cât matricea respectivă mai conține elementele complexe (faza semnalului). Dacă se face transformarea matricii către spectrogramă/magnitudine, din acel punct transformarea inversă Fourier nu mai are niciun efect, deoarece elementele complexe ale semnalului s-au pierdut în momentul aflării valorii absolute pentru magnitudine.

Dacă se dorește să se modifice spectrul complex al semnalului, acesta mai poate fi recuperat folosind metoda „overlap – add (OLA)”.

### Măști de frecvențe în timp

Se presupune că avem două surse, un semnal pentru voce și un semnal pentru muzică. Presupunem că mixul celor două surse este risipit, adicpă majoritatea punctelor de tip frecvență – timp sunt nule.

Presupunem că primul semnal, muzica, are majoritatea punctelor de energie joasă, iar al doilea semnal are majoritatea punctelor de energie înaltă.

Se mai presupune și că cele două semnale sunt disjuncte, adică punctele lor de tip frecvență – timp nu se suprapun, sau se suprapun foarte puțin.

Având mixul risipit și disjuncția, cele două surse din mix se pot discrimina destul de ușor. Punctele care au probabilitatea să aparțină de o sursă sunt marcate cu valoarea constată „1”, iar restul cu „0”. Astfel rezultă o mască binară (varianta mai riguroasă a algoritmului).

După calcularea măștii, prin înmulțirea măștii cu sursa mixată, se va obține sursa pentru care s-a calculat masca.

În majoritatea cazurilor, mixurile nu sunt foarte risipite și nici sursele disjuncte. Punctele din mix care sunt probabil să aparțină unei surse au valori aproape de valoarea 1, iar restul aproape de valoarea 0 ( varianta de „soft – masking”). Ecuația cu respect pentru prima sursa se semnal din mix, este următoarea :

Ecuația 1: Calcularea măștii de frecvență cu respect pentru prima sursă de semnal

Având masca pentru o sursă, pentru cealaltă sursă masca se poate calcula fie în același mod, dar înlocuind numărătorul cu sursa pentru al doilea semnal, fie scăzând masca dintr-o matrice construită, care va avea aceeași dimensiune și va conține doar valori de 1.

Ecuația 2: Calcularea măștii de frecvență cu respect pentru ambele surse de semnal

## Articol îndrumător “Low-latency sound source separation using deep neural networks”

Separarea sursei de semnal la o latență scăzută necesită ca fiecare input, o fereastră audio, să fie procesată la o temporizare cât mai mică, iar outputul să fie generat cât mai repede.

Articolul propune o rețea neuronală profundă, deep neural network (DNN), cu latență scăzută. Această metodă are avantajul de a folosi un context extins „din trecut”, mai apoi generându-se filtre/măști de tipul „soft time frequencies masks”, care mai apoi sunt aplicate peste sunetul mixat, astfel încât să fie oferită o performanță de separare mai bună, comparând cu „NMF baseline”.

### NMF

Separarea surselor poate fi văzută ca o problemă de factorizare a matricei, unde sursa mixtură este modelată ca o matrice care conține reprezentarea spectrogramei aferente.

Spectrogramele sunt de obicei folosite pentru vizualizarea timpului cu variația spectrală

a densității unei înregistrări audio. Semnalele audio pot, astfel, să fie reprezentate în întregime de o matrice cu :

* linii, axa orizontală ( reprezentarea timpul)
* coloane, axa verticală ( reprezentarea frecvențelor)
* valori ale elementelor, care corespund intensitatății sau culorii fiecărui punct din imagine ( indicând amplitudinea unei frecvențe particulare la un anumit moment „t”).

Spectrograma unui semnal x(t) poate fi estimată prin calculul magnitudinii la pătrat a

transformatei Fourier (prezentată anterior la secțiunea TODO), a semnalului x(t), astfel încât x(t) poate fi reconstruit din spectrogramă prin inversa transformatei Fourier (prezentată de asemenea în secțiunea TODO), după procesarea semnalului în domeniu spectral.

## Deep neural networks ( Rețele neuronale profunde)

# Descrierea soluției

## Obiective

## Implementare

### Setul de date de antrenare

### Setul de date de validare

### Setul de date de testare

### Încărcarea semnalelor și mixarea lor

### Construirea unui set de date de antrenare/validare/testare

- construirea inputului

- construirea targetului -> mastile

### Strutura rețelei neuronale

### Antrenarea rețelei

### Testarea rețelei

### Performanțe

## Rezultate

# Index figuri

# Concluzie

# Bibliografie și webografie

* „Blind Audio Source Separation Pipeline and Algorithm Evaluation” - Wisam Reid, Kai-Chieh Huang & Doron Roberts-Kedes

(<http://cs229.stanford.edu/proj2015/124_report.pdf>)

* <https://musicinformationretrieval.com/stft.html>
* https://en.wikipedia.org
* [https://librosa.github.io/librosa](https://librosa.github.io/librosa/)
* https://docs.scipy.org/doc/numpy
* “Time frequency masking”, Machine Perception of Music & Audio - Zafar Rafii, 2014 (<http://users.cs.northwestern.edu/~pardo/courses/eecs352/lectures/MPM14-Time-Frequency-Masking.pdf> )

1. [https://ro.wikipedia.org/wiki/Spectrogram%C4%83](https://ro.wikipedia.org/wiki/Spectrogramă) [↑](#footnote-ref-2)