

UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAŞI  
FACULTATEA DE INFORMATICĂ



Evaluarea pronunției în aplicații de învățare a limbilor străine

**Autor:** Camelia Georgiana Stativă

**Sesiunea:** iulie, 2021

**Coordonator științific:** Prof. Dr. Adrian Iftene

**UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAŞI**

**FACULTATEA DE INFORMATICĂ**

Evaluarea pronunției în aplicații de învățare a limbilor străine

Camelia Georgiana Stativă

**Sesiunea: iulie, 2021**

**Coordonator științific**  
Prof. Dr. Adrian Iftene

Avizat,

**Îndrumător Lucrare de Licență**

Titlul, Numele și prenumele **Prof. Dr. Iftene Adrian**

Data \_\_\_\_\_ Semnătura \_\_\_\_\_

**DECLARAȚIE privind originalitatea conținutului lucrării de licență**

Subsemnata **Stativa Camelia Georgiana** domiciliul în str. **Dorului, nr. 127, sat Dancu, com. Holboaca, jud. Iași** născut(ă) la data de **16/10/1999**, identificat prin CNP **2991016226731.**, absolventă a Universității „Alexandru Ioan Cuza” din Iași, Facultatea de **Informatică** specializarea **Română**, promoția **2021**, declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul: **Evaluarea pronunției în aplicații de învățare a limbilor străine**, elaborată sub îndrumarea **dl. Conf. Dr. Iftene Adrian**, pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimtind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diplomă sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am intreprins-o.

Date azi, 21.06.2021

Semnătură student



## DECLARAȚIE DE CONSUMĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „Evaluarea pronunției în aplicații de învățare a limbilor străine”, codul sursă al programelor și celealte conținuturi (grafice, multimedia, date de test etc.) care însotesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, 21.06.2021

Absolvent Prenume Nume

Stativă Camelia -Georgiana

(semnătura în original) 

# Cuprins

<b>1. Introducere</b>	<b>7</b>
<b>2. Învățare limbilor străine asistată de mobil</b>	<b>9</b>
2.1 Introducere	9
2.2 Avantaje	9
2.3 Dezavantaje	10
2.4 Aplicații existente	10
2.5 Concluzii	13
<b>3. Metode de evaluarea a pronunției</b>	<b>14</b>
3.1 Introducere	14
3.2 Evaluarea pronunției utilizând recunoașterea fonetică	14
3.2.1 Sistem de Recunoaștere Vocală	15
3.2.2 Evaluare pronunției	18
3.3 Evaluare pronunției utilizând Rețele Neuronale Convoluționale	20
3.3.1 Rețele Neuronale Convoluționale pentru clasificare audio	20
3.3.2 Învățarea prin transfer	23
3.3.3 Evaluarea pronunției	24
3.4 Concluzii	26
<b>4. Tehnologii utilizate</b>	<b>28</b>
4.1 Android	28
4.2 Firebase	28
4.3 CMU Sphinx	29
4.4 Praat	30
4.5 Montreal Forced Aligner	30
4.6 Tensorflow	31
4.6.1 Keras	31
4.7 Concluzii	32
<b>5. Dezvoltarea aplicației</b>	<b>33</b>
5.1 Introducere	33
5.2 Arhitectura aplicației	33
5.2.1 Introducere	33
5.2.2 Modulul de autentificare și tipurile de utilizatori	34

5.2.3 Modulul de evaluare a pronunției	35
5.2.4 Modulul de dezvoltare al vocabularului	40
5.2.5 Modulul de asignare și de rezolvare a temelor	40
5.3 Evaluarea pronunției utilizând recunoașterea fonetică	43
5.3.1 Configurări	43
5.3.2 Experimente	45
5.3.3 Concluzii	46
5.4 Evaluare pronunției utilizând Rețele Neuronale Convoluționale	47
5.4.1 Preprocesare date	47
5.4.2 Modele pre-instruite utilizate	50
VGG16	51
ResNet50	51
5.4.3 Experimente	52
5.4.4 Concluzii	60
<b>6. Evaluare și feedback din partea utilizatorilor</b>	<b>62</b>
6.1 Profilul utilizatorilor	62
6.2. Interesul față de învățarea unei limbii străine asistată de mobil	63
6.3 Experiența din timpul utilizării aplicației propuse	66
6.4 Îmbunătățiri	70
6.5 Concluzii	71
<b>7. Concluzii finale</b>	<b>72</b>
7.1. Concluzii	72
<b>8. Bibliografie</b>	<b>74</b>

# 1. Introducere

Pronunția este o componentă esențială în învățarea unei noi limbi și are capacitatea de a face diferență între un vorbitor experimentat și unul cu mai puțină experiență. Perfecționarea pronunției este unul dintre cele mai dificile aspecte în învățarea unei limbi străine, deoarece necesită practică constantă, preferabil alături de vorbitori nativi, dar și pentru că este foarte ușor influențată de particularitățile limbii native a celui care ia parte la acest demers.

Lucrarea prezintă un exemplu de aplicație pentru un dispozitiv mobil, care poate fi folosită atât de adulți, cât și de copii în vederea învățării, respectiv a exersării pronunției într-o limbă de circulație internațională, mai exact limba engleză. Componenta principală va fi reprezentată de evaluare pronunției și o parte din lucrare se va focaliza pe două metode prin care se poate realiza aceasta și beneficiile aduse de fiecare metodă descrisă. Aceste două metode descrise vor fi integrate în aplicație astfel, vom propune utilizatorului o serie de exerciții orientate în principal pe reproducerea unor cuvinte în limba respectivă, care au ca scop atât dezvoltarea vocabularului, cât și punerea în evidență a greșelilor de pronunție. Cel care folosește aplicația are parte de exersare continuă, dar și de feedback, doi factori importanți în îmbunătățirea pronunției într-o limbă străină.

Aplicația urmărește să împlinească nevoile ambilor tipuri de utilizatori, adulți și copii, în aşa fel încât procesul de învățare să fie unul cât se poate de echilibrat. Modulul destinat copiilor este orientat spre învățarea de cuvinte simple, uzuale, întâlnite frecvent în conversațiile zilnice (animale, numere) și are la bază o interfață interactivă și prietenoasă, bazată pe provocări care să antreneze atenția și interesul copilului și care să transforme procesul de asimilare a cunoștințelor într-o activitate plăcută. Având în vedere cuvintele prezentate drept exerciții, aplicația poate servi și drept suport în deprinderea limbii materne în cazul copiilor proveniți din familii vorbitoare de limba engleză, cât și pentru identificarea eventualelor probleme de vorbire ale acestora prin analiza de către o persoană avizată a parcursului avut în cadrul aplicației.

În ceea ce privește modulul destinat adulților, se urmărește perfecționarea pronunției. Acest modul poate fi o unealtă utilă atât pentru cei care utilizează engleza drept limbă secundară, având în vedere că pronunția corectă este o deprindere care necesită o atenție specială și o antrenare continuă, lucru care cu greu se poate realiza în cadrul unor cursuri la care sunt prezenți mai mulți elevi, cât și

pentru cei care vorbesc engleza drept limbă principală, punând în evidență anumite greșeli apărute datorită unui accent specific sau chiar a unor probleme de vorbire.

Ambele module, vor oferi posibilitatea utilizatorului de a asculta o simulare a pronunției cu ajutorul propriei voci, această funcționalitatea are scopul de a-l ajuta pe utilizator să compare modul în care acesta vorbește în mod uzual, cu pronunția corectă, disponibilă și ea în aplicație.

De asemenea, pentru a promova adaptarea alternativelor de învățare și perfecționare în mediile tradiționale aplicația are inclus și un modul prin care un student este asignat unui clase și poate primi teme prin intermediul acesteia, teme a căror răspuns este trimis ulterior spre a fi analizat de un profesor sau de o persoana avizata în acest domeniu care dorește să urmărească procesul de învățare a respectivului student și care poate oferi sfaturi și o analiză mai amănunțită a datelor oferite drept răspuns de către aplicație.

Lucrarea este structurată în următoarele cinci capitulo menite să pună în evidență atât utilitatea acestor aplicații și precedentul care există în acest domeniu, cât și detaliile particulare legate de implementare și tehnologii folosite:

- **Primul capitol** prezintă conceptul de învățare a limbilor străine asistată de mobil și direcția spre care se îndreaptă acest domeniu.
- **Al doilea capitol** înfățișează o serie de aspecte legate de tehnologiile existente la momentul actual pentru a evalua abilitățile de pronunție într-o anumită limbă. Vor fi prezentate cele două metode de evaluare a pronunției propuse, aspectele teoretice necesare pentru implementarea acestora, dar și avantajele și dezavantajele asociate fiecărei abordări.
- **Al treilea capitol** ilustrează tehnologiile utilizate în realizarea aplicației, atât în ceea ce privește utilitatea acesteia ca aplicație de învățare a limbilor străine asistată de mobil, cât și în ceea ce privește implementarea celor două metode de evaluare a pronunției.
- **Al patrulea capitol** prezintă arhitectura aplicației. Este pus aici în evidență întregul proces de implementare, pornind cu procesarea și colectarea datelor necesare, până la serviciile principale și la cum sunt acestea integrate în aplicația finală. În acest capitol vor fi prezentate și experimentele realizate cu cele două metode de evaluare a pronunției abordate.
- **Al cincilea capitol** expune un feedback general legat de aplicație, o evaluare realizată de către utilizatori în urma unui test de uzabilitate.
- **Al șaselea capitol** prezintă concluziile finale și respectiv direcțiile spre care poate fi dezvoltată aplicația propusă și serviciile oferite de aceasta.

## **2. Învățare limbilor străine asistată de mobil**

### **2.1 Introducere**

Învățarea limbilor străine asistată de aplicații de pe dispozitive mobile este un precursor natural al învățării limbilor străine asistată de calculator și se înscrie în curentul general al momentului actual de introducere a tehnologiei în procesul de învățare. Noile tehnologii apărute în domeniul dispozitivelor mobile, accesibilitatea acestora, cât și faptul că au devenit o parte importantă, aproape indispensabilă în viața omului modern, fac din acestea o unealtă foarte eficientă, care vine în completarea modului clasic de învățare a unui limbi noi și acoperă unele din lipsurile acestuia. Una din primele inițiative în folosirea dispozitivelor mobile în învățarea limbilor străine a fost dezvoltată de Universitatea Stanford în cadrul unui program de învățare a limbii spaniole în 2001 (Brown, 2001), se consideră totuși că au existat și alte încercări în acest domeniu, mai exact, prin utilizarea dispozitivelor mobile și a teleconferințelor în procesul de învățare la distanță.

### **2.2 Avantaje**

Georgiev, Georgieva și Smrikarov descriu învățarea asistată de mobil ca “Abilitatea de a învăța oriunde și oricând fără o conexiune fizică permanentă la o rețea de cabluri” [1]. Acest citat înglobează utilitatea metodelor alternative de predare care utilizează dispozitivele mobile, punând în evidență avantajul principal al acestui tip de învățare, mai exact faptul că depășește barierele spațiale și temporale și deschide posibilitatea exersării și cultivării unor abilități și cunoștințe indiferent de locația și momentul zilei, cât timp există accesul la un dispozitiv mobil și respectiv în unele cazuri la o conexiune de internet.

Un alt aspect important este faptul că prin învățarea asistată de instrumente mobile, dobândirea unor cunoștințe se depărtează de metodele standard, care nu pot fi eficiente pentru toți, și individualizează acest demers prin posibilitatea de a alege aplicația cea mai adecvată pentru profilul utilizatorului, aplicație care poate încorpora și o activitate plăcută și relaxantă cum ar fi jocurile, muzica sau alte domenii de interes pentru acesta.

## **2.3 Dezavantaje**

Problema generală legată de numărul mare de resurse și aplicații disponibile la momentul actual, se manifestă și în cazul învățării cu ajutorul dispozitivelor mobile, mai exact, problema filtrării informațiilor pentru a putea alege ceva cu o utilitate reală, care să ne ajute să evoluăm. Astfel, utilizatorul care pornește pe acest drum trebuie să reușească să nu se lase copleșit de numărul mare de informații și să aleagă o metodă cu rezultate demonstrate și care să se plieze pe modul acestuia de învățare.

De asemenea, fie că e utilizată în cadrul unei instituții de învățământ, fie că apare ca o inițiativă personală, învățarea ajutată de mobil necesită o mai bună gestionare a motivației și o autoorganizare eficientă, comparativ cu procedeele clasice de învățare care sunt încurajate și susținute într-o oarecare măsură și de factori externi.

## **2.4 Aplicații existente**

Cum am menționat în partea de introducere învățarea asistată de dispozitive mobile a început prin exploatarea teleconferințelor pentru învățarea la distanță. Ulterior, odată cu dezvoltarea funcționalităților, în special odată cu apariția telefoanelor inteligente, s-au dezvoltat și metodele de învățare, la momentul actual fiind exploataate din ce în ce mai multe din funcțiile puse la dispoziție de un telefon mobil intelligent. La momentul actual există aplicații pasive care oferă informațiile necesare învățării unei noi limbi în format text, video, audio, dar și aplicații care oferă posibilitatea de a participa activ la procesul de învățare prin evaluarea în acest mod a abilităților utilizatorului, utilizator care interacționează cu acestea în special prin intermediul microfonului, dar și prin date de intrare de tip text, în funcție de abilitatea care este evaluată.

Având în vedere tema principală a lucrării, identificarea problemelor de pronunție în procesul de învățare al unei limbi străine cu ajutorul unei aplicații mobile, ne vom axa în continuarea pe aplicații în care interacțiunea se realizează prin date de tip audio, întrucât aceasta este una din singurele metode prin care pot fi evaluate abilitățile în ceea ce privește pronunția.

**Rosetta Stone**<sup>1</sup> este o aplicație populară atât în domeniul învățării asistată de mobil, dar și în ceea ce privește învățarea asistată de calculator. Aplicația oferă un număr mare de funcționalități și de lecții pentru învățarea unei limbi noi. Într-o lecție sunt inițial prezentate concepte de bază, ulterior lecția fiind separată în exerciții scurte de maxim 10 minute, care se focusează pe un anume aspect, gramatică, vocabular, pronunție. Popularitatea aplicației se leagă în special de motorul de speech recognition utilizat TruAccent, motor care compară vocea utilizatorilor cu cea a vorbitorilor nativi, cât și cu cea a vorbitorilor nenativi, și care oferă feedback instant în ceea ce privește evaluarea pronunției, astfel odată cu înaintarea în lecții utilizatorul are posibilitatea de a crește nivelul de acuratețe pentru a finisa greșeli din ce în ce mai subtile (Rosetta Stone, n.d.).

