Detectia genului vorbitorului

Nume & Prenume: Alexi Mariuta   
Facultatea : *ETTI*   
*email: alexi.mariuta@yahoo.com*

*Abstract*— **Retelele neuronale sunt adesea utilizate pentru abordarea problemei detectarii genului vorbitorului. Acestea pot fi antrenate cu seturi de date mari care conțin exemple de voci feminine și masculine. Caracteristicile semnificative ale vocii, precum frecvențele, amplitudinea și modelele temporale, pot fi extrase de catre rețelele neuronale pentru a distinge intre genuri. Algoritmi precum rețelele neuronale convolutionale (CNN) pot fi eficienti in captarea acestor caracteristici si in realizarea de predictii corecte pe baza lor. Acest proiect exploreaza detectarea genului unui vorbitor utilizand algoritmi de invatare automata, in special retelele neuronale TensorFlow si CNN2D. In etapa inițiala, au fost modelati si antrenati acesti algoritmi folosind o baza de date extinsa. Apoi, au fost efectuate teste utilizand doua metode diferite. Pentru primul model, s-a inregistrat un set propriu de date si s-au extras parametrii corespunzatori din baza de date inițiala. Acești parametri au fost apoi utilizați pentru a realiza predictii. Pentru al doilea model, s-a selectat un sample din baza de date de testare si s-au efectuat predictii pe baza acestuia. Rezultatele obtinute au confirmat eficienta ambelor metode, oferind rezultatele așteptate si promitatoare in detectarea genului vorbitorului. Această abordare deschide perspective interesante pentru dezvoltarea ulterioara a aplicatiilor de identificare a genului pe baza discursului utilizand algoritmi de invatare automata.**

Key:Retele neuronale TensorFlow, CNN,layer

# Introducere

Evoluțiile in invatarea automata, in special in rețelele neuronale, au permis dezvoltarea de metode eficiente de identificare a genului vorbitorului in ultimii ani. Unul din scopurile lor este sa gaseasca caracteristici distincte ale vocii care să permita identificarea precisa a genului vorbitorului. In acest scop, s-a utilizat o baza mare de date care contine o serie de caracteristici relevante pentru identificarea genului.

In timpul vorbirii, vorbitorul produce un semnal vocal care contine un spectru vast de informatii legate de acesta cum ar fi: varsta,genul, accentual sau starea de sanatate.Sistemele folosite pentru extragerea informatiilor din semnalul vocal pot sa fie folosite in numeroase aplicatii precum identificarea personala in momemntul utilizarii unor sisteme bancare,aplicatii de ingrijire a clientilor, cum ar fi centrele de apeluri. Astfel de servicii sunt deja oferite de companii mari precum Google,Amazon si Technmo[1].

Sistemele de recunoastere a genului vorbitorului au fost implementate in sisteleme de raspuns vocal(IVR),de exemplu pentru a redirectiona vorbitorul catre un consultant corespunzator sau pentru a reda o muzica potrivita pentru o persoana de un anumit gen.In decursul aniilor, s-au dezvoltat mai multe mdetode de determinare a genului vorbitorului[1].

Modelul de invatare profunda bazata pe Multilayer Perceptron(MLP) are o acuratete de 96% pe setul de date primit si acesta foloseste un set de date IViE care implementeaza mai multe clasificatoare ca de exemplu: MLP,GMM,cuantificatrea vectoriala de invatare[2].

Alhussein și colab au dezvoltat o metoda cu acuratete de 98.27% pentru vorbirea curate si 96.55% pentru vorbirea zgomotoasa. Aceasta metoda a fost implementata in domeniul timp si in plus a masurat intensitatea vocii pentru a face diferenta cat mai buna intre barbati si femei.[2].

# Metoda

In acest proiect, tehnicile de invatare automata si retelele neuronale sunt utilizate pentru a realiza detectarea genului vorbitorului pe setul de date.

