

# IMPLEMENTAREA ALGORITMULUI DE IDENTIFICARE A COMENZILOR VOCALE PENTRU SISTEME DE CONTROL ÎN TIMP REAL

Olesea BOROZAN\*

*Universitatea Tehnică a Moldovei, Facultatea Calculatoare Informatică și Microelectronică,  
Departamentul Informatică și Ingineria Sistemelor,  
Școala Doctorală Știința Calculatoarelor, Electronică și Energetică, Chișinău, Republica Moldova*

\*Autorul corespondent: Olesea Borozan: [olesea.borozan@ia.utm.md](mailto:olesea.borozan@ia.utm.md)

**Rezumat:** În lucrarea de față sunt prezentate rezultatele proiectării unui sistem de control în baza comenzilor vocale și o posibilă implementare a acestuia folosind limbajul de programare Python. Funcționalitatea sistemului este bazată pe achiziția comenzilor vocale generate de utilizator, digitalizarea acestor comenzi și recunoașterea acestora în baza modelelor de Rețele Neuronale.

**Cuvinte cheie:** Python, Rețele Neuronale, recunoașterea vocii, comenzi vocale, sistem de control.

## Introducere

Recunoașterea vorbirii este procesul de recunoaștere automată a vorbirii umane. Sistemele de recunoaștere a vorbirii au o gamă largă de aplicații, de la interacțiunea om-computer până la transcrierea automată a vorbirii sau generarea de comenzi pentru procese de timp real. Asistenții personali, precum Asistentul Google, Amazon Alexa sau Siri-ul Apple, folosesc sisteme de recunoaștere a vorbirii pentru a recunoaște vocile umane și pentru a răspunde în consecință.

Totuși, sistemele de recunoaștere automată a vorbirii prezintă unele aspecte sensibile: acuratețea sistemului, timpul în care acesta procesează vorbirea, resursele consumate, etc. Fiecare aspect poate fi mai mult sau mai puțin relevant, în funcție de scopul final al aplicației. O acuratețe scăzută va conduce la transcrieri greșite ale semnalului vocal, iar acest fapt va genera interpretări greșite sau generarea de comenzi eronate [1].

Procesul recunoașterii automate a vorbirii vizează transformarea unui semnal audio ce conține vorbire într-o succesiune de cuvinte. Recunoașterea automată a vorbirii este unul dintre primele domenii în care tehnicile de modelare statistică (în care modelele se crează pe baza unor cantități mari de date) s-au impus ca și standard [2].

## Recunoașterea vocii cu ajutorul Rețelelor Neuronale și Limbajul Python

Rețelele neurale sunt nucleul Inteligenței Artificiale (AI) și sunt esențiale pentru învățarea mecanică. Acestea sunt rețele interconectate de algoritmi, inspirate de rețelele neurale biologice, existente în creierul uman. Aceste rețele fac schimb de date și pot fi programate pentru a îndeplini anumite sarcini într-un mod complex care încearcă să imite gândirea omenească [3-5].

Ideea de Inteligență Artificială este legată de rețelele neurale, pentru că dezvoltarea acestora este singura metodă actuală prin care se poate dezvolta AI-ul.

Rețelele Neuronale sunt sisteme adaptive de tip cutie neagră (black-box) care permit extragerea unui model pornind de la date printr-un proces de învățare [3].

## Implementarea unui model de recunoaștere voce în Limbajul Python

Există multiple limbaje de programare cu ajutorul cărora putem implementa o Rețea Neuronală sau crea un soft complex pentru recunoașterea vorbirii, unul dintre cele mai populare de pe piață este limbajul Python.

Python este un limbaj de programare de nivel înalt și poate fi utilizat pentru scenarii complexe, are biblioteci pre-construite precum *Numpy* pentru calcul științific, *Scipy* pentru calcul avansat și *Pybrain* pentru învățarea automată (*Python Machine Learning*), făcându-l una dintre cele mai bune limbi pentru AI. La fel și librăriile gratuite care pot fi folosite în acest limbaj, de exemplu: (Tensorflow, Keras, Scikit-learn), și care facilitează implementarea algoritmilor de învățare automată, cum ar fi Rețelele Neuronale.

În calitate de exemplu de implementare vom descrie un simplu model de recunoaștere a vorbitorului (bărbat/femeie), cu ajutorul limbajului Python.