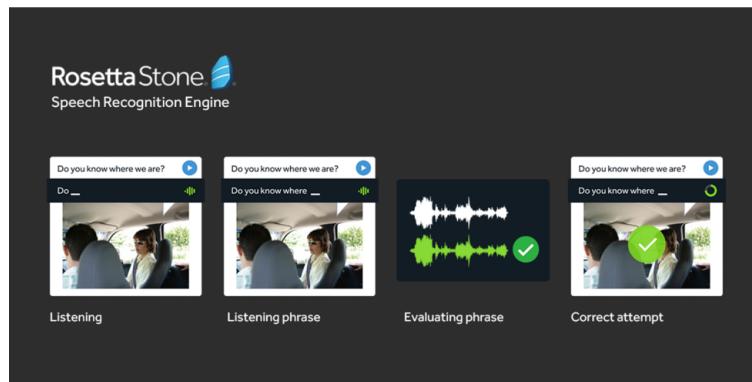


Fig. 2.1: Evaluare pronunție Rosetta Stone

**Babbel**<sup>2</sup> este o altă aplicație cu o istorie bogată în acest domeniu, fondată în Germania, aplicația este disponibilă pe bază de abonament pentru iOS și Android și pune la dispoziție lecții în 14 limbi. Conținutul este realizat de echipe formate din peste 100 de pedagogi și lingviști. Lecțiile sunt împărtășite pe nivele de dificultate și la fel ca în cazul Rosetta Stone sunt cuprinse mai multe aspecte cum ar fi gramatica, vocabularul și respectiv pronunția. La momentul lansării aplicației aceasta nu conținea un modul destinat pronunției, acesta fiind adăugat doi ani mai târziu. De asemenea, îmbunătățirea pronunției se face cu ajutorul oferirii de feedback, utilizatorul ascultă o frază trebuie să o repete, ulterior primește un scor cuprins între 0 și 100 care reprezintă calitatea pronuntiei. Un scor de peste 50 reprezintă o pronunție inteligibilă [2].

<sup>1</sup> Mai multe informații despre Rosetta Stone pot fi găsite pe pagina oficială, disponibilă la adresa <https://www.rosettastone.com/>

<sup>2</sup> Mai multe informații despre Babbel pot fi găsite pe pagina oficială a aplicației, disponibilă la adresa <https://www.babbel.com/>

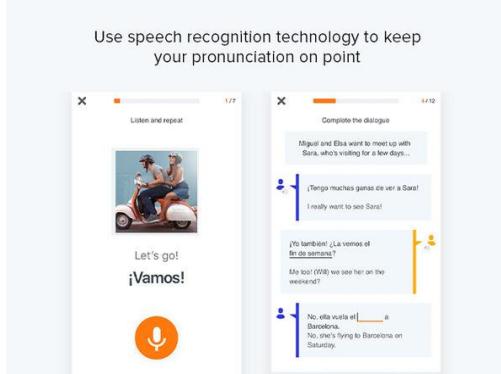


Fig 2.2: Exerciții evaluare pronunție Babbel<sup>3</sup>

**Duolingo**<sup>4</sup> este una din cele mai descărcate aplicații în domeniul educației la nivel mondial. La fel ca în cazul aplicațiilor menționate anterior oferă suport pentru mai multe limbi, iar procesul de învățare se bazează pe rezolvarea de exerciții a căror dificultate crește progresiv odată cu avansare în lecții. De asemenea, o lecție cuprinde mai multe componente importante în învățarea unei limbii, gramatică, vocabular, pronunție. Motivația de a continua este susținută printr-o structurarea a nivelelor foarte asemănătoare cu aceea a unui joc, se bazează pe recompense constante. Funcționalitățile de bază ale aplicației sunt gratis, dar există abonamente pentru funcții suplimentare. Aplicația conține exerciții de pronunție care folosesc propriul motor de recunoaștere vocală.

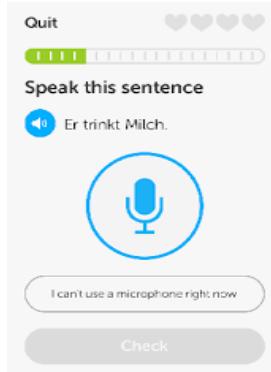


Fig 2.3: Exercițiu pronunție Duolingo, limba engleză

<sup>3</sup> <https://www.joyus.com/sales/babbel-lifetime-subscription-all-languages>

<sup>4</sup> Mai multe informații despre Duolingo, pot fi găsite pe pagina oficială, disponibilă la adresa <https://ro.duolingo.com/>

## **2.5 Concluzii**

Având în vedere aspectele enunțate mai sus putem spune că domeniul învățării unei limbi străine asistată de mobil este un domeniu în continuă dezvoltare, care își găsește indubitabil din ce în ce mai mult locul în activitățile clasice de predare și învățare.

Deși există o serie de probleme în ceea ce privește acest tip nou de învățare, simplu fapt că reușește să adune experiența și cunoștințele unor experți și să o pună la dispoziția oricui, la orice oră și în orice moment, ne spune că acesta este un domeniu care necesită în continuare explorat și îmbunătățit.

### **3. Metode de evaluarea a pronunției**

#### **3.1 Introducere**

Învățarea unei limbi străine presupune dobândirea de abilități în următoarele patru arii: citire, ascultare, scriere și vorbire. Dintre acestea primele trei au putut fi incluse natural în ceea ce înseamna învățarea asistată de calculator și de mobil. Computerele sunt totuși mai puțin versatile în ceea ce privește antrenarea recunoașterii vocale, iar acest lucru se datorează dificultății cu care se realizează procesarea de semnale audio, semnale care variază de la vorbitor la vorbitor (stilul de vorbit, viteza cu care vorbește, regiunea din care provine) și sunt foarte ușor influențate de mediul (distanța până la microfon, zgomot de fundal, ecou) și de dispozitivul cu care se realizează capturarea lor [3].

Procesarea de semnale audio este deja încorporată în viața noastră de zi cu zi, prin aplicații și dispozitive precum Siri, Google Assistant, Alexa, dar rămâne în continuare un domeniu care nu și-a atins potențialul maxim și se află într-un proces continuu de îmbunătățire și inovare, iar dificultatea acestui proces este dată de factorii mai sus menționați, la care se adaugă probleme legate de ambiguitatea limbajului și de anume particularități ale acestuia, cum ar fi cuvintele omofone, cuvinte care se scriu la fel, dar se pronunță diferit (cuminte și cu minte, deal și de-al).

Corectarea pronunției și oferirea de feedback individualizat cu ajutorul aplicațiilor mobile și web, poartă numele de CAPT (Computer Assisted Pronunciation Training) și este un domeniu care beneficiază de pe urma dezvoltării și cercetării realizată în ultimii ani, atât asupra tehnologiilor de tip ASR (Automated Speech Recognition), cât și în ceea ce privește aria inteligenței artificiale.

#### **3.2 Evaluarea pronunției utilizând recunoașterea fonetică**

Recunoașterea fonetică este procesul prin care în loc să convertim o secvență audio într-o transcriere care să conțină cuvintele care o compun, convertim secvența audio în transcrierea fonetică corespunzătoare. Într-o transcriere fonetică fiecărui sunet îi este asociat un simbol, simbol care ne indică modul în care trebuie să fie pronunțat acel sunet. De-a lungul timpului au apărut mai multe

sisteme de transcriere, dar cel mai utilizat la momentul actual este alfabetul fonetic internațional<sup>5</sup>, alfabet al cărui scop este, conform Wikipedia, “*să poată reprezenta în mod unic și precis toate procesele fonologice<sup>6</sup> din toate limbile*”. Un alt sistem de transcriere fonetică utilizat frecvent în domeniul sintetizării de voce este ARPAbet<sup>7</sup>, sistem care este utilizat de Carnegie Mellon University în realizarea CMU Pronouncing Dictionary, un dicționar care mapează peste 134.000 de cuvinte la pronunția Nord Americană.

IPA Symbol	ARPAbet Symbol	Word	IPA Transcription	ARPAbet Transcription
[p]	[p]	parsley	[ˈparsli]	[p a a r s l i]
[t]	[t]	tarragon	[ˈtærəgən]	[t ae r a x g aa n]
[k]	[k]	catnip	[ˈkætnip]	[k ae t n ix p]
[b]	[b]	bay	[beɪ]	[b ey]
[d]	[d]	dill	[dɪl]	[d ih l]
[g]	[g]	garlic	[ˈgarlik]	[g aa r l ix k]
[m]	[m]	mint	[mɪnt]	[m ih n t]
[n]	[n]	nutmeg	[ˈnʌtmeg]	[n ah t m eh g]
[ŋ]	[ŋ]	ginseng	[dʒɪnseɪn]	[jh ih n s ix ng]
[f]	[f]	fennel	[ˈfēnl]	[f eh n el]
[v]	[v]	clove	[kləʊv]	[k l ow v]

Fig 3.1: Transcrierea unei serii de cuvinte utilizând IPA și ARPAbet<sup>8</sup>

Idea pe care se bazează evaluare pronunției cu ajutorul recunoașterii fonetice este următoarea: extragem transcrierea fonetică din segmentul audio și o comparăm cu transcrierea fonetică corespunzătoare textului înregistrat. Astfel putem observa dacă un fonem a fost înlocuit cu altul sau dacă au fost adăugate sau eliminate foneme în timpul pronunției textului.

### 3.2.1 Sistem de Recunoaștere Vocală

Una din metodele de obținere a unei transcrieri fonetice este cu ajutorul unui sistem de recunoaștere vocală. În ceea ce privește aplicațiile disponibile pentru realizarea acestui lucru enumerăm: CMU Sphinx, HTK, Kaldi și DeepSpeech.

În cadrul acestei lucrări în vederea realizării experimentelor vom utiliza CMU Sphinx, drept urmare în continuare vom prezenta modul de funcționare a unui astfel de sistem de recunoaștere vocală, mai exact, un sistem de recunoaștere bazat pe Modele Markov Ascunse.

<sup>5</sup> Mai multe despre IPA: [https://en.wikipedia.org/wiki/International\\_Phonetic\\_Alphabet](https://en.wikipedia.org/wiki/International_Phonetic_Alphabet)

<sup>6</sup> Ramură a lingvisticii care se ocupă cu studiul sunetelor dintr-o anumită limbă

<sup>7</sup> Mai multe despre ARPAbet: <https://en.wikipedia.org/wiki/ARPABET>

<sup>8</sup> <https://www.cis.lmu.de/~micha/praesentationen/rechtschreibkorrektur/GrundideeSpracherkennung.html>

Un sistem de recunoaștere vocală este compus în general din următoarele componente principale: o componentă care extrage caracteristicile fișierului audio, un dicționar de pronunție, un model acustic și un model de limbă, care împreună poartă numele de knowledge database și un decodor [5].

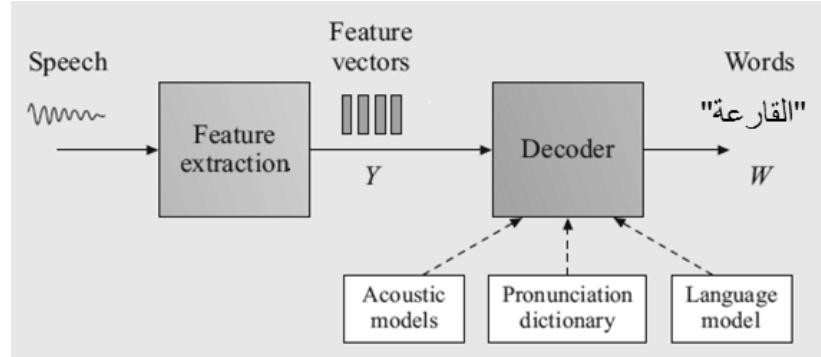


Fig. 3.2: Arhitectura unui sistem de recunoaștere vocală bazat pe Hidden Markov Model<sup>9</sup>

Procesul de extragere a caracteristicilor fișierului audio este un proces de preprocesare a semnalului sonor și de transformare a acestuia în vectori, care conțin informații relevante cu privire la fragmentul audio interceptat. Cele mai multe sisteme de recunoaștere vocală moderne utilizează parametrii mel cepstrali - MFCC (Mel frequency cepstral coefficients) [8] cu număr variabil de parametri drept caracteristici extrase din semnalul sonor. MFC (Mel-frequency cepstrum) este o reprezentare compactă a spectrului<sup>10</sup> unui semnal audio, iar MFCC (Mel frequency cepstral coefficients) sunt coeficienți care împreună formează un MFC.

The resulting MFCCs:

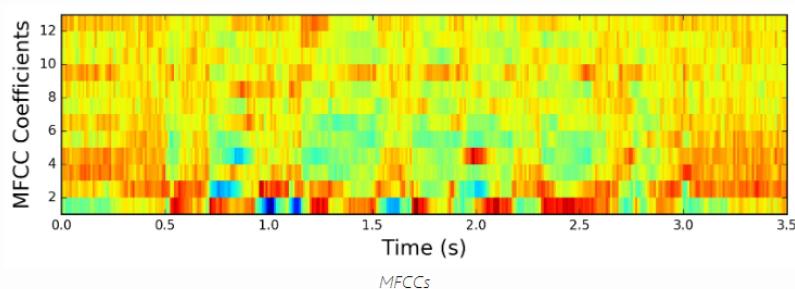


Fig. 3.3:Coeficientii MFCC pentru o secvență dintr-un semnal sonor<sup>11</sup>

<sup>9</sup> [https://www.researchgate.net/figure/Architecture-of-an-HMM-based-recognizer\\_fig4\\_222942097](https://www.researchgate.net/figure/Architecture-of-an-HMM-based-recognizer_fig4_222942097)

<sup>10</sup> Spectrul reprezintă frecvențele care sunt combinate pentru a obține un anume sunet

<sup>11</sup> <https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd>

Knowledge base este sursa folosită de decodor pentru a putea obține transcrierea segmentului audio primit la intrare sub formă de vectori care conțin caracteristici ale semnalului sonor. Este compus din dicționarul de pronunție, modelul acustic și modelul de limbă [8].