Pentru realizarea proiectului se foloseste o baza de date care contine 3168 de inregistrari ale vocii, prelevate atat de la femei cat si de la barbati. Baza de date contine o multime de caracetristici cum ar fi:frecvența medie (meanfreq),deviația standard a frecvenței(sd),interval interquantile (IQR),entropie spectrală(sp.ent),frecvență de mod(mod),frecvența de vârf (frecvența cu cea mai mare energie)(peakf),medie a frecvenței fundamentale măsurată pe semnalul acustic,frecvența fundamentală minimă măsurată în semnalul acustic, frecvența fundamentală maximă măsurată în semnalul acustic(meanfun),media frecvenței dominante măsurată pe semnalul acustic(meandom), interval de frecvență dominantă măsurată pe semnalul acustic,indice de modulație( calculat ca diferența absolută acumulată între măsurătorile adiacente ale frecvențelor fundamentale împărțită la intervalul de frecvență),etichetă(bărbat sau femeie)

Frecventa fundamentala medie este un indicator important al sexului vocii, pentru femei avem o frecventa in jur de 210 Hz, pentru barbati o frecventa de 120 Hz, iar la copiii poate sa ajunga pana la o frecventa de 300 Hz.

Pentru pregatirea datelor, se utilizeaza bibliotecile pandas și numpy pentru manipularea datelor, iar StandardScaler din biblioteca scikit-learn este utilizat pentru normalizarea datelor. LabelEncoder este, de asemenea, utilizat pentru a converti variabilele tinta în numere.

Reteaua neuronala este creata cu ajutorul bibliotecii TensorFlow si consta dintr-o serie de straturi dense și o funcție de activare sigmoidă, care produce o singura iesire binara.Pentru antrenarea modelului este utilizat optimizatorul Adam, funcția de pierdere binary\_crossentropy, precum și metricile de precizie și AUC. EarlyStopping este, de asemenea, utilizat pentru a intrerupe antrenarea modelului in cazul in care nu exista o imbunatatire semnificativa a performanței.

Tehnicile de preprocesare a datelor sunt utilizate pentru a imbunatati performanța modelului prin aplicarea unui filtru convoluțional 2D (CNN) la date. Acest lucru este realizat cu ajutorul bibliotecii Keras, unde sunt utilizate doua straturi Conv2D si doua straturi MaxPooling2D pentru a reduce dimensiunea datelor. Acest lucru este urmat de doua straturi Flatten si doua straturi Dense pentru a ajunge la stratul de iesire.

Această abordare se bazează pe teoria rețelelor neuronale si aplica filtre convolutionale pentru a preprocesa datele. Utilizarea optimizatorului Adam si a funcției de pierdere binary\_crossentropy este, de asemenea, validata ca fiind eficienta pentru problemele de clasificare binara.

Prin urmare, combinatia acestor tehnici si metode duce la crearea unui model de invatare automata care poate detecta sexul vorbitorului cu o precizie de peste 99%.

In literatura de specialitate sunt prezentate mai multe tipuri de retele care ar putea fi folosite pentru detectarea genului unui vorbitor.

## Retele neuronale Tensorflow

Rețelele neuronale de invatare profunda sunt subiecte cheie in stiinta si tehnologia informatica si ofera in prezent solutii optime pentru numeroase probleme din domeniile recunoasterii imaginilor, recunoașterii vorbirii și procesarii limbajului natural.

Retelele neuronale sunt alcatuite din neuroni. Din punct de vedere biologic, neuronii sunt conectati prin sinapse prin care circula informatii (calcularea ponderilor modelelor). Neuronii artificiali reprezinta unitatea elementarea dintr-o retea neoronala si prezinta o structura arborescenta cu noduri de intrare si un nod de iesire conectat la toate nodurile de intrare.

A picture containing text, diagram, line, font

Description automatically generated

Figure - Neuron biologic/ neuron artificial

Atunci cand se antreneaza o rețea neuronala, se doreste ca neuronii sa fie declansati atunci cand invata un anumit model din date, motiv pentru care se utilizeaza o funcție de activare pentru a realiza modelarea.

Tipuri de rețele neuronale

1. Rețea neuronală de tip feed-forward

O retea neuronala de tip feed-forward este un tip de retea neuronala artificiala in care conexiunile dintre unitati nu formeaza bucle. In aceasta retea, informatia merge într-o singura directie, de la un nod de intrare printr-un nod ascuns la un nod de iesire. In retea nu exista bucle sau inele[3] [4].

1.1 Perceptron cu un singur strat

Aceasta este cea mai simpla retea neuronala de tip feed-forward si nu contine un strat ascuns, adică contine doar un singur strat de noduri de iesire. Stratul de intrare nu efectueaza niciun calcul, deoarece intrarea este alimentata direct la iesire prin intermediul unei serii de ponderi[3].