Procesul de recunoaștere automată a vorbirii poate fi descris ca un proces în patru etape și anume [3-5]:

- a. Înregistrarea eșantioanelor vocale;
- b. Prelucrarea semnalelor;
- c. Extragerea caracteristicilor;
- d. Clasificarea și prezicerea vorbitorului.

Primul pas este înregistrarea eșantioanelor vocale cu ajutorul unui dispozitiv, de exemplu microfon. În Python cu ajutorul librăriei *Pyaudio* putem înregistra suntele.

În Figura 1 este prezentat un exemplu de cod pentru înregistrarea unui sunet de la microfon [5].

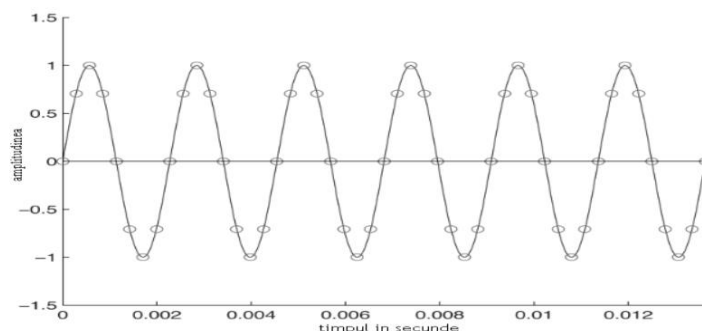
```
import pyaudio
def record():
    p = pyaudio.PyAudio()
    stream = p.open(format=pyaudio.paInt16, channels=1, rate=16000, input=True,
                    output=False, frames_per_buffer=1024)
    stream.read(1024)
    stream.stop_stream()
    stream.close()
```

**Figura 1. Înregistrarea sunetului**

Următorul pas este prelucrarea semnalelor și constă în analiza spectrală a semnalului electronic pentru a transforma undele sonore în numere digitale.

Undele sonore sunt unidimensionale. În fiecare moment de timp, acestea au o singură valoare bazată pe înălțimea undei. Pentru recunoașterea vorbirii, o rată de eșantionare de 16KHz (16.000 pe secundă) este suficientă pentru a acoperi intervalul de frecvență al vorbirii umane.

Pentru a transforma o undă sonoră în format digital, se înregistrează înălțimea undei în punctele egal distanțate. Ca rezultat vom obține o serie de numere în punctele discrete. În Figura 2 este prezentată o undă de eșantionare [3-4].



**Figura 2. Undă de eșantionare**

Librăria Python *Librosa* destinată pentru analiză audio, este foarte eficientă și conține un set de funcții utilitare care ajută la transformarea datelor audio în numere digitale.

Deasemenea *Librosa* include funcții pentru extragerea caracteristicilor din fișierul audio, exemplu Figura 3.

Deci, cu doar câteva linii de cod putem obține rezultatul dorit și informația din format audio în format digital.

```
def extract_feature(file_name, **kwargs):
    ...
    X, sample_rate = librosa.core.load(file_name)
    mel = np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T,axis=0)
    result = np.hstack((result, mel))
    ...
```

**Figura 3. Extragerea caracteristicilor**

Odată ce obținem sunetul într-un format ușor de procesat, digital, vom aplica datele originale ale sunetului, împărțite în fragmente, într-o Rețea Neuronală.

Crearea unei rețele neuronale, în Python, implică mai mulți pași:

- a. Pregătirea setului de date;
- b. Construirea modelului;
- c. Instruirea modelului;
- d. Testarea modelului.

Putem menționa următoarele biblioteci care sunt foarte utile în crearea unei rețele neuronale. Librăria **Pandas** pentru citirea datelor din fișiere, librăria TensorFlow cu ajutorul căreia vom crea rețeaua neuronală.

În Figura 4 este reprezentată definirea unui model de rețea neuronală cu 5 straturi și un neuron la ieșire, cu o funcție de activare sigmoidă în stratul de ieșire. Modelul va emite cifra 1 (sau aproape de 1) atunci când sunetul audio este un bărbat și 0 (sau aproape de 0) când sunetul este o femeie [6].

```
def create_model(vector_length=128):
    model = Sequential()
    #5 straturi dense de la 256 la 64 de unități
    model.add(Dense(256, input_shape=(vector_length,)))
    model.add(Dropout(0.3))
    model.add(Dense(256, activation="relu"))
    model.add(Dropout(0.3))
    model.add(Dense(128, activation="relu"))
    model.add(Dropout(0.3))
    model.add(Dense(128, activation="relu"))
    model.add(Dropout(0.3))
    model.add(Dense(64, activation="relu"))
    model.add(Dropout(0.3))
    # un neuron de ieșire cu funcție de activare sigmoidă, 0 reprezintă femeie, 1 bărbat
    model.add(Dense(1, activation="sigmoid"))
    # folosind crossentropia binară, deoarece este clasificarea masculină / feminină (binară)
    model.compile(loss="binary_crossentropy", metrics=["accuracy"], optimizer="adam")
    # afișarea rezumatului
    model.summary()
```