Dicționarul de pronunție este, cum se poate deduce și din denumire, o mapare a cuvintelor din limba analizată la forma corespunzătoare dintr-un sistem de transcriere fonetică. ARPAbet este unul din cele mai frecvent folosite sisteme de transcriere utilizat pentru realizarea acestei sarcini.

Modelul acustic oferă probabilitatea după care într-un anumit interval de timp se întâlnește un anume fonem și este antrenat folosind un număr foarte mare de ore de înregistrări audio [8], motiv pentru care cele mai multe motoare de recunoaștere vocală includ mai multe modele acustice antrenate în prealabil, oferind totuși posibilitatea de a antrena un model acustic propriu sau de a încorpora ulterior noi caracteristici în modelul deja existent.

Categoria de sisteme de recunoaștere vocală utilizate în această lucrare, sisteme de recunoaștere bazate pe Modele Markov Ascunse, poartă acest nume ca urmare a faptului că modelul acustic este defapt un Model Markov Ascuns<sup>12</sup>.

Modelul de limbă este un model statistic care ajută decodorul să facă diferență între cuvinte asemănătoare în funcție de contextul în care este întâlnit cuvântul. De exemplu un model de limbă va prezice că într-o frază precum “Toni Morrison won the Nobel”<sup>13</sup> este mai probabil ca urmatorul cuvânt să fie “Prize” și nu “dries”, deși cele două cuvinte sunt foarte asemănătoare din punct de vedere acustic. Există mai multe tipuri de modele de limbă, N-gram, bidirectional, exponentional, dar cel mai folosit în recunoașterea vocală este N-gram. Pentru a prezice un cuvânt(sau o altă parte din vorbire, silabă, fonem) folosind N-gram sunt utilizate cele n-1 cuvinte care îl preced.

Decodorul este componenta principală a sistemului de recunoaștere vocală. Acesta primește la intrare informația legată de fragmentul audio pentru care se dorește să se obțină o transcriere și returnează vectori care conțin caracteristicile acestuia, informația primită la intrare este corelată cu informația deținută de knowledge base, și astfel se obține secvența de cuvinte (sau o altă parte din vorbire, silabă, fonem) cu cea mai mare probabilitate de a fi prezentă în fragmentul audio [8].

---

<sup>12</sup> Un Model Markov Ascuns este o metodă de reprezentare grafică a datelor, care ne permite să prezicem o serie de variabile “ascunse” utilizând alte variabile cunoscute

<sup>13</sup> <https://www.amazon.science/blog/how-to-make-neural-language-models-practical-for-speech-recognition>

Concret, decodorul calculează:

$$\hat{w} = \operatorname{argmax} P(w|Y),$$

unde  $w$  este secvența de cuvinte cu care mai mare probabilitate de a fi generat fragmentul audio, iar  $Y$  este rezultatul returnat de componenta care preprocesează semnalul audio.

Utilizând formula lui Bayes, ajungem la următoarea formă:

$$\hat{w} = \operatorname{argmax} P(Y|w)P(w)$$

În această formă trebuie calculată probabilitatea de apariție a unui anume cuvânt, lucru care poate fi calculat utilizând modelul de limbă, și probabilitatea ca știind cuvântul să obținem un vector de caracteristici ca cel dat la intrare, lucru care poate fi estimat cu ajutorul modelului acustic [5].

### 3.2.2 Evaluare pronunției

În urma proceselor descrise anterior, ca urmare a introducerii unui fragment audio drept dată de intrare, vom primi drept rezultat o secvență de foneme care compun cuvintele extrase din fragmentul audio. Această secvență de foneme, comparată cu transcrierea în același sistem fonetic a textului original, reprezintă o evaluare a pronunției utilizatorului, întrucât, urmărind ambele transcrieri se pot observa foneme lipsă, adăugate în plus sau înlocuite cu altele. Există totuși o serie de aspecte care îngreunează această sarcină.

În primul rând, chiar și în sistemele de recunoaștere vocală cu acuratețe ridicată în ceea ce privește sarcina de recunoaștere a cuvintelor și frazelor, extragerea transcrierii fonetice poate avea o performanță scăzută. Acest lucru se datorează faptului că, precum am văzut în descrierea prezentată anterior, recunoașterea folosește componente legate de contextul în care este regăsit cuvântul, mai exact, probabilitățile asignate de modelul de limbă care oferă sistemului informații cu privire la probabilitatea de apariției a unui cuvânt. Aceste probabilități, prezente și la nivel fonetic, nu oferă informații relevante sistemului deoarece constrângerile nu sunt aşa mari ca cele de la nivelul cuvântului [13].

În al doilea rând, o problemă care apare este dată și de faptul că un model acustic potrivit pentru sarcina de recunoaștere și transcriere vocală, este antrenat de cele mai multe ori utilizând fișiere audio înregistrate de vorbitorii nativi ai limbii în care se face recunoașterea, motiv pentru care posibilele greșeli care pot surveni în timpul învățării unei limbi noi pot conduce la un comportament imprevizibil al modelului acustic. Acest fenomen apare, deoarece probabilitățile în timpul antrenării nu

iau în calcul aceste erori. Pentru a putea rezolva aceasta problemă va trebui să antrenăm un model acustic sau să adaptăm modelul acustic existent pentru a include și greșelile de pronunție. De asemenea, aceste posibile erori trebuie să fie prezente și în dicționarul de pronunție, astfel încât în momentul în care pronunția nu corespunde exact pronunției native, sistemul să poată alege eroarea cea mai probabilă în loc să intre într-o stare necunoscută care să conducă la rezultate neașteptate.

Pentru că aplicația își propune să ofere o soluție cât mai personalizată pe nevoile fiecărui utilizator, modelul acustic va fi adaptat și în funcție de înregistrările trimise în timpul procesului de exersare a pronunției. Astfel, pentru fiecare utilizator vom avea un model acustic individual care să încorporeze caracteristicile vocale ale acestuia, dar și aspecte ce țin de mediul în care sunt realizate înregistrările.

În concluzie, pentru a putea realiza sarcina de evaluare a pronunției utilizând un sistem care realizează o transcriere fonetică pe baza unui recunoașterii vocale vom antrena un model acustic atât cu date care provin de la vorbitori nativi de limba engleză, de exemplu, cât și cu date care provin de la vorbitorii care învață limba engleză drept a doua limbă. Dacă ne dorim ca sistemul să fie orientat spre recunoașterea greșelilor de pronunție pe care le fac vorbitorii de limba română când învață engleză, vom include în antrenarea sistemului înregistrări cu vorbitori nativi de limba romană care vorbesc în engleză. Acest lucru este necesar deoarece ne dorim ca modelul să încorporeze greșelile specifice și particularitățile acestui tip de vorbitor. De asemenea, în transcrierile fișierelor folosite pentru antrenare va trebui să notăm aceste greșeli, pentru ca sistemul să poată să le detecteze și să nu le interpreteze ca pronunții corecte.

Metode de detectare a greșelilor de pronunție pe baza unui sistem capabil să realizeze o transcriere fonetică sunt descrise atât în [5], unde este descrisă o metodă care utilizează de asemenea sistemul CMU Sphinx, dar procesează rezultatul obținut de acesta și îl folosește drept date de intrare pentru un Model de mixtură Gaussiană<sup>14</sup> care este folosit pentru adaptarea modelului și în [4], unde este folosit sistemul de recunoaștere vocală Kaldi, care este antrenat inițial cu o bază de date a unor vorbitori de engleză nativi, dar și vorbitori de limba germană care învață engleză, dar care nu este adaptat ulterior în funcție de vocea utilizatorului.

---

<sup>14</sup> Model probabilistic care grupează datele după o distribuție normală

### **3.3 Evaluare pronunției utilizând Rețele Neuronale Convoluționale**

#### **3.3.1 Rețele Neuronale Convoluționale pentru clasificare audio**

Rețele neuronale au fost concepute pentru a găsi o metodă de rezolvare a problemelor cât mai asemănătoare cu modul în care acestea sunt modelate de creierul uman<sup>15</sup>. Pentru realizarea acestei sarcini rețelele neuronale sunt construite pentru a putea identifica tipare în datele primite și ulterior să poată răspundă la diverse întrebări în funcție de tiparele pe care le pot recunoaște.

Rețelele neuronale au avantajul de a putea modela probleme care nu pot fi modelate prin metodele clasice. Totodată învățarea este bazată pe exemple, drept urmare nu necesită algoritmi de complexitate ridicată. Aceste avantaje ale rețelelor neuronale au fost intens folosite în aria Învățării Profunde, în domenii precum: recunoașterea vocală, recunoașterea scrisului de mână, recunoașterea de fețe umane sau pentru furnizarea de sisteme capabile să joace jocuri (e.g. șah, Go).

Rețelele neuronale convoluționale sunt o clasă de rețele neuronale artificiale utilizate cel mai frecvent pentru clasificarea imaginilor. Aceste rețele utilizează straturi convoluționale care filtrează datele de intrare pentru a obține cele mai utile informații din acestea.

Eliminarea necesității de a extrage caracteristicile imaginii în prealabil și posibilitatea de a realiza acest lucru în timpul antrenării reprezintă unul din avantajele majore ale rețelelor neuronale convoluționale. De asemenea, rețele neuronale convoluționale sunt construite astfel încât să profite de bidimensionalitatea unei imagini și să surprindă mai bine relațiile spațiale între zonele și între pixelii imaginii [10]. Acest lucru nu poate fi reprezentat cu ajutorul rețelelor neuronale clasice deoarece informația din imagine trebuie aplatizată și astfel se pierd aceste informații legate de relațiile spațiale, dar și pentru că în cazul unei imagini avem de-a face cu un număr foarte mare de parametrii de intrare ( $L \times l \times c$ ), număr greu de procesat de o rețea neuronală clasică.

Clasificarea unei imagini cu ajutorul unei rețele neuronale convoluționale se face utilizând următoarea arhitectură.

---

<sup>15</sup> Mai multe despre rețele neuronale [https://www.sas.com/ro\\_ro/insights/analytics/neural-networks.html](https://www.sas.com/ro_ro/insights/analytics/neural-networks.html)

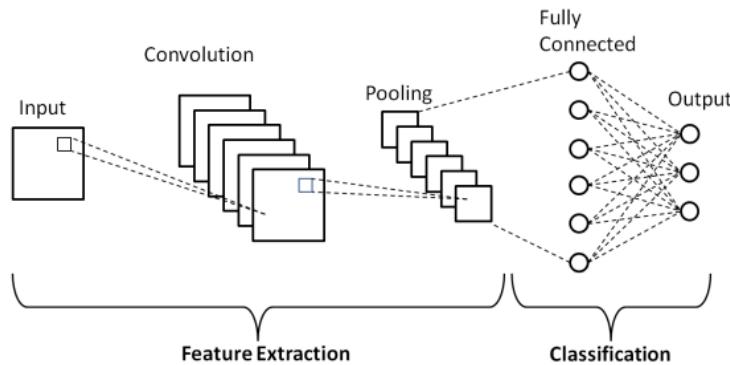


Fig. 3.3.1: Arhitectura generală a unei rețele neuronale conoluționale<sup>16</sup>

**Date de intrare:** Imaginile care reprezintă datele de intrare sunt reprezentate cu ajutorul unor tensori (obiect geometric care poate include date N dimensionale). În cazul imaginilor dimensiunea acestor tensori este:  $L \times l \times C$ , unde  $L$  reprezintă lățimea imaginii,  $l$  înălțimea și  $C$  numărul de canale de culoare. Cea mai întâlnită valoare pentru  $C$  este 3 și este dată de cele 3 canale de culoare RGB (Red, Green, Blue).

**Straturile Conoluționale:** Stratul conoluțional este compus din mai multe filtre, un filtru are lungimea și lățimea mai mici decât cele corespunzătoare datelor de intrare, dar aceeași adâncime (e.g. datele de intrare au 3 canale de culoare, atâtea vor fi folosite și pentru filtru). Valorile corespunzătoare filtrelor vor fi învățate cu ajutorul rețelei. Un astfel de filtru este mutat atât pe lungime, cât și pe lățimea imaginii primite la intrare, și se realizează produsul scalar între filtru și secvența de imagine la care se află la un moment dat. Prin acest proces rețea învățăfiltrele care sunt activate când identifică o zonă din imagine cu anumite caracteristici. Rezultatul acestui strat este reprezentat de toate matricile rezultate ca urmare a aplicării filtrelor [10].

**Straturile de agregare (pooling):** Sunt prezente, în general, între straturile conoluționale și au rolul de a reduce numărul parametrilor implicați în rețea. Una din cea mai folosită metodă de agregare poartă numele de max pooling și presupune selectarea celei mai mari valori dintr-o zonă cu lungime și lățimea oferite ca parametri[10]. O altă metodă de agregare presupune alegerea mediei aritmetice a valorilor dintr-un perimetru.

<sup>16</sup>[https://www.researchgate.net/figure/Schematic-diagram-of-a-basic-convolutional-neural-network-CNN-architecture-26\\_f1\\_336805909](https://www.researchgate.net/figure/Schematic-diagram-of-a-basic-convolutional-neural-network-CNN-architecture-26_f1_336805909)

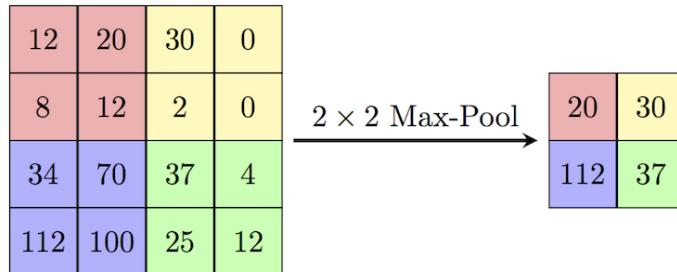


Fig. 3.3.2 Operația de max-pool

**Straturile conectate complet:** Primesc ca date de intrare toate valorile obținute în urma realizării operațiilor anterior descrise și se comportă ca o rețea neuronală clasică, mai exact calculează rezultatul funcției de cost utilizând înmulțiri matriciale [10].

**Date de ieșire:** Sunt reprezentate de un vector care conține probabilitatea apartenenței la fiecare din clasele pe care este antrenată rețeaua să le identifice.

Cum am menționat anterior, și cum se poate observa și din procesul de funcționare al rețelelor neuronale convoluționale, aceste rețele sunt specializate spre funcționalitatea de clasificare a imaginilor. Având în vedere acest aspect, pentru a putea utiliza o rețea neuronală convoluțională în contextul datelor audio, va fi necesară transformarea fragmentelor sonore în imagini, imagini care să surprindă, în continuare, caracteristicile sunetelor surprinse.