A picture containing text, diagram, line, screenshot

Description automatically generated

Figure - Percepton monostrat

1.2 Perceptron multistrat (MLP)

Astfel de rețele sunt formate din mai multe straturi de unitati de calcul care sunt de obicei interconectate in mod feed-forward. Fiecare neuron dintr-un strat este conectat direct la neuronii din stratul urmator. In multe aplicații, aceste unitati ale retelei utilizeaza o functie sigmoida ca funcție de activare, motiv pentru care MLP sunt atat de utile: pot învata reprezentari neliniare[7].

A picture containing diagram, line, origami

Description automatically generated

Figure - Percepton multistrat

## CNN

In zilele noastre, cele mai comune utilizari ale retelelor neuronale convolutionale sunt detectarea si segmentarea obiectelor in imagini, precum și recunoasterea si clasificarea scenelor si obiectelor. Principalele beneficii ale CNN sunt urmatoarele: elimina necesitatea extragerii manuale a caracteristicilor, deoarece acestea sunt invatate automat din datele de antrenare; nivelul ridicat de performanță la recunoastere – comparabil cu sau peste cel al oamenilor; și permite construcții ulterioare pe baza partilor inițiale ale CNN care au fost preantrenate.

O rețea neuronala convolutionala este un tip de retea neuronala cu mai multe straturi. Procesarea datelor cu un aranjament similar cu cel al unei grile pentru an extrage caracteristicile esențiale. Un mare avantaj al utilizarii CNN-urilor este ca nu necesita o cantitate semnificativa de preprocesare a imaginilor[5].

Utilizarea convolutiilor pentru a gestiona matematica din afara sistemului este o mare diferenta intre CNN si o retea neuronala obisnuita. In cel puțin un strat al CNN, se folosește o convolutie in loc de inmultirea matricei. Transformarile produc doua functi si returneaza o functie.

CNN-urile filtreaza datele de intrare. CNN-urile sunt excepționale pentru capacitatea lor de a ajusta filtrele lor în timpul antrenamentului. Chiar și în cazul seturilor de date uriase, cum ar fi fotografiile, rezultatele sunt reglate în timp real in acest fel[5].

1D CNN: cu acestea, nucleul CNN se mișca intr-o directie. CNN-urile 1D sunt utilizate de obicei pentru datele din serii temporale.

2D CNN: Nucleele acestui tip de CNN se misca in doua directii. Acestea sunt folosite in procesarea imaginilor și etichetarea lor.

3D CNN: Un nucleu al acestui tip de CNN se misca in trei directii. Acest tip de CNN este folosit de cercetatori pe imagini 3D, cum ar fi scanarile CT și RMN.

Deoarece CNN-urile sunt de obicei legate de date de imagine, veți vedea de obicei CNN-uri 2D. Acestea sunt câteva dintre aplicațiile în care CNN-urile sunt utilizate.

A picture containing screenshot, diagram, text, line

Description automatically generated

Figure - Retea CNN 2D

# Implementarea metodei

Implementarea metodei de detectare a genului vorbitorului este realizata in limbajul de programare Python, folosind diverse biblioteci precum Pandas, NumPy, Matplotlib, Scikit-learn și TensorFlow.

Pentru o solutie cat mai corecta, implementarea este impartita in 4 mari etape:

## Importarea bazei de date

Codul incepe prin importarea bibliotecilor necesare pentru a rula modelul și pentru prelucrarea datelor. Se incarca baza de date, care este un fisier CSV numit 'voice.csv' si se face o verificare cu ajutorul functiei info() pentru a vedea daca exista valori lipsa. In cazul in care exista valori lipsa, acestea trebuie gestionate in mod corespunzator inainte de a se trece la crearea modelului.

## Prelucrarea parametrilor

In urmatoarea etapa, se aplica o codificare de eticheta pentru a transforma variabila tinta dintr-un set de date nominal(female si male) intr-un set de date numeric(0 si 1). Acest lucru este necesar pentru a putea aplica modelul de invatare automata. Datele sunt impartite in doua parti - o parte este variabila tinta si cealalta parte este matricea de caracteristici.

In continuare, se aplica un scalar pentru a ne asigura ca toate variabilele au aceeași scala.Scopul scalarii este de a imbunatati performanta modelului. Datele sunt impartite in setul de antrenare și setul de test cu ajutorul functie train\_test\_split.