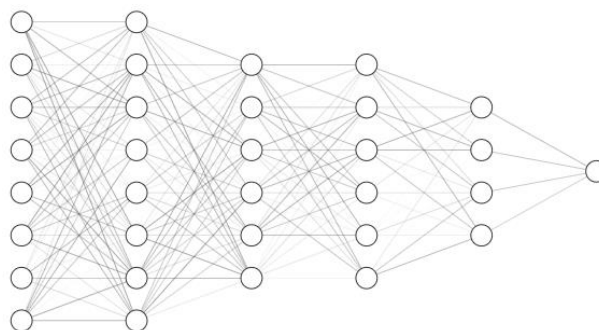
**Figura 4. Model de Rețea Neuronală**

În Figura 5 este reprezentată structura Rețelei Neuronale în formă grafică, unde:

Input Layer - este nivelul de intrare ce atașează informația de la senzori (sunetul de intrare);

Hidden Layer 2 - 5 - nivele ascunse.

Ouput Layer - este nivelul de ieșire, ce emite cifra 1 sau 0;



**Figura 5. Rețea Neuronală**

### **Concluzii**

Recunoașterea automată a vorbirii reprezintă un pas important în dezvoltarea sistemelor bazate pe Inteligența Artificială, în special a sistemelor controlate prin intermediul comenzilor vocale, astfel ușurând accesul persoanelor neinstruite în domeniul informaticii, la resurse controlate de calculator și alte dispozitive inteligente.

În lucrarea dată s-a efectuat o analiză a Rețelelor Neuronale cu mai multe nivele și s-a prezentat implementarea unui model a Rețelei Neuronale cu ajutorul limbajului de programare Python, pentru recunoașterea vorbitorului în timp real.

Pentru viitor sunt planificate dezvoltarea de noi modele de Rețele Neuronale pentru recunoașterea vorbirii în limba română și dezvoltarea unui set de comenzi digitale pentru comanda unui robot mobil.

### **Mențiuni**

Lucrarea de față a fost elaborată în cadrul tezei de doctorat cu tema "Procesor specializat cu proprietăți cognitive" în cadrul Departamentului Informatică și Ingineria Sistemelor.

### **Referințe**

1. CUCU, H., GEORGESCU, A.L., *Sistem de recunoaștere automată a vorbirii în limba română bazat pe rețele neuronale profunde*, București 2018 , [accesat 01.03.2021] <https://speed.pub.ro/speed3/wp-content/uploads/2018/07/Binder1.pdf>.
2. CUCU, H., *Arhitectura generală a unui sistem de recunoaștere automată a vorbirii*, [accesat 03.03.2021] <http://speed.pub.ro/speed3/wp-content/uploads/2014/02/Indrumar-de-proiect-PCDTV-v11.pdf>.
3. SIGMUND, M. *Automatic Speaker Recognition by Speech Signal*, [accesat 03.03.2021] [https://www.researchgate.net/publication/221787287\\_Automatic\\_Speaker\\_Recognition\\_by\\_Speech\\_Signal](https://www.researchgate.net/publication/221787287_Automatic_Speaker_Recognition_by_Speech_Signal).
4. GEITGEY, A. *How to do Speech Recognition with Deep Learning*, [accesat 04.03.2021] <https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a>.
5. ABDOU, R. *How to Perform Voice Gender Recognition using TensorFlow* [accesat 04.03.2021] <https://www.thepythoncode.com/code/gender-recognition-by-voice-using-tensorflow-in-python#test-py>.
6. ABDOU, R. *How to Perform Voice Gender Recognition using TensorFlow* [accesat 04.03.2021] [https://www.thepythoncode.com/article/gender-recognition-by-voice-using-tensorflow-in-python#Prepare\\_data](https://www.thepythoncode.com/article/gender-recognition-by-voice-using-tensorflow-in-python#Prepare_data).