Cea mai frecvent folosită metodă de reprezentare a datelor audio într-un format vizual este cu ajutorul spectrogramelor. O spectrogramă este un grafic bidimensional, unde timpul este reprezentat pe axa orizontală, frecvența sunetului este reprezentată pe axa verticală, la care se adaugă și o a treia dimensiune reprezentată de culoare, culoare care variază în funcție de cât de puternică este o frecvență într-un anumit interval de timp[16].

Deoarece clasificarea semnalului sonor în bazele de date de antrenament se realizează prin intermediul percepției sonore a oamenilor, cele mai multe metode de clasificare a sunetului cu ajutorul rețelelor neuronale convoluționale primesc drept date de intrare Mel Spectograme. Scara Mel oferă o scală liniară pentru spectrul auditiv uman [11].

După ce datele audio sunt transformate în spectrograme Mel, procesul de clasificare utilizând rețele neuronale convoluționale, decurge la fel ca în cazul clasificării de imagini.

### 3.3.2 Învățarea prin transfer

Învățarea prin transfer este o metodă prin care un model antrenat pentru o anumită sarcină, utilizând un număr foarte mare de date de antrenament, este utilizat în cadrul unei noi cerințe.

În practică, o rețea neuronală conoluțională este foarte rar antrenată de la zero, deoarece acest proces necesită un număr foarte mare de date, în schimb este utilizată o rețea antrenată în prealabil pe o mulțime mare de date (un exemplu de astfel de mulțime de date este ImageNet), iar ultimul strat dintre cele conectate complet, va fi înlocuit cu un strat care corespunde claselor pentru care dorim să folosim noul model [10].

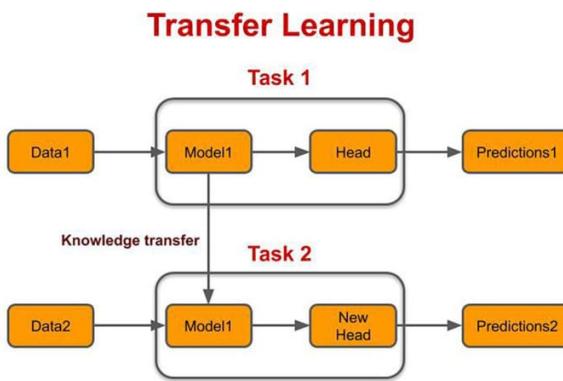


Fig. 3.3.3: Procesul de învățare prin transfer<sup>17</sup>

În ceea ce privește clasificarea de date de tipul audio sunt folosite frecvent două tipuri de modele. Modele antrenate special pentru clasificarea audio, utilizând drept bază de date de antrenament AudioSet, bază de date formată din secvențe de 10 secunde extrase din videoclipuri de pe Youtube, sau modele antrenate pentru clasificarea de imagini, antrenate pe baze de date precum ImageNet [12]. Cel mai cunoscut model pentru clasificarea audio poartă numele de VGGish, în ceea ce privește clasificarea imaginilor există mai multe modele cunoscute, dar cele mai utilizate sunt VGG16, ResNet50 și InceptionV3.

În cele ce urmează vom apela la procedeul de învățare prin transfer, deoarece baza de date de antrenament pe care o avem la dispoziție nu conține suficient de multe date pentru a putea realiza un model calitativ doar pe baza acesteia. Deoarece, modelele de clasificare a imaginilor au fost intens dezvoltate și cercetate în ultima perioadă și au ajuns la rezultate cu o precizie ridicată, dar și deoarece din căte se poate vedea și în [6] și [11] acestea pot fi adaptate cu succes în ceea ce privește sarcina de

<sup>17</sup> <https://www.topbots.com/transfer-learning-in-nlp/>

clasificare a semnalelor sonore, vom alege să folosim acest tip de modele antrenate în prealabil pentru realizarea clasificărilor pe care le vom face cu scopul evaluării pronunției.

### 3.3.3 Evaluarea pronunției

Pentru evaluare pronunției vom încerca două abordări:

Prima abordare, presupune să antrenăm câte un model de rețea neuronală conoluțională pentru fiecare din cuvintele pe care vrem să le folosim. Evaluare pronunției se va face în modul următor, pentru fiecare cuvânt prezent în aplicație, vom avea un model individual. Pentru fiecare înregistrare se va încărca modelul corespunzător cuvântului prezent în înregistrare, iar modelul va răspunde la următoarea întrebare: *“Este aceasta o pronunție corectă a cuvântului în cauză sau conține o greșeală specifică vorbitorilor de română pentru acel cuvânt”*. Greșelile asociate fiecărui cuvânt vor fi obținute cu ajutorul Speech Accent Archive. Pentru antrenare vom folosi înregistrări cu pronunții corecte ale cuvântului și pronunții greșite adnotate în prealabil. În [7] poate fi întâlnită o abordare similară pentru detecția greșelilor de pronunție, utilizată pentru a identifica greșelile tipice de pronunție realizate de vorbitorii de limbă finlandeză, în procesul de învățare a limbii engleze.

Cea de-a doua abordare, presupune să antrenăm un model de rețea neuronală conoluțională care face diferența dintre o pronunție greșită și una corectă indiferent de înregistrarea primită drept date de intrare. Pentru realizarea acestui lucru vom antrena un singur model capabil să răspundă la această întrebare. Deoarece nu putem identifica o regulă general valabilă legată de nivelul la care se întâlnesc problemele de pronunție, vom antrena modelul în două moduri diferite și vom vedea care din aceste două modele rezultate va obține rezultate mai bune.

În prima variantă vom antrena modelul folosind drept bază de date de antrenament înregistrările originale împărțite în secvențe de câte o secundă. Pentru evaluare pronunției cu ajutorul acestui model vom proceda în modul următor: înregistrarea trimisă de utilizator va fi împărțită în secvențe de câte o secundă, pentru fiecare secvență se va genera o spectrogramă Mel, pentru fiecare spectrogramă obținută vom prezice dacă aceasta conține o pronunție corectă sau una greșită, răspunsul final trimis utilizatorului. Răspunsul va fi reprezentat de rezultatul unui vot majoritar aplicat pe aceste predicții, care ne va spune dacă cuvântul a fost preponderent pronunțat corect sau greșit. Această variantă este inspirată din [16], unde o rețea neuronală conoluțională este antrenată tot cu secvențe de câte o secundă din înregistrarea completă, în vederea realizării sarcinii de clasificare a unor accente.

În ceea de-a doua variantă modelul vom antrena un model utilizând cuvintele extrase pentru prima abordare, doar că nu vom mai recurge la varianta în care antrenăm un model diferit pentru fiecare cuvânt întâlnit, ci le vom folosi pe toate împreună drept bază de date de antrenament. Evaluare pronunției se face în următorii pași: spectrograma Mel a înregistrării întregului cuvânt este trimisă în întregime către model, iar acesta ne va spune dacă pronunția este conform pronunției unui nativ sau dacă poate fi îmbunătățită.

Se pot enunța o serie de avantaje și dezavantaje, în ceea ce privește cele două tipuri de modele antrenate pentru cea de-a două abordare. Un avantaj al primului model prezentat este dat de faptul că se păstrează consistența datelor, deoarece este urmărită aceeași fereastră de timp de o secundă. De asemenea, un alt avantaj al datelor de intrare și de antrenare utilizate de primul model este dat de faptul că, împreună cu o unealtă capabilă să facă o aliniere forțată a textului cu înregistrarea, punem surprinde greșelile la nivel de fonem. Având predicția pentru fiecare secundă din cuvânt și corespondența dintre intervalele de timp și sunetele emise în acele intervale, putem pune în evidență ce sunete au fost pronunțate greșit și ce sunete au fost pronunțate corect. Un dezavantaj al primei metode, rezolvat prin cel de-al doilea model este dat de faptul că păstrând întreaga înregistrare vom avea un context mai general asupra pronunției și nu vom exclude greșeli de pronunție care apar în punctele de legătură dintre sunete. Deși, analiza asupra spectrogramei întregului cuvânt reprezintă o sarcină mai dificilă comparativ cu analiza unei secvențe de o secundă, în final având datele de antrenament necesare putem obține un model mai robust în ceea ce privește legăturile dintre sunete.

De asemenea, putem enumera avantajele și dezavantajele celor două abordări majore încercate, cea de extragere a greșelilor tipice cu un model individual pentru fiecare cuvânt, și cea care evaluatează pronunția la modul general independent de conținutul înregistrării.

Un dezavantaj al primei abordări este legat de faptul că este constrâns de datele de antrenament disponibile, întrucât avem nevoie de înregistrări atât corecte, cât și greșite, pentru fiecare cuvânt în parte. Din acest motiv cuvintele oferite drept exerciții se vor rezuma la cuvintele pentru care găsim un număr rezonabil de date de antrenament. În cazul celei de-a două abordări cuvintele pentru care se realizează predicția pot fi independente de cuvintele utilizate în procesul de antrenare.

Totuși în ceea ce privește răspunsul oferit utilizatorului prima abordare oferă mai multe date cu privire la ce ar putea fi îmbunătățit în pronunției, comparativ cu cea de-a două abordare care printr-un răspuns negativ nu oferă detalii suplimentare cu privire la greșeala apărută și poate doar să încurajeze

utilizatorul să asculte încă odată pronunția corectă și să încerce să trimită o nouă înregistrare spre evaluare.

Anterior realizării experimentelor putem afirma că prima abordare este mai utilă în ceea ce privește procesul de învățare, în sensul în care încearcă să pună în evidență greșeli exacte spre a fi corectate, pe când a doua abordare este mai eficientă în ceea ce privește procesul de colectare a datelor de antrenament, dar și în ceea ce privește scalabilitatea, deoarece putem extinde numărul de cuvinte oferite drept exerciții oricât de mult.

### 3.4 Concluzii

Având în vedere cele două soluții prezentate în acest capitol, putem prezenta avantajele și dezavantajele prezentate de cele două având în vedere mai mulți factori.

Un prim aspect urmărește să surprindă cât de eficient este răspunsul returnat de fiecare din cele două metode, în procesul de învățare. În ceea ce privește această componentă, putem afirma că prima variantă, cea care returnează o transcriere fonetică oferă un răspuns, mult mai ușor de interpretat de către orice tip de utilizator și surprinde mai multe tipuri de greșeli posibile, lipsa unui fonem, înlocuirea acestuia cu un altul sau adăugarea unui fonem inutil în pronunție. În timp ce a doua metodă, este capabilă, în una din variantele ei să returneze o greșală generală, fără a se focaliza spre particularitățile de vorbire ale utilizatorului, lucru care face ca procesul de învățare să fie mai puțin eficient și să necesite și îndrumarea unei persoane avizate în acest domeniu. Această componentă va fi testată utilizând un test de utilizare din care va reieși varianta de răspuns preferată de utilizatori.

Un al doilea aspect evidențiază cât de ușor poate fi adaptată soluția pentru a putea fi utilizată pentru învățarea unei alte limbi, de către utilizatori cu diverse particularități datorate limbii materne. În ceea ce privește prima soluție prezentată, pentru adaptarea modelului acustic de bază, astfel încât acesta să conțină și posibile greșeli de pronunție, vom avea nevoie de un număr relativ mic de înregistrări, chiar și în jur de 30 de minute pot aduce o îmbunătățire în ceea ce privește calitatea modelului acustic. Problema apare, în acest caz, în situația în care limba pentru care vrem să realizăm evaluarea pronunției nu are asociat un model acustic. Pentru a realiza acest lucru va fi necesară antrenare unui nou model acustic, sarcină care necesită peste 200 de ore de înregistrări, ale unor utilizatori diferiți, pentru a putea obține un model funcțional. În ceea ce privește cea de-a doua metodă, în orice variantă, putem spune că avem nevoie de cel puțin o oră de înregistrări, de asemenea care să provină de la utilizatori diferiți și

care să conțină și greșeli de pronunție adnotate, pentru a putea realiza un model funcțional cu o capacitate de predicție corectă de peste 50%.

Drept urmare, prima variantă este ușor de adaptat pentru a putea realiza evaluare pentru limbile pentru care există un model acustic antrenat în prealabil, dar foarte greu în cazul în care antrenarea modelului acustic trebuie realizată de către cel care realizează aplicația, pe când cea de-a doua variantă poate fi adaptat la fel de ușor atât pentru limbile utilizate frecvent, cât și pentru limbi mai puțin utilizate și pentru care există un număr mai mic de date de antrenament.

# 4. Tehnologii utilizate

## 4.1 Android<sup>18</sup>

Conform Wikipedia, Android este o platformă software și un sistem de operare pentru dispozitive și telefoane mobile, bazată pe nucleul Linux. Android permite dezvoltatorilor să scrie un cod gestionat în limbajul Java , controlând dispozitivul prin intermediul bibliotecilor Java dezvoltate de Google<sup>19</sup>.

La momentul actual peste 70% din dispozitivele mobile utilizează sistemul de operare Android, iar utilizatorii acestor dispozitive au de ales dintre peste 2.5 milioane de aplicații disponibile pe Google Play, aplicații din domenii cât se poate de diverse, începând cu educație, dezvoltare personală, până la divertisment.

Kotlin este un limbaj de programare open source, dezvoltat de JetBrains, total interoperabil cu Java, și care poate fi folosit drept urmare pentru dezvoltarea aplicațiilor utilizând platforma Android, având beneficiul de a fi mai sigur și mai lizibil decât predecesorul său.

## 4.2 Firebase<sup>20</sup>

Deținut de Google, Firebase oferă un pachet întreg de produse care permit dezvoltarea de aplicații web și mobile. Firebase a început ca un serviciu de baze de date în timp real, iar acum cuprinde nu mai puțin 18 servicii și multiple API-uri dedicate. Prin aceste servicii se numără următoarele:

- *Cloud Firestore*: bază de date NoSQL care permite modele de date complexe și metode avansate de interogare a datelor;
- *Cloud Storage*: spațiu de stocare pentru fișierele multimedia necesare în dezvoltarea aplicațiilor;

---

<sup>18</sup> Mai multe informații despre Android se pot găsi în documentația oficială disponibilă la adresa: <https://www.android.com/>

<sup>19</sup> [https://ro.wikipedia.org/wiki/Android\\_\(sistem\\_de\\_operare\)](https://ro.wikipedia.org/wiki/Android_(sistem_de_operare))

<sup>20</sup> Mai multe informații despre Firebase se pot găsi în documentația oficială disponibilă la adresa: <https://firebase.google.com/docs>

- *Firebase Authentication*: sistem de autentificare pentru utilizatori care cuprinde atât partea de interfață cu utilizatorul, cât și posibilitatea de autentificare prin multiple metode, cum ar fi, prin credențiale personalizate, email sau rețele social media;
- *Cloud Functions*: instrument care permite rularea de cod direct în cloud, fără un server, într-o manieră bazată pe evenimente.