## Modelarea si antrenarea modelelor

### Retea neuronala

Primul model a implicat utilizarea mai multor layere. Fiecare layer are rolul de a transforma datele de intrare într-un format adecvat pentru a fi procesate ulterior. De asemenea, fiecare strat poate include o funcție specifica de activare, care determina modul in care semnalele sunt propagate prin reteaua neuronala.

* 1. Dense layer

In orice retea neuronala, un strat dens este un strat care este profund conectat cu stratul sau precedent, ceea ce inseamna ca neuronii stratului sunt conectati la fiecare neuron al stratului sau precedent. Acest strat este cel mai frecvent utilizat în retelele neuronale artificiale .

Intr-un model, neuronul stratului dens primeste informatii de la fiecare neuron din stratul sau anterior, unde efectueaza multiplicarea matricei-vector.In inmultirea vectorului matrice, vectorul rand al rezultatului din straturile anterioare este egal cu vectorul coloana al stratului dens. Vectorul rand trebuie sa aiba tot atatea coloane ca vectorul coloana, conform regulei generale a inmulțirii matrice-vector[7].

Diagram

Description automatically generated

Figure - Dense layer

Keras Dense Layer prezinta o serie de parametri in definitia sa, insa pentru acest proiectul au fost folosite doar o parte dintre acestia.Pe baza codului de mai jos, o sa se prezinte fiecare parametru.

**x =tf.keras.layers.Dense(64,activation='relu')(x)**

**64**- reprezinta unitatile.Unitatile sunt unul dintre cei mai fundamentali si esentiali parametri ai stratului Dense în Keras, iar acestia definesc dimensiunea iesirii stratului dens. Acest parametru specifica numarul de neuroni din stratul respectiv si determina dimensionalitatea vectorului de iesire. Unitatile trebuie să fie definite ca un numar intreg pozitiv, deoarece reprezinta dimensiunea spațiului per caracteristici sau dimensiunea vectorului de iesire al stratului Dense. Cu cat numarul de unitati este mai mare, cu atat modelul poate captura mai multe informații și poate avea o capacitate mai mare de invatare, insa acest lucru poate creste si complexitatea si timpul de antrenare al retelei.

**Activation**- In retelele neuronale, functia de activare este o functie ce este utilizata pentru transformarea valorilor de intrare ale neuronilor. Practic, introduce neliniaritatea în retelele neuronale, astfel incat retelele sa poata invata relația dintre valorile de intrare și de ieșire.

- O funcție de activare utilizata in rețelele neuronale este **functia ReLU**, care transforma valorile de intrare într-o unitate liniara si elimina valorile negative transformandu-le în zero.

Dupa definirea fiecarui layer, se defineste modelul si se aplica o compilare pentru a specifica metricile de evaluare.

|  |
| --- |
| model1.compile(  optimizer='adam',  loss='binary\_crossentropy',  metrics=[  'accuracy',  tf.keras.metrics.AUC(name='auc')  ]  )  #stocam modelul nostru  history = model1.fit(  X\_train,  y\_train,  validation\_split=0.2,  batch\_size=32,  epochs=100,  callbacks=[  tf.keras.callbacks.EarlyStopping(  monitor='val\_loss',  patience=3,  restore\_best\_weights=True  )  ]  ) |

In definirea modelului, au fost folositi o serie de parametrii care urmeaza sa fie descrisi mai jos.

**a)Batch**: Setul de data de antrenament este impartit in loturi mai mici (batches) pe parcursul antrenarii pentru a fi procesat cu modelul neural. Aceasta se face deoarece procesarea intregului set de date de antrenament intr-o singura iterație poate fi costisitoare din punct de vedere computational și poate necesita mai multa memorie. Cand se alege dimensiune de batch, cum ar fi batch\_size=32, se poate determina cate example de antrenament sunt incluse intr-un singur batch. Modelul primește acest flux si ajusteaza ponderile in functie de erorile calculate pentru exemplele respective in fiecare iteratie a antrenamentului. Apoi se preia si se proceseaza fluxul urmator. Acest proces se repeta pana cand toate loturile din setul de date de antrenament sunt finalizate.

**b)Epoch**:O epoca este o trecere in intregime a setului de data de antrenament. Fiecare exemplu de antrenament dintr-o perioada de timp este prezentat modelului neural, iar ponderile sunt ajustate in functie de erorile calculate. Cu alte cuvinte, un numar de epoci specificat arata de cate ori modelul trece prin intregul set de data de antrenament. Modelul a incercat in fiecare epoca sa invete și sa-si imbunatateasca performanta prin ajustarea ponderilor.