Firebase, serviciul de Backend As-a-service al unei companii de renume, Google, are pe lângă multiplele funcționalități, avantajul de a fi foarte bine documentat și de a avea o comunitate vastă care contribuie constant la îmbunătățirea serviciilor disponibile.

### 4.3 CMU Sphinx<sup>21</sup>

CMU Sphinx, este un termen general care descrie un grup de sisteme de recunoaștere vocală dezvoltate de Universitatea Carnegie Mellon<sup>22</sup>. Pachetul CMU Sphinx cuprinde mai multe aplicații specializate pe diverse sarcini:

- *PocketSphinx*: interfață de nivel înalt pentru recunoaștere vocală;
- *Sphinxbase*: suport pentru librăriile utilizare de PocketSphinx;
- *Sphinxtrain*: unelte pentru antrenarea unui model acustic;
- *Sphinx4*: sistem de recunoaștere vocală, ajustabil, scris în Java.

CMU Sphinx pune la dispoziție utilizatorului posibilitatea de a antrena un model acustic<sup>23</sup> propriu sau de a adapta un model deja existent.

Adaptarea modelului deja existent este utilă în momentul în care ne dorim să încorporăm în analiză anumite particularități legate de vocea vorbitorului. Dacă aplicația are drept cerință recunoașterea vocală a unui dialect sau a unei limbi pentru care nu există un model acustic sau se dorește specializare asupra unei multimi restrânse de cuvinte, se folosește funcționalitatea de antrenare a unui model propriu<sup>24</sup>.

---

<sup>21</sup> Mai multe informații despre CMU Sphinx se pot găsi în documentația oficială disponibilă la adresa:

<https://cmusphinx.github.io/>

<sup>22</sup> [https://en.wikipedia.org/wiki/CMU\\_Sphinx](https://en.wikipedia.org/wiki/CMU_Sphinx)

<sup>23</sup> Model care definește caracteristicile fiecărui sunet întâlnit în limbajul analizat.

<sup>24</sup> <https://cmusphinx.github.io/wiki/tutorialadapt/>

## 4.4 Praat<sup>25</sup>

Praat este un instrument de analiză a vorbirii creat de către Paul Boersma și David Weenink în cadrul Institutului de Științe Fonetice al Universității din Amsterdam. Praat poate analiza, sintetiza și manipula semnalele audio<sup>26</sup>. În ceea ce privește analiza semnalului audio sunt puse la dispoziție următoarele: *analiza diapazonului* (întinderea sunetelor), *a intensității, a frecvenței sunetelor și realizarea de spectrograme și cochelegrame* (tip specific de spectrogramă, care reprezintă modul în care sunetul este percepț de urechea internă). De asemenea Praat oferă posibilitatea de a vizualiza fragmente de sunete adnotate cu ajutorul IPA (International Phonetics Alphabet), oferind posibilitatea de a observa ce sunet a fost interceptat într-un anume interval de timp din fragmentul audio analizat. Adnotările pot fi realizate direct utilizând Praat sau pot fi introduse și vizualizate împreună cu fragmentul audio corespunzător. Fișierele cu adnotări au formatul TextGrid și conțin sunetul și intervalul de timp corespunzător acestuia.

## 4.5 Montreal Forced Aligner

Alinierea forțată este procesul prin care având transcriptul unui fragment audio se determină în ce interval de timp apare un anumit cuvânt sau chiar sunet.<sup>27</sup>

Montreal Forced Aligner este un utilitar din linia de comandă care realizează automat sarcina de alinieră forțată a fragmentelor audio cu transcripția acestora. Pentru a realiza acest lucru Montreal Forced Aligner apelează la instrumentele de la Kaldi ASR (Automatic Speech Recognition).

Pentru a putea realiza alinierea forțată, utilitarul are nevoie de fișierele audio care urmează să fie aliniate, în formatul .wav, de transcriptul corespunzător fiecărui fișier audio, în formatul .lab, și de un dicționar de pronunție al cuvintelor întâlnite în transcript. Montreal Forced Aligner utilizează informațiile legate de specificul vocii utilizatorului pentru a adapta alinierea, motiv pentru care fișierele audio trebuie să fie grupate în fișiere în funcție de vorbitor.

---

<sup>25</sup> Mai multe informații despre Praat se pot găsi în documentația oficială disponibilă la adresa următoare:

<https://www.fon.hum.uva.nl/praat/>

<sup>26</sup> <https://github.com/praat/praat>

<sup>27</sup> Mai multe informații despre Montreal Forced Aligner se pot găsi în documentația oficială disponibilă la adresa:  
<https://montreal-forced-aligner.readthedocs.io/en/latest/>

În urma utilizării acestui utilitar se vor obține fișiere de tipul TextGrid, corespunzătoare fiecărui fișier audio existent în setul de date.

## 4.6 Tensorflow

Tensorflow este o platformă open source care pune la dispoziție o serie de algoritmi necesari în domeniul învățării automate. Librăria Tensorflow este creată de către Google și oferă versiuni stabile pentru limbajele de programare Python și C++.<sup>28</sup>

Arhitectura mecanismului de execuție în cadrul Tensorflow este sub formă de grafuri, lucru care facilitează distribuția într-o rețea de calculatoare utilizând GPU.

### 4.6.1 Keras

Keras<sup>29</sup> este un API de nivel înalt al Tensorflow, scris în Python. Scopul principal al acestui API este acela de diminua efortul depus pentru realizarea de experimente în domeniul învățării automate și de a face o tranziție cât mai agreabilă de la idee la implementare.

Keras Applications sunt o serie de modele antrenate în prealabil. Aceste modele pot fi utilizate pentru predicții, extragere de caracteristici sau în cadrul procesului de învățare prin transfer, proces prin care ponderile obținute în urma antrenării unui model pentru o anumită problemă, sunt reutilizate pentru o problematică asemănătoare. Sunt disponibile următoarele modele pentru clasificarea de imagini (antrenare utilizând ca multime de date ImageNet<sup>30</sup>).

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88

Fig. 3.1: Modele disponibile și precizia acestora măsurată utilizând datele de validare ImageNet

<sup>28</sup> Mai multe informații despre Tensorflow pot fi găsite în documentația oficială disponibilă la adresa: <https://www.tensorflow.org/>

<sup>29</sup> <https://keras.io/api/applications/>

<sup>30</sup> Bază de date cu imagini organizată conform ierarhiei WordNet, în care fiecare cuvânt are asociate sute de mii de imagini

## 4.7 Concluzii

În această secțiune au fost prezentate tehnologiile folosite pentru realizarea interfeței, a serviciilor utilizate de aceasta, cât și folosite în procesul de analiză al semnalului sonor.

Platforma Android a fost aleasă deoarece este o platformă de dezvoltare mobile stabilă, accesibilă și populară, fiind drept urmare bine documentată. Având în vedere că scopul principal al aplicației este de a facilita procesul de învățare a unei limbi străine prin intermediul unei aplicații mobile, platforma Android se mulează pe această cerință esențială a aplicației dezvoltate.

Firebase vine în completarea platformei Android cu serviciul de Mobile Backend as a Service, oferind funcționalități precum: accesul la o bază și la un spațiu de stocare pentru informațiile utilizatorului, autentificarea utilizatorilor și posibilitatea de trimiterea de notificări, aspecte importante în ceea ce privește experiența celor care utilizează aplicația și respectiv, gradul lor de implicare în procesul de învățare prin intermediul acestieia.

CMU Sphinx este utilizat în experimentele legate de detecția greșelilor de pronunție utilizând recunoașterea fonetică, ca sistem de recunoaștere vocală automată, CMU Sphinx are avantajul de a putea fi personalizat în funcție de cerințele proiectului.

Praat și Montreal Forced Aligner vor fi folosite pentru a oferi o analiză utilizatorului, dar și pentru a realiza o comparație între pronunția acestuia și cea unui vorbitor nativ, de asemenea ambele utilitar contribuie la vizualizarea datelor implicate în procesul de antrenare.

Librăria Tensorflow va fi folosită pentru antrenarea unui model capabil să diferențieze între pronunția unui vorbitor nativ și o pronunție greșită, având toți algoritmii necesari pentru lucrul cu rețele neuronale convoluționale.

Având în vedere faptul că Praat, Tensorflow și CMU Sphinx, prin PocketSphinx, sunt disponibile în Python, acesta va fi limbajul utilizat în procesul de detecție a erorilor de pronunție, iar comunicare cu interfața din Android se va realiza cu ajutorul unui server Flask.

# **5. Dezvoltarea aplicației**

## **5.1 Introducere**

În capitolele ce urmează vor fi prezentate atât aspecte ce țin de arhitectura și de funcționalitățile generale ale aplicației dezvoltată, aplicație ce are drept scop încurajarea și sprijinirea procesului de învățare a unui limbă străine, cât și experimentele realizate în ceea ce privește evaluarea pronunției, funcționalitate inclusă în aplicație.

Astfel, primul capitol va urmări să prezinte modulele principale disponibile și să descrie modul în care acestea au fost implementate, iar următoarele două capitole vor fi reprezentate de experimentele realizate pornind de la aspectele teoretice prezentate în capitolul 3, cum pot fi acestea reproduse și o serie de concluzii cu privire la performanța metodelor abordate.

## **5.2 Arhitectura aplicației**

### **5.2.1 Introducere**

Având în vedere că aplicația dezvoltată este o aplicație de învățare a limbilor străine asistată de mobil, modulele principale ale aplicației sunt dezvoltate astfel încât să surprindă componentele necesare în acest proces, mai exact, un modul de evaluare a pronunției, un modul de evaluare a gramaticii și un modul de asignare a temelor.

Aplicația este dezvoltată utilizând platforma Android, pentru realizarea interfeței și a funcționalităților, Firebase, pentru sarcinile care necesită stocarea de date, dar și framework-ul Flask, necesar pentru a expune pe baza unui API serviciile de evaluare a pronunției.

Aplicația Android și funcționalitățile implementate utilizând această platformă sunt organizate conform unei arhitecturi monolitice. Arhitectura monolică presupune că atât interfața de comunicare cu utilizatorul, cât și parte de procesare a comenziilor introduse de acesta și respectiv comunicarea cu baza de date se realizează în cadrul aceluiași cod sursă. Astfel, deși fiecare componentă e decuplată la nivel logic de celelalte, dacă se dorește modificarea uneia dintre ele este necesară actualizarea întregii

aplicații. Acest monolit comunică prin intermediul unui API cu serviciul care realizează evaluare pronunției, dar și cu serviciul care realizează înregistrări cu vocea clonată a utilizatorului.

În ceea ce privește stocarea datelor vom folosi, cum am menționat anterior Firebase, mai exact serviciul Firestore. Firestore este o bază de date nerelațională, în care datele sunt reprezentate sub formă de documente în formatul JSON, iar documentele sunt grupate conform unor colecții. Documentele pot conține atât informații simple, precum siruri de caractere sau date numerice, dar și date complexe, cum ar fi liste și dicționare. Deși la nivel structural bazele de date nerelaționale nu impun existența constrângerilor sau o formă fixă a datelor, acestea vor fi necesare pentru realizarea legăturilor logice din cadrul aplicației. Informațiile despre colecțiile utilizate și relațiile dintre acestea sunt reprezentate în următoarea schemă, în care fiecare entitate reprezintă o colecție, iar atributele definesc forma documentelor din acestea.

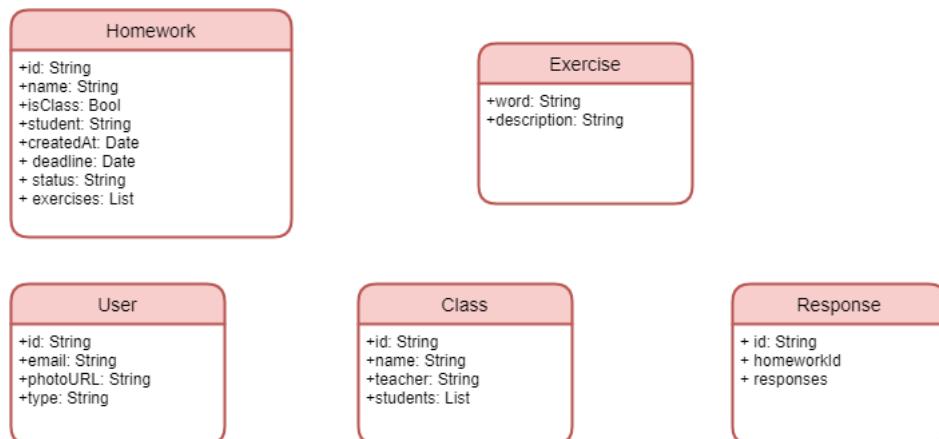


Fig. 5.2.1: Colecțiile utilizate în aplicație

## 5.2.2 Modulul de autentificare și tipurile de utilizatori

Înregistrarea utilizatorilor aplicației se realizează utilizând serviciul Firebase Authentication, care pune la dispoziție atât interfață de comunicare cu utilizatorul în vedere realizării procesului de autentificare, cât și mulți furnizori ai funcționalității de înregistrare. În cadrul aplicației prezentată în această lucrare utilizatorii se vor putea autentifica utilizând fie o înregistrare pe bază de e-mail și parolă, fie înregistrare pe baza unui cont Google.

Pentru gestionarea mai eficientă a atributelor și relațiilor în care sunt implicați utilizatori, pentru fiecare utilizator autenticat există un document corespunzător în colecția denumită *User*. Atributele id, email și type sunt setate în momentul în care utilizatorul se înregistrează pentru prima dată în aplicație și nu pot fi modificate ulterior. Atributul photoUrl este un atribut de tipul sir de caractere care

identifică locația din Firebase Storage în care poate fi găsită poza de profil asociată utilizatorului curent.

Funcționalitățile aplicației sunt disponibile exclusiv utilizatorilor înregistrați, iar în funcție de interesul acestora se pot delimita mai multe grupuri de public țintă, grupuri vizibile și la nivelul aplicației.

În primul rând, avem în vedere că aplicația se adresează atât copiilor, considerați utilizatori începători, care se află în faza incipientă în ceea ce privește procesul de învățare a unei limbi, cât și adulților, priviți drept utilizatori experimentați, care doresc să-și perfecționeze anumite abilități în ceea ce privește limba engleză. Această separare între tipurile de utilizatori se realizează la nivelul aplicației prin prezența a două zone care conțin tipuri de activități diferite destinate fiecărui tip de utilizator.

În ceea ce privește zona destinată începătorilor avem următoarele tipuri de exerciții: exerciții de pronunție care conțin cuvinte din viața de zi cu zi, noțiuni legate de timp, numere și animale și exerciții de dezvoltare a vocabularului focusate pe obiecte de îmbrăcăminte, fructe, legume, obiecte întâlnite în diverse încăperi din casă.

Zona destinată utilizatorilor experimentați cuprinde următoarele activități: vizualizarea și audierea fiecărui sunet asociat unei simbol din ARPAbet, exerciții de pronunție care conțin cuvinte considerate avansate, exerciții de dezvoltare a vocabularului pe tematici precum conversația la un restaurant, conținutul unui CV sau discuții din cadrul unui interviu de angajare, o zonă de informații în care utilizatori pot citi mai multe despre problemele de pronunție și o hartă pe care pot vizualiza instituțiile din apropiere în care pot consulta un logoped.

În al doilea rând, există și o separare în ceea ce privește rolul utilizatorului, separare necesară pentru dezvoltarea modului de asignare a temelor. Din acest punct de vedere avem utilizatori de tipul elev (student) și profesor. Această delimitare este marcată la nivelul bazei de date prin atributul *type*, prezent în documentele din colecția *User* (Fig. 5.2.1). Atributul *type* poate fi setat doar în momentul în care utilizatorul se înregistrează în aplicație. Mai multe detalii cu privire la atribuțiile fiecărui tip de utilizator vor fi prezentate în capitolul 5.2.5, destinat modului de asignare a temelor.

### 5.2.3 Modulul de evaluare a pronunției

Modulul de evaluare și de exersare al pronunției este modul în care sunt puse în practică aspectele teoretice prezentate pe parcursul lucrării. Acest modul conține mai multe funcționalități, funcționalități care îmbină duc la realizarea sarcinii finale, perfecționarea pronunției în limba engleză.

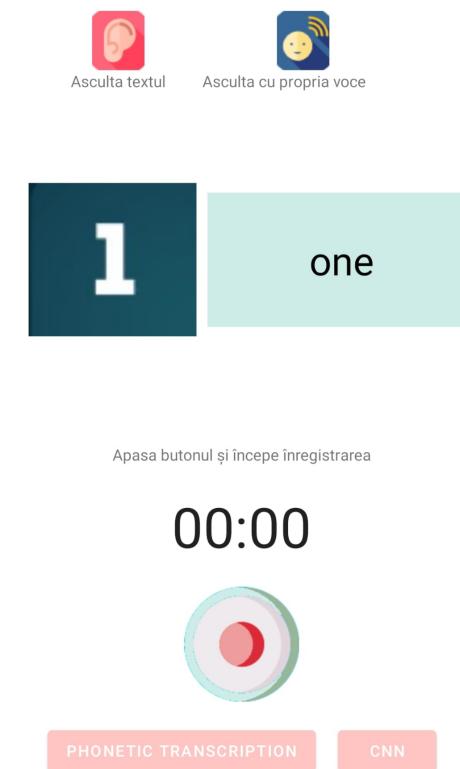


Fig. 5.2.2: Interfața funcționalității de evaluare a pronunției

Procesul de evaluare al pronunției funcționează în modul următor. În funcție de zona în care se află utilizatorul, zona dedicată începătorilor sau zona dedicată utilizatorilor avansați, acesta poate alege din mai multe categorii de exerciții de pronunție. Pentru fiecare categorie există o listă cu mai multe cuvinte din aria categorii alese, aceste cuvinte vor reprezenta exercițiile de pronunție. Listele de exerciții pentru toate categoriile sunt salvate, de asemenea în Firestore și sunt reprezentate de colecții individuale pentru fiecare categorie, colecții care conțin documente cu formatul entității *Exercise* descris în Fig. 5.2.1. Pentru exercițiile dedicate utilizatorilor începători câmpul *description* conține adresa din Firebase Storage la care poate fi găsită imaginea asociată cuvântului, imagine încărcată în momentul în care se vizualizează exercițiul. Pentru exercițiile dedicate utilizatorilor avansați, câmpul *description* conține definiția cuvântului oferit drept exercițiu. Odată ales un exercițiu se deschide activitatea din Fig. 5.2.2. În această activitate are loc evaluarea abilităților de pronunție.

În partea superioară, în Fig. 5.2.2, se observă două butoane, un buton cu textul “*Asculta textul*” și un buton care are asociat textul “*Ascultă cu propria voce*”. Scopul acestor butoane este de a aduce la cunoștință utilizatorului pronunția corectă a cuvântului ales drept exercițiu.

Apăsând butonul “*Asculta textul*” utilizatorul aude pronunția cuvântului cu ajutorul clasei TextToSpeech disponibilă în Android. Această clasă este capabilă să convertească un fragment de text în versiunea audio a acestuia. Clasa este capabilă să realizeze această sarcină în multiple limbi, dar în cazul aplicației curentă vom stabili drept limbă în care se face translatarea, limba engleză. Pentru a utiliza TextToSpeech este suficientă realizarea unei instanțe a clasei și apelarea metodei *speak* prezentă în această, având ca argument textul pe care dorim să-l auzim.

Butonul “*Ascultă cu propria voce*” se află în corelație cu abilitatea aplicației de a clona vocea utilizatorului și de a avea astfel posibilitatea să realizeze înregistrări cu vocea acestuia citind textul exercițiilor propuse. Funcționalitatea de clonare a vocii utilizatorului se realizează utilizând codul sursă pentru Real-Time Voice Clone [17] un framework de învățare automată cu o interfață grafică, capabil să realizeze o clonă a vocii unei persoane utilizând doar 5 secunde de înregistrare conținând vocea acestuia. Real-Time Voice Clone este o implementare a lucrării “*Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis (SV2TTS)*”. Obținerea înregistrării care va fi folosită pentru clonarea vocii se realizează în momentul în care utilizatorul se înregistrează în aplicație. Pentru a realiza acest lucru cel care folosește aplicația realizează o înregistrare în care citește un text în limba engleză afișat pe ecran. Utilizatorul poate alege să sară peste acest pas și drept urmare în momentul în care va încerca să acceseze această funcționalitate va fi anunțat că nu este disponibilă. Înregistrarea realizată este încărcată în Firebase Storage și în același timp locația acestei înregistrări este trimisă în corpul unei apel http la serverul din Flask responsabil de această funcționalitate. Este necesară comunicarea cu un server, deoarece Real-Time Voice Clone este disponibil doar în limbajul de programare Python.

La nivel de server ca răspuns la apelul făcut de aplicație, se descarcă fișierul audio și se trimită la Real-Time Voice Clone drept monștră privind vocea utilizatorului. Întrucât realizarea unei înregistrări cu vocea clonată poate dura până la aproximativ 2 minute și deoarece baza de date care conține cuvintele oferite drept exerciții nu se modifică în timpul utilizării aplicației, vom alege să realizăm toate înregistrările cu vocea clonată în momentul în care utilizatorul se înregistrează. Înregistrările obținute vor fi stocate pe server. În momentul în care persoana care folosește aplicația apasă butonul “*Ascultă cu propria voce*” se realizează un apel http, cu metoda get, care conține numele utilizatorului și cuvântul pentru care se dorește o înregistrare, iar înregistrarea cu vocea clonată este returnată și poate fi audiată de utilizator. Dacă se dorește modificarea înregistrărilor pentru utilizator se

poate retrimită o nouă probă de voce din pagina de profil și se vor urma aceeași pași ca în cazul primei înregistrări trimise.

În continuare în Fig. 5.2.2 observăm zona în care este afișat textul exercițiului de pronunție selectat și butonul care oprește și pornește înregistrarea.

După ce a realizat o înregistrare utilizatorul are două variante de evaluare, variante asociate celor două metode de evaluare a pronunției descrise în cadrul lucrării.

Modul de funcționare a celor două servicii de evaluare este prezentat în capitolele 5.3, pentru evaluarea utilizând recunoașterea fonetică, și respectiv 5.4, pentru evaluare utilizând rețele neuronale convoluționale.

În Fig. 5.3.1 sunt ilustrate evenimentele care au loc în momentul în care utilizatorul selectează evaluarea utilizând recunoașterea fonetică, iar în Fig. 5.4.1 sunt ilustrate evenimentele corespunzătoare celei de-a doua metode. Din căte se poate observa cele două servicii au un mod de funcționare asemănător, diferența dintre cele două apare în ultimul pas, pas în care se realizează efectiv evaluarea pronunției, dar și în ceea ce privește rezultatul returnat utilizatorului. Astfel, apăsând unul din butoane se declanșează următoarea serie de evenimente. În primul rând înregistrarea este încărcată în Firebase Storage și se apelează API-ul serverului care realizează evaluare, cu metoda post, apelul la conține informații cu privire la locația înregistrării în Firebase Storage. Înregistrarea este descărcată de către server și convertită în formatul .wav, necesar în etapele ce urmează. Indiferent de metoda aleasă înaintea evaluării se apelează serviciul de recunoaștere vocală care confirmă dacă cuvântul din înregistrare este cuvântul care corespunde exercițiului, dacă cuvântul nu este recunoscut, utilizatorului își cere să realizeze o nouă înregistrare, întrucât cea curentă nu este destul de inteligibilă. În continuare în funcție de metoda de evaluare aleasă, fie se va apela serviciul de transcriere fonetică pe baza înregistrării, fie vor fi realizate preprocesarea necesare, transformarea în spectrogramă, pentru a putea realiza o predicție cu privire la pronunție de către modelul de rețea neuronală convoluțională ales.

În final, dacă evaluarea se face utilizând transcrierea fonetică, utilizatorul primește drept răspuns secvența de foneme asociate înregistrării și secvența de foneme corectă pentru cuvântul exersat. Dacă evaluarea se face utilizând rețele neuronale convoluționale, răspunsul este reprezentat doar de eticheta de corect sau greșit cu privire la pronunție.

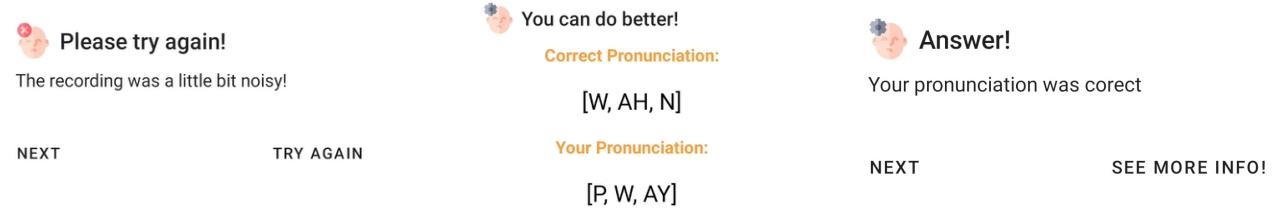


Fig. 5.2.3: Răspunsuri returnate de aplicație: răspunsul în cazul în care înregistrarea nu trece faza de recunoaștere vocală, răspunsul pentru transcrierea fonetică, răspunsul pentru evaluarea cu rețele neuronale convoluționale

Suplimentare, pe lângă evaluare propriu-zisă, serviciile de evaluare ale pronunției generează și 3 grafice în care sunt prezentate caracteristici ale înregistrării trimise spre evaluare. Graficele returnate sunt următoarele: un grafic în care este prezentat comparativ diferența dintre înălțimea sunetelor în înregistrarea trimisă de utilizator și înregistrarea unui vorbitor nativ pentru același cuvânt, spectrograma înregistrării utilizatorului și spectrograma înregistrării vorbitorului nativ. Aceste grafice pot fi analizate de o persoană avizată în domeniul lingvistic și în final aceasta poate observa tipare și fie să le asocieze cu probleme de pronunție, fie să ofere sfaturi legate de ce trebuie remediat, pe baza acestora. Graficele sunt realizate utilizând Parselmouth, librărie din Python, care profită de librăriile disponibile de realizare a graficelor, cum ar fi Matplotlib și creează reprezentări vizuale ale rezultatelor obținute în urma analizei asupra datelor audio realizată de algoritmii disponibili în Praat.

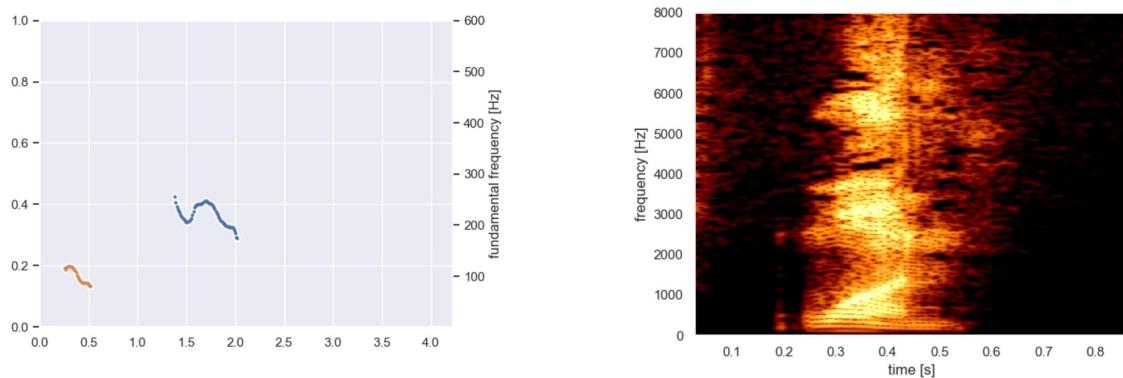


Fig. 5.2.4: Exemplu de comparație de înălțime a sunetelor(roșu-nativ, albastru-utilizator), exemplu de spectrogramă

#### **5.2.4 Modulul de dezvoltare a vocabularului**

O altă componentă importantă în ceea ce privește învățarea unei limbi străine este legată de complexitatea vocabularului pe care acesta îl are în limba respectivă. Aplicația conține o componentă dedicată dezvoltării acestei abilități, componentă a cărui mod de funcționare va fi descris în continuare.