De obicei, in timpul antrenarii, modelul trece prin mai multe epoci, iar in fiecare epoca se foloseste o metoda de optimizare („adam”) pentru a actualiza ponderile și pentru a minimiza functia de pierdere („binar\_crossentropy”). Scopul este sa se obțina o performanta satisfacatoare pe datele de antrenament, precum și o generalizare adecvata pe datele noi. In timpul antrenarii, metricii precum acuratetea și AUC sunt monitorizate și inregistrate. Acest lucru este facut pentru a evalua performanța modelului atat pe setul de data de antrenament, cat si pe setul de date de validare, care au fost specificate prin validation\_split.

**c)Optimizatorii** sunt algoritmi utilizați pentru a modifica și actualiza ponderile modelelor in timpul procesului de antrenare in functie de functia de pierdere.

Adaptive Moment Estimation (Adam) este un algoritm de optimizare utilizat in tehnica de coborare a gradientului. Acesta este extrem de eficient atunci cand se lucreaza cu probleme ample, care implica o cantitate mare de date sau parametri. Algoritmul Adam necesita mai putina memorie si este conceput sa funcționeze în mod eficient.

Pe scurt, Adam poate fi vazut ca o combinatie intuitiva a doi algoritmi de optimizare: coborarea gradientului cu impuls și algoritmul Root Mean Square Propagation (RMSP). El utilizeaza momente de ordinul intai si de ordinul doi pentru a ajusta ponderile, luand în considerare directia și magnitudinea gradientului anterior și estimand variabilitatea acestuia. Astfel, algoritmul Adam poate adapta ratele de invatare in mod individual pentru fiecare parametru, obtinand rezultate mai bune în problemele de invatare automata[10].

**d)Functia de pierderi**- Pentru a antrena modelele , a fost utilizata funcția de pierdere 'binary\_crossentropy'. Aceasta functie urmareste sa reduca diferenta dintre distributia reala a claselor și distributia de probabilitate prezisa de model. Valoarea funcției de pierdere este mai mica cu cat predictiile modelului se apropie de valorile reale.

In timpul antrenarii, algoritmul de optimizare (in acest caz, „Adam”) este responsabil de ajustarea ponderilor modelului pentru a minimiza valoarea funcției de pierdere. In cele din urma, scopul este crearea unui model care sa poata clasifica exemplele in mod corespunzator si sa generalizeze in mod corespunzator la noi date.

**e)Metrica 'accuracy'** (acuratețe) calculeaza procentul de exemple clasificate corect de catre model in raport cu totalul exemplelor evaluate. Este o metrica comuna utilizata in problemele de clasificare si furnizeaza o valoare cuprinsa intre 0 și 1. O acuratete de 1 indica faptul ca toate exemplele au fost clasificate corect, in timp ce o acuratețe de 0 înseamna ca niciun exemplu nu a fost clasificat corect[11].

Prin urmare, in timpul antrenaii și validarii modelului, metrica 'accuracy' este utilizata pentru a evalua performanța acestuia in clasificarea corecta a datelor. Cu cat valoarea 'accuracy' este mai mare, cu atat modelul prezinta o capacitate mai mare de a clasifica exemplele în mod corect.unidimensional

### CNN 2D

In cele din urma, se aplica un strat convolutional bidimensional pentru a transforma datele de intrare. Aceasta transformare se realizeaza prin preluarea unor matrice de date pixeli și preluarea de date din imagini. Datele sunt apoi transformate intr-o matrice mai mica, astfel incat sa fie mai usor de procesat. Modelul este apoi antrenat din nou, deoarece aceasta transformare a datelor necesita un alt tip de model.

In prelucrarea acestui model se foloses o serie de layere.

**2.1 Conv 2D**

Stratul Conv2D reprezinta un tip de strat utilizat in rețelele neuronale convoluționale (CNN) pentru procesarea imaginilor și recunoașterea obiectelor.