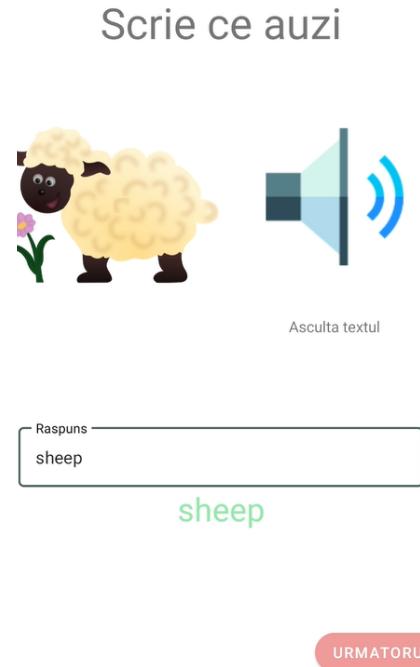


Fig. 5.2.5: Interfață modul de dezvoltare a vocabularului

Formatul documentelor utilizate pentru exercițiile din acest modul este același cu formatul exercițiilor de pronunție descrise în subcapitolul anterior.

Rezolvarea unui exercițiu, presupune ascultarea textului acestuia, lucru realizat ca în cazul modulului anterior cu ajutorul clasei TextToSpeech și introducere textului în caseta de text. Textul introdus este comparat cu textul original al exercițiului, iar drept răspuns utilizatorul vede dacă textul este corect sau în caz contrar cum ar fi trebuit să fie scris cuvântul audiat.

#### **5.2.5 Modulul de asignare și de rezolvare a temelor**

Modului de asignare și de rezolvare a temelor are scopul de a introduce aplicația prezentată în această lucrare în procesul clasic de predare, proces bazat pe două entități, student și profesor.

Pentru realizarea acestui modul vor fi necesare toate entitățile prezentate în Fig. 5.2.1, iar interacțiunile dintre acestea sunt cele care stau la baza funcționalităților din acest capitol.

În primul rând, cele două entități implicate în procesul de învățare, student și profesor, vor fi stabilitate cu ajutorul atributului *type* din documentele de tipul *User*, valoarea atributului este stabilită de utilizator în momentul înregistrării și nu poate fi modificată ulterior. Celor două roluri, cel de student și cel de profesor le vor corespunde atribuțiile din viața reală asociate cu acestea. Un utilizator de tipul profesor poate să gestioneze o clasă de elevi, să asigneze teme unei clase de studenți sau unui student individual și să primească rezolvarea acestora, iar un utilizator de tipul student poate să facă parte din mai multe clase, clase în care va primi teme spre a le rezolva.

Utilizatorul de tipul profesor poate realiza toate operațiile CRUD (Create, Read, Update, Delete), cu privire la clasele pe care le gestionează. Crearea unei clase, vizualizarea claselor asociate unui profesor și ștergerea unei clase se realizează dintr-un meniu dedicat prezent în profilul utilizatorului, iar acțiunile vor fi realizate cu ajutorul operațiilor corespunzătoare asupra colecției *Class*. Operația de modificare se referă la adăugarea studenților într-o clasă. Adăugarea unui student într-o clasă se poate realiza în două moduri: profesorul introduce adresa de email a acestuia sau studentul poate citi un cod QR asociat clasei și astfel va fi adăugat în aceasta. La nivelul bazei de date acest lucru presupune inserarea identificatorului studentului în câmpul *students*, câmp de tipul listă, din documentul asociat clasei.

În momentul în care un student este adăugat într-o clasă acesta este înștiințat prin notificarea. Notificare este trimisă cu ajutorul Firebase Function, serviciu care ne permite să realizăm acțiuni ca urmare a unor acțiuni asupra documentelor dintr-o colecție, în cazul de față acțiunea așteptată este acțiunea de *update* asupra colecției *Class*.

Utilizatorul de tipul student poate doar să vizualizeze clasele în care se află.

În ceea ce privește asignarea temelor, utilizatorului de tipul profesor îi este afișat următorul meniu în momentul în care apasă pe unul din exercițiile sau pe o listă de exerciții din aplicație.

### Assign Homework

Creează o temă pentru următoarea  
clasă

homework android

Class VI

Class V

SAU

Introdu adresa de email a studentului

Deadline:

DATA

ORA

CANCEL ASSIGN

Fig. 5.2.6: Meniul destinat asignării unei teme

La fel ca în cazul claselor, un profesor poate realiza operații de creare, citire, modificare și ștergere și asupra colecției *Homework*. În ceea ce privește crearea aceasta se poate realiza atât pentru doar un student, cât și pentru o clasă, întrucât documentele de tip *homework* sunt realizate prin corespondență dintre un student și o temă, drept urmare vom avea câte un document individual pentru fiecare student, chiar dacă tema este asignată unei clase. Citirea presupune vizualizarea temelor asignate, atât la nivel de clase, cât și la nivelul studenților individual. Modificarea unei teme se poate realiza în mai multe sensuri. În primul rând data limită a temei poate fi modificată de către profesor. De asemenea la o temă se pot adăuga ulterior noi exerciții, acest lucru se realizează prin inserarea exercițiilor în câmpul *exercises*. La fel ca în cazul anterior studentul care primește o temă este notificat cu ajutorul Firebase Functions, care vor urmări metoda de crearea pentru colecția *Homework*.

Utilizatorul de tip student, poate vizualiza temele curente și respectiv exercițiile conținute de acestea. Modul de afișare a exercițiilor este cel descris în subcapitolele 5.2.3 și 5.2.4. Fiecare răspuns este adăugat în câmpul *responses* asociat documentului din colecția *Response* care corespunde temei pe care o rezolvă. Câmpul *responses* este o listă de dicționare de două tipuri, răspunsurile asociate exercițiilor de dezvoltare a vocabularului și răspunsurile asociate exercițiilor de pronunție. Răspunsurile la exercițiile de dezvoltare a vocabularului conțin cuvântul corect și cuvântul introdus de

utilizator, iar răspunsurile asociate exercițiilor de pronunție conțin: răspunsul returnat de metoda prin care se realizează evaluarea, locația fișierului audio trimis spre evaluare și locația graficelor suplimentare generate pe baza acestuia.

La final după rezolvarea exercițiilor studentul poate trimite tema profesorului. Trimiterea temei se realizează prin modificarea statutului acesteia din “în proges”, fie în “Predată”, fie în predată cu întârzierea în cazul în care aceasta depășește termenul limită. Ulterior profesorul care a asignat tema are acces la răspunsurile trimise de student și poate analiza evoluția acestuia.

## 5.3 Evaluarea pronunției utilizând recunoașterea fonetică

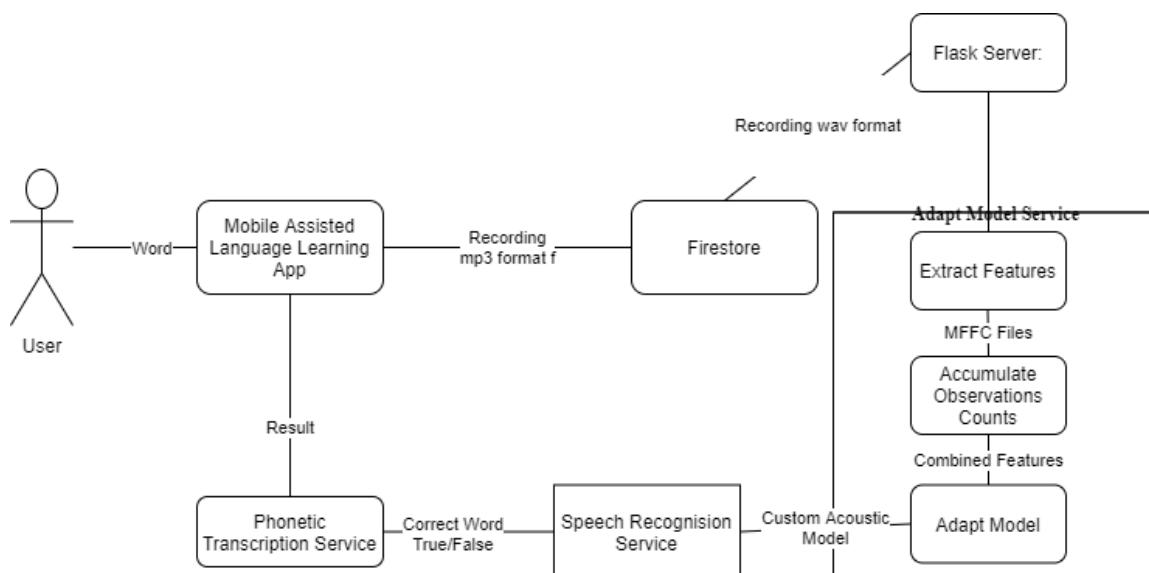


Fig 5.3.1: Evaluarea pronunției utilizând recunoașterea fonetică

### 5.3.1 Configurări

Pentru realizarea evaluării pronunției cu ajutorul recunoașterii fonetice vom folosi sistemul CMU Sphinx, mai exact Sphinx4, ultima variantă de motor de recunoaștere vocală propus de CMU.

Motorul de recunoaștere vocală Sphinx4 poate fi descărcat de pe pagina oficială a sistemului<sup>31</sup>, alături de instrucțiunile și dependințele necesare pentru instalare, și vine împreună cu un model acustic deja antrenat pentru limba engleză. Cum am menționat în capitolul 3.2.2, modelul acustic deja antrenat, va fi ulterior adaptat pentru a include posibilele greșeli de pronunție, dar și caracteristicile vocale ale

<sup>31</sup> <https://cmusphinx.github.io/wiki/download/>

utilizatorului. Pentru a putea utiliza și testa modelul vom folosi PocketSphinx, librărie care face parte din pachetul CMU Sphinx și care este disponibilă din Python.

Pentru a realiza sarcina de adaptare a modelului vom avea nevoie de o bază de date care să conțină următoarele: fișierele audio în formatul .wav (16 kHz sample rate, 16 bit rate), un fișier de tipul .fileids care să conțină numele fișierelor audio și un fișier de tipul .transcription care conține numele fiecărui fișier și transcrierea conținutului acestuia [13].

Pentru a obține fișierele audio vom apela la The Speech Accent Archive<sup>32</sup>, o arhivă care conține înregistrări ale aceleiași fraze în limba engleză, aleasă astfel încât să conțină cuvinte cât mai semnificative în ceea ce privește erorile de pronunție, citită de persoane din diferite țări și regiuni. Din această mulțime de date vom folosi 25 de înregistrări ale unor vorbitori de limba engleză, originari din America de Nord, 25 de înregistrări ale unor vorbitori de limba engleză, originari din Regatul Unit al Marii Britanii și 25 de înregistrări ale unor nativi români. Am ales numărul de 25 deoarece acesta este numărul de înregistrări ale vorbitorilor de limbă română și ne dorim să existe un echilibru în ceea ce privește datele de antrenament.

De asemnea, deoarece ne dorim ca sistemul să îndeplinească cu o acuratețe cât mai mare sarcina de recunoaștere a cuvintelor pe care le vom folosi pentru antrenarea pronunției, vom modifica și dicționarul de pronunție, dar și modelul de limbă, astfel încât acesta să conțină doar cuvintele utilizate pentru exercițiile de pronunție. Pentru a realiza acest lucru va trebui să creăm două noi tipuri de fișiere, un fișier de tipul .dic, care va conține dicționarul de pronunție, și un fișier de tipul .lm, care va conține modelul de limbă.

EIGHT	EY T
FIVE	F AY V
FOUR	F AO R
NINE	N AY N
ONE	W AH N
SEVEN	S EH V AH N
SIX	S IH K S
TEN	T EH N
THREE	TH R IY
TWO	T UW

Fig. 5.3.3: Exemplu de dicționar de pronunție

```
\data\
ngram 1=15
ngram 2=26
ngram 3=13

\1-grams:
-0.7782 </s> -0.3010
-0.7782 <s> -0.2218
-1.8921 BUNNY -0.2218
-1.8921 CAT -0.2218
-1.8921 COW -0.2218
-1.8921 DOG -0.2218
-1.8921 ELEPHANT -0.2218
-1.8921 FLAMINGO -0.2218
-1.8921 GIRAFFE -0.2218
-1.8921 HORSE -0.2218
-1.8921 KANGOROO -0.2218
```

Fig. 5.3.4: Exemplu de model de limbă

<sup>32</sup> <http://accent.gmu.edu/>

### 5.3.2 Experimente

Ulterior, după pregătirea bazei de date de antrenament, vom putea începe procesul de adaptare al modelului original pentru limba engleză oferit de CMU Sphinx<sup>33</sup>, pentru a include și greșeli de pronunție în limba engleză caracteristice vorbitorilor de limba română. Vom realiza acest lucru utilizând executabile puse la dispoziție de CMU Sphinx apelate din Python cu ajutorul librăriei subprocess. În primul rând, va trebui să transformăm fișierele audio în fișiere de tipul .mdef, fișiere care conțin MFCC (Mel frequency cepstral coefficients). În al doilea rând, caracteristicile fișierelor audio extrase în pasul anterior, vor fi combinate, ca în final acestea să fie incluse în modelul pe care îl adaptăm. După realizarea acestor pași vom obține un nou model acustic.

În momentul în care un utilizator încearcă pentru prima dată funcționalitatea de evaluare a pronunției utilizând transcrierea fonetică, decodorul care realizează această funcționalitate va apela la modelul acustic creat în timpul antrenării.

De asemenea, din Fig. 5.3.5 se observă că anterior transcrierii fonetice este apelat un serviciu de recunoaștere vocală, acest lucru se realizează utilizând același decodor. În contextul aplicației, acest pas ne ajută să identificăm dacă înregistrarea nu este destul de clară sau dacă nu conține cuvântul de antrenament, situații în care nu ne dorim să trecem la următoarea etapă.

Dacă înregistrarea este destul de inteligibilă pentru a putea fi recunoscută de sistemul de recunoaștere vocală, va fi returnată transcrierea fonetică a cuvântului sau frazei pe care o conține.