Conv2D aplica operatia de convolutie pe o imagine de intrare bidimensionala. Acesta utilizeaza kernel-uri, sau filtre, pentru a extrage caracteristicile specifice ale imaginii. Operația de convoluție implica deplasarea filtrului peste imaginea de intrare și determinarea produsului punctual între valorile filtrului și zona corespunzătoare a imaginii de intrare. Rezultatele sunt apoi insumate si se aplica o funcție de activare, cum ar fi ReLU, asupra sumei. Acest proces generează o nouă imagine, numită harta a caracteristicilor, care poate fi prelucrata ulterior de alte straturi ale retelei neuronale.[9]

Un strat Conv2D are parametri precum numarul de filtre, dimensiunea filtrelor, pasul de deplasare (stride), metoda de completare a marginilor (padding) și funcția de activare. Acești parametri pot fi ajustați in funcție de problemele specifice și de arhitectura retelei neuronale.[9]

A picture containing text, electronics, cellphone

Description automatically generated

Figure - Layer Conv 2D

**x = tf.keras.layers.Conv2D(32, 1, activation='relu')(x)**

**32**: Acesta reprezinta numarul de filtre (kernel-uri) utilizate in stratul Conv2D. Fiecare filtru va detecta și extrage un set de caracteristici specifice din imaginea de intrare. Valoarea 32 indica ca se vor utiliza 32 de filtre în acest strat[8].

**1=kernel\_size**.Acest parametru determina dimensiunile nucleului. Dimensiunile comune includ 1×1, 3×3, 5×5 și 7×7, care pot fi trecute ca tupluri (1, 1), (3, 3), (5, 5) sau (7, 7).Este un numar intreg sau tuplu/listă de 2 numere intregi, specificand inaltimea si lațimea ferestrei de convoluție 2D.Acest parametru trebuie sa fie un numar intreg impar[8].

* 1. **MaxPooling2D layer**

Stratul MaxPooling2D este utilizat in retelele neuronale convolutionale pentru a reduce dimensiunea hartilor de caracteristici si a extrage caracteristicile importante ale imaginii de intrare. Acest strat imparte harta caracteristicilor in regiuni disjuncte si extrage valoarea maxima din fiecare regiune, rezultand o harta caracteristica redusa dimensional. Aceasta tehnica are mai multe beneficii, cum ar fi reducerea numarului de parametri ai retelei, ceea ce duce la o eficiența sporita in timpul antrenarii si inferensei, și prevenirea supraadaptarii prin eliminarea detaliilor redundante.

Pentru alegerea dimensiunii de subesantionare (pooling) si a dimensiunii deplasarii (stride), trebuie avute în vedere cateva consideratii importante. Daca dimensiunea de subesantionare este prea mare, se poate pierde informația fina si detalii importante ale imaginii. Pe de alta parte, daca dimensiunea de subesantionare este prea mica, se poate obtine o harta caracteristica prea mica, in care informația esentiala este comprimată excesiv. Prin urmare, alegerea dimensiunii de subeșantionare și a dimensiunii deplasarii trebuie facuta cu atenție, luand in considerare complexitatea imaginii si specificul problemei de rezolvat in cadrul retelei neuronale convolutionale.[9]

A picture containing table

Description automatically generated

Figure - MaxPooling2D layer

**2.3 Flatten layer**

Rețelele neuronale artificiale folosesc stratul Flatten pentru a converti un tensor multidimensional intr-un vector unidimensional. Este folosit in principal pentru a conecta straturi de invatare profunde care genereaza tensori multidimensionali cu straturi care necesita vectori unidimensionali, cum ar fi straturile dense (straturi complet conectate).

In loc sa modifice valorile tensorului de intrare, stratul de flattenare le rearanjeaza și le transforma intr-un vector unidimensional. De exemplu, stratul ingust va transforma tensorul de intrare intr-un vector unidimensional cu dimensiunile [batch\_size, numar\_canale, inaltime, latime]. Acest strat nu are parametri de invatare proprii și nu influențeaza decat rearanjarea valorilor tensorului de intrare intr-un vector unidimensional.

Rețelele neuronale convoluționale (CNN) folosesc frecvent straturile subeșantionare de convoluție pentru a le conecta cu straturile dense ce urmeaza. Straturile dense pot fi procesate și utilizate pentru clasificare sau regresie prin transformarea caracteristicilor din imagini sau alte date multidimensionale intr-un format unidimensional.

## Testarea

In ceea ce privește testarea, s-au adoptat doua metode diferite pentru cei doi algoritmi. Inițial, s-a verificat algoritmul bazat pe retele neuronale 2D unde s-a selectat o inregistrare din baza de date de verificare și s-a verificat daca eticheta prezisa coincide cu eticheta reala. Pentru algoritmul bazat pe rețele neuronale TensorFlow, s-a utilizat o inregistrare personală a propriei voci. S-a aplicat o serie de prelucrari asupra acestei inregistrari pentru a extrage caracteristicile vocii și s-a efectuat o verificare a etichetei.

In plus fața de prelucrarile bazate pe algoritmi, s-au afișat și spectrogramele pentru primele 9 inregistrari.

# Rezultate

In tabelul urmator exista valorile a trei metrici importante pentru evaluarea performanței a celor doi algoritmi: LOSS, ACCURACY și AUC. Numele celor doi algoritmi sunt prezentate in prima coloana, iar celelalte trei coloane reprezintă valorile pentru LOSS, ACCURACY și AUC pentru fiecare algoritm..

| Algoritmul | Parametrii | | |
| --- | --- | --- | --- |
| LOSS | Accuracy | Auc |
| Retele neuronale | 0.0645177289843 | 0.9821240901947021 | 0.997222304344772 |
| CNN 2D | 0.11314608156681061 | 0.9537329077720642 | 0.9925647974014282 |

Table 1- Compararea rezultatelor

## Retele neuronale TensorFlow:

### Curba de invatare- acurateta modelului

Din graficul urmator se poate observa ca, la inceputul antrenarii acuratetea este relativ scazuta si creste foarte repede in primele epoci, deoarece modelul invata sa generalizeze si sa faca predictii mai precise pe datele de antrenare.

Dupa aceasta crestere abrupta, acuratetea continua sa creasca treptat in urmatoarele epoci, dar cu o rata mai lenta. Aceasta se datoreaza faptului ca modelul se apropie din ce in ce mai mult de optimul global, ceea ce face ca imbunatatirile sa fie mai mici si mai dificil de obtinut.

In cele din urma, dupa aproximativ 8 epoci, acuratetea se stabilizeaza si atinge valoarea maxima de 0.991. Aceasta valoare reprezinta performanta modelului pe datele de test, ceea ce inseamna ca modelul este capabil sa faca predictii precise cu o acuratete de aproximativ 99%.

Chart

Description automatically generated

Figure -Curba de invatare-Acuratetea modelului

## CNN 2D :

### Reprezentarea valorilor sub forma de imagini

S-a creat o reprezentare sub forma de imagini pentru primele 9 elemente dintr-un set de date. Fiecare imagine este formata dintr-o matrice de dimensiune 3x3, care este impartita in 5 linii și 5 coloane de cadrane mai mici. Ultima linie a imaginii este formată din culori solide, deoarece reprezintă valorile zero din setul de date. Fiecare cadran are o culoare sau o intensitate diferita, și aceasta poate fi interpretata astfel: cu cat culoarea este mai stralucitoare, cu atat valoarea din setul de date este mai mare. Astfel, se obtine o harta a culorilor: culoarea solida reprezinta zerourile, culoarea mai inchisa reprezinta valorile negative, iar culorile mai deschise reprezinta valorile pozitive. Fiecare patrat din imagine este o reprezentare a unei valori din setul de date inițial, deci exista valori diferite, iar fiecare este reprezentat de un patrat nou. Chiar daca aceste imagini par a fi doar niște culori, pentru algoritm ele reprezintă inca un set de date numerice.

Background pattern

Description automatically generated

Figure - Reprezentarea valorilor sub forma de imagini

### Curba de invatare- acurateta modelului

Graficul urmator este o reprezentare a curbei de invatare a celui de-al doilea model. In primele patru epoci, modelul pare sa imbunatateasca rapid acuratețea, ajungand la o acuratețe de 0.90. Apoi, pentru urmatoarele intervale de epoci, se observa cresteri continue, dar mai mici, ale acurateții, sugerand ca modelul nu se mai imbunatateste la fel de rapid ca inainte.In cele din urma, la epoca 30, modelul pare sa fi ajuns la o acuratețe de 0.9558, ceea ce poate fi considerat un nivel ridicat de performanță,

Chart

Description automatically generated

Figure - Curba de invatare- Acuratetea modelului

### Spectogramele

Scala de culori este o reprezentare a intensitatii semnalului, unde culorile deschise corespund intensităților mai mici, iar culorile mai inchise corespund intensităților mai mari. In cazul spectrogramei, culorile deschise arata zonele cu putina energie, in timp ce culorile mai inchise arata zonele cu energie mai mare. De exemplu, zonele albastre reprezinta zonele cu energie scazuta, in timp ce zonele portocalii si rosii reprezinta zonele cu energie ridicata. Prin urmare, scala de culori ajuta sa se interpreteze nivelul de energie și distribuția frecvențelor in semnalul existent.

A picture containing bar chart

Description automatically generated

Figure -Spectrogramele

## Testarea algoritmilor:

In cadrul testarii celor doi algoritmi, s-au utilizat doua metode diferite. Pentru primul model,s-a inregistrat o voce feminina și s-au extras parametrii corespunzatori acestuia din baza de date inițiala. Apoi, s-a realizat predicția pe baza acestor parametri. Pentru al doilea model, s-a selectat un sample din baza de date de testare și s-a efectuat predicția pe acesta.S-au obținut rezultatele așteptate utilizând ambele metode.

# Concluzii

Dupa performanța superioara a primului algoritm in ceea ce privește acuratețea și stabilitatea in procesul de antrenare se indica faptul ca acesta este capabil sa faca predictii mai precise și sa obtina rezultate mai constante pe parcursul antrenarii. Acuratețea ridicata sugereaza ca modelul este capabil sa clasifice corect majoritatea datelor de test.

Analizand tabelul cu metrici (LOSS, ACCURACY și AUC), se observa ca primul algoritm inregistreaza valori mai mici pentru LOSS (pierdere), ceea ce indica o capacitate mai buna de ajustare a modelului la datele de antrenare. De asemenea, primul algoritm obține valori mai mari pentru ACCURACY (acuratețe), indicand o mai mare precizie în clasificarea datelor, si valori mai mari pentru AUC (aria de sub curba ROC), care este o măsură a performanței în clasificarea binara. Aceste rezultate sugereaza ca primul algoritm este mai eficient in rezolvarea problemei in discutie.

Rezultatul pozitiv obținut in demo sample, unde modelul a realizat o predicție corectă pentru un eșantion din setul de date de validare, sporește increderea in performanța algoritmului. Acest rezultat sugereaza ca modelul este capabil sa generalizeze și să faca predicții precise nu doar pe datele de antrenare, ci si pe date noi.

In concluzie, avand in vedere performanța superioara a primului algoritm in ceea ce priveste acuratetea, stabilitatea in procesul de antrenare, valorile mai bune in metrici și rezultatul pozitiv in demo sample, acesta poate fi considerat o alegere mai buna pentru a rezolva problema specifica. Utilizarea primului algoritm ar oferi o soluție mai eficienta, mai precisa și mai fiabila pentru nevoile proiectului.

# Referinte

1. G. . D. Doukhan, J. Carrive, F. Vallet, A. Larcher and S. Meignier, "An Open-Source Speaker Gender Detection Framework for Monitoring Gender Equality," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018, pp. 5214-5218, doi: 10.1109/ICASSP.2018.8461471.
2. JA. Graves, A. -r. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 2013, pp. 6645-6649, doi: 10.1109/ICASSP.2013.6638947.
3. O. Abdel-Hamid, A. -r. Mohamed, H. Jiang, L. Deng, G. Penn and D. Yu, "Convolutional Neural Networks for Speech Recognition," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, no. 10, pp. 1533-1545, Oct. 2014, doi: 10.1109/TASLP.2014.2339736.
4. S. Basu, J. Chakraborty and M. Aftabuddin, "Emotion recognition from speech using convolutional neural network with recurrent neural network architecture," 2017 2nd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2017, pp. 333-336, doi: 10.1109/CESYS.2017.8321292
5. <https://www.freecodecamp.org/news/convolutional-neural-network-tutorial-for-beginners/>
6. García-Ordás, M. T., Benítez-Andrades, J. A., García-Rodríguez, I., Benavides, C., & Alaiz-Moretón, H. (2020). Detecting respiratory pathologies using convolutional neural networks and variational autoencoders for unbalancing data. Sensors, 20(4), 1214.
7. <https://analyticsindiamag.com/a-complete-understanding-of-dense-layers-in-neural-networks/>
8. <https://www.geeksforgeeks.org/keras-conv2d-class/>
9. Karpathy, A. (2017). Convolutional neural networks for visual recognition. *Notes accompany the Stanford CS class CS231*
10. <https://www.geeksforgeeks.org/intuition-of-adam-optimizer/>
11. <https://www.iguazio.com/glossary/model-accuracy-in-ml/>