Aceeași abordare va fi folosită și în timpul utilizării aplicației, pentru a adapta modelul acustic și în funcție de vocea utilizatorului, dar și în funcție de condițiile în care este realizată înregistrarea. În această situație drept fișiere utilizate pentru adaptare, vom folosi înregistrările încărcate de utilizator, iar rezultatul va fi reprezentat de un nou model acustic personalizat.

Astfel, pentru fiecare utilizator care a trimis cel puțin o înregistrare, vom avea un model acustic unic, utilizat atât procesul de recunoaștere vocală, cât și cel de transcriere fonetică.

---

<sup>33</sup> Mai mult detalii despre adaptarea modelului acustic cu CMU Sphinx: <https://cmusphinx.github.io/wiki/tutorialadapt/>



Fig. 5.3.7: Transcriere fonetică realizată de sistemul neadaptat, transcrierea fonetică realizată de sistemul adaptat doar cu baza de date de antrenament și transcrierea fonetică realizată de sistemul adaptat cu baza de date de antrenament și cu vocea utilizatorului pentru cuvântul flamingo (F L AH M IH NG G OW)

### 5.3.3 Concluzii

În concluzie observăm că odată cu adaptarea modelului acesta ajunge să returneze modele din ce în ce mai apropiate de realitate și să includă greșeli obișnuite și previzibile. Exemplele prezentate anterior sunt realizate pornind de la înregistrările aceluiași utilizator, în același mediu la intervale relativ scurte de timp, și totuși în primul exemplu observăm o însiruire de sunete care nu este deloc corelată cu transcrierea corectă a cuvântului propus drept exercițiu, pe când, chiar și după antrenarea modelului acustic cu o bază relativ mică de date care să-l expună și greșelilor de pronunție, observăm o îmbunătățire semnificativă, iar odată cu introducerea în model și a caracteristicilor vocale ale utilizatorului, transcrierea are o acuratețe și mai ridicată.

Privind cele trei exemple putem afirma că cea mai mare îmbunătățire apare între varianta neadaptată și cea adaptată doar cu bază de date de antrenament.

O problemă care apare totuși utilizând această metodă este dată de faptul că modelul funcționează din ce în ce mai bine cu cât există mai multe date din partea utilizatorului, drept urmare în fazele incipiente când utilizatorul doar se familiarizează cu aplicația înregistrările trimise de acestea ar putea nici să nu treacă de faza de recunoaștere vocală, urmând ca ulterior, odată cu acumularea de date, această problemă să se remedieze.

De asemenea, se remarcă faptul că în procesul de adaptare nu am mai inclus un pas descris în secțiunea teoretică 3.2.2, mai exact modificarea dicționarului de limba astfel încât acesta să cuprindă și greșelile surprinse în pronunție în înregistrările vorbitorilor pentru care limba engleză este o limbă secundară, aceasta etapă ar fi condus la rezultate mai precise din punct de vedere al transcrierii, dar este

o etapă care se poate realiza doar de către, sau în prezență unui persoane cu experiență în domeniul lingvistic. În lipsa unei astfel de persoane, am sărit peste acest pas, pas care realizat greșit ar putea aduce cu sine o deteriorare a rezultatelor obținute.

Avantajul principal al acestei metode este dată de formatul rezultatului, mai exact transcrierea fonetică obținută din înregistrare și transcrierea fonetică corectă. Acest format de rezultat este foarte explicit și ușor de interpretat de orice tip de utilizator care accesează aplicația.

## 5.4 Evaluare pronunției utilizând Rețele Neuronale Convoluționale

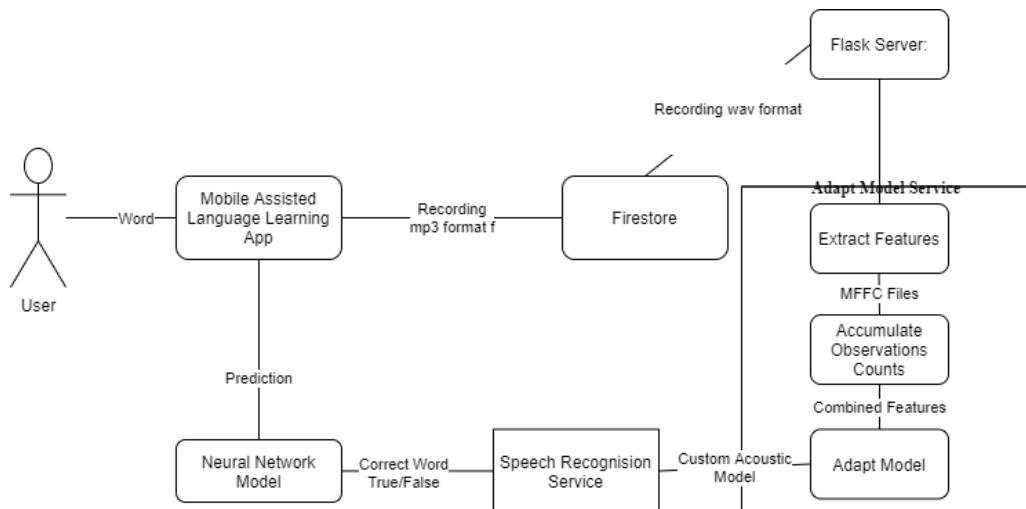


Fig. 5.4.1: Evaluarea pronunției utilizând Rețele Neuronale Convoluționale

### 5.4.1 Preprocesare date

Datele utilizate pentru antrenarea și testarea modelului sunt obținute tot apelând la The Speech Accent Archive<sup>34</sup>. Pentru antrenarea modelelor vom extrage din această arhivă, 25 de înregistrări ale unor vorbitori nativi de engleză care provin din Anglia și 25 de înregistrări ale unor vorbitori de limba română citind următoarea frază:

*“Please call Stella. Ask her to bring these things with her from the store: Six spoons of fresh snow peas, five thick slabs of blue cheese, and maybe a snack for her brother Bob. We also need a small plastic snake and a big toy frog for the kids. She can scoop these things into three red bags, and we will go meet her Wednesday at the train station.”*

<sup>34</sup> <http://accent.gmu.edu/>

Aceste înregistrări vor fi utilizate în două moduri.

- 1) Pe deoarete, vom utiliza frazele complete împărțite în fragmente de câte o secundă, fără a realiza această fragmentare după o regulă anume, cu excepția dimensiunii ferestrei de o secundă. Acest tip de date vor fi utilizate pentru antrenarea modelului care face diferența între o pronunție nativă și una nenativă, fără a oferi date suplimentare cu privire la greșeala de pronunție produsă.
- 2) Pe de altă parte, vom extrage cuvintele individuale din înregistrări și le vom folosi ca date de intrare. Acest tip de date va fi utilizat pentru a antrena câte un model pentru fiecare cuvânt întâlnit (cu excepția cuvintelor vide<sup>35</sup> și numele proprii), un model care ne va spune dacă cuvântul este pronunțat corect sau dacă există o anumită greșală comună pentru vorbitorii de limba română care învață engleză drept a doua limbă. Aceeași bază de date va fi utilizată și pentru antrenarea același model prezentat anterior, care face diferența dintre pronunția nativă și nenativă.

Pentru extragerea cuvintelor din fișiere vom parcurge următorii pași. În primul rând vom folosi utilitarul Montreal Forced Aligner descris în capitolul 4.5 pentru a obține intervalele de timp în care este întâlnit fiecare cuvânt. Apoi pornind de la aceste intervale vom împărți fișierul audio și vom obține astfel câte o bază de date pentru fiecare cuvânt.

Am recurs la această variantă deoarece la momentul actual nu este disponibilă o bază de date care să conțină înregistrări individuale pentru mai multe cuvinte în limba engleză pronunțate de nativi români.

În cazul modelului care distinge pronunția nativă în engleză, de pronunția unui român care învață engleză drept limbă secundară, eticheta datelor de intrare va fi stabilită în funcție de fișierul audio din care este extras fragmentul de o secundă, respectiv cuvântul.

Pentru modelul care determină dacă un anume cuvânt este pronunțat corect sau dacă conține o anumită greșală specifică celor care au drept limbă maternă, limba română, vom eticheta datele de antrenament utilizând informațiile oferite de The Speech Accent Archive, deoarece această bază de date furnizează nu doar înregistrări, dar și adnotări ale greșelilor generale identificate în pronunție.

---

<sup>35</sup> Cuvinte vide sau stop words sunt cuvinte filtrare înainte de procesarea limbajului natural

Please call Stella. Ask her to bring  
**these things** with her from the store:  
Six spoons of fresh snow peas, **five**  
thick slabs of blue cheese, and maybe  
a snack for her brother Bob. We also  
need a small plastic snake and a big  
toy frog for the kids. She can scoop  
**these things** into three red bags, and  
we will go meet her Wednesday at the  
train station.

Key:  
**blue** = potential areas for this generalization  
**red** = actual areas for this generalization

Fig. 5.4.2: Adnotări ale locurilor în care apare fenomenul de desonorizare finală (final obstruent devoicing) într-o înregistrarea unui vorbitor de limba română

Observăm că există două tipuri de notații, cele cu albastru și cele cu roșu, semnificația lor fiind următoarea, cele cu roșu sunt zone în care greșeala este cu siguranță prezentă, iar cele cu albastru sunt zone în care există probabilitatea existenței acelei erori. În timpul etichetării vom considera doar zonele notate cu roșu.

Eroare generală de pronunție	Definiție	Cuvinte etichetate cu respectivă greșeală
Desonorizare finală	Sunetele devin mute în fața consoanelor mute	<b>please, things, five, slabs, cheese, kids, bags, big</b>
Coborârea unei vocale	Vocala este mai puțin sonoră	<b>ask, snack, plastic, spoons, wednesday</b>
Ridicarea unei vocale	Vocala se aude prea puternic	<b>call, small, frog, six, thick</b>
Scurtarea vocalei	O vocală lungă este pronunțată ca o vocală scurtă	<b>blue, peas, scoop, snow, station, train, toy</b>
R graseiat	A pronunță sunetul «r» din gât (cu ajutorul uvulei), aşa cum îl pronunță francezii din nord	<b>bring, brother, three, red, fresh, store</b>
Inserarea vocală	Inserarea unei vocale suplimentare	<b>cheese</b>

Fig. 5.4.3: Cuvintele pentru care vom realiza modele și tipul de eroare care poate fi identificat în acestea

Din căte se poate observa, există cuvinte din fraza inițială care nu se regăsesc în tabel (cu excepția pronumelor proprii și a cuvintelor vide), acest lucru se datorează faptului că acestea nu au o greșeală generală asociată și drept urmare nu vor putea fi supuse unei clasificări în acest sens.

Ulterior toate datele utilizate, vor fi transpusă în spectrograme Mel, descrise în capitolul 3.3.1, iar aceste spectrograme vor fi utilizate ulterior drept date utilizate în procesul de antrenare a modelului, reducând problema la o problemă de clasificare a imaginilor.

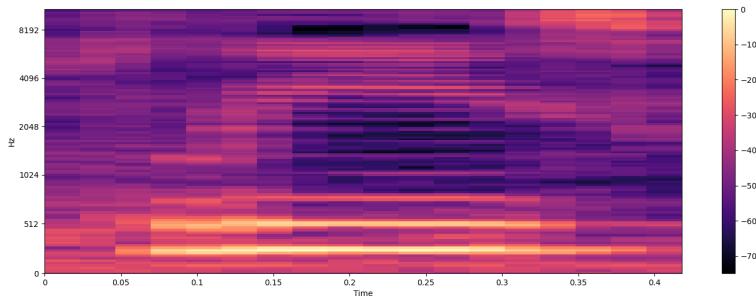


Fig. 5.4.4: Spectrogramă Mel pentru cuvântul “Please” în pronunția unui nativ român

Spectrogramele vor fi realizate utilizând librăria Librosa<sup>36</sup> disponibilă în Python 3. Pentru toate modelele încercate vom împărți mulțimea de date astfel, 80% date de antrenament și 10% date de validare și 10% date de test, iar dimensiunea spectrogramelor Mel va fi modificată în funcție de dimensiunea datelor de intrare acceptate de modelul utilizat.

### 5.4.2 Modele pre-instruite utilizate

Pentru realizarea proiectului de învățare prin transfer vom folosi două modele antrenate utilizând datele de la ImageNet, care sunt specializate în clasificarea de imagini. Parametrii obținuți din aceste modele vor fi folosiți pentru extragerea caracteristicilor din spectrogramele Mel pe care ne propunem să le analizăm. Având aceste caracteristici vom putea configura în funcție de necesități straturile conectate complet în care are loc clasificare.

Straturile conectate complet introduse în continuarea modelului pre-instruit vor avea următoarea structură, structură care va fi folosită pentru toate modelele încercate. Rezultatul obținut în urma extragerii caracteristicilor va fi aplatizat și prezentat drept date de intrare pentru o rețea formată din următoarele straturi: un strat ce folosește drept funcție de activare funcția Relu<sup>37</sup> și 512 neuroni și un strat care va returna probabilitățile finale de asignare într-o clasă utilizând funcția de activare softmax<sup>38</sup>.

---

<sup>36</sup> Mai multe detalii despre librăria Librosa pot fi găsite în documentația oficială disponibilă la adresa: <https://librosa.org/doc/main/generated/librosa.feature.melspectrogram.html>

<sup>37</sup> Funcție de activare care va returna valoarea de intrare, în cazul în care aceasta este pozitivă și 0, altfel.

<sup>38</sup> Funcție de activare care asignează probabilități într-o problemă de clasificare a mai multor clase

## VGG16

Rețeaua VGG16 (de la cele 16 straturi neuronale) a fost propusă de K. Simonyan și A. Zisserman de la Universitatea Oxford în lucrarea “*Very Deep Convolutional Networks for Large-Scale Image Recognition*”. La momentul actual tipul de rețele VGG ocupă primele poziții în ceea ce privește sarcina de clasificării imaginilor. Ideea principală din spatele rețelelor VGG este dimensiunea foarte mică a filtrelor din straturile conoluționale, comparativ cu alte modele cunoscute, mai exact  $3 \times 3$  cu deplasare de un pas, dar și a straturilor de max pooling, unde ferestrele sunt de dimensiune  $2 \times 2$ . Datele de intrare sunt reprezentate de o imagine cu dimensiunea de  $224 \times 224$ . Datele de ieșire în varianta originală sunt reprezentate de un tensor cu dimensiunea  $1000 \times 1$ , dar în cazul nostru dimensiunea va fi  $2 \times 1$ . O rețea de tipul VGG16 are în total 138 de milioane de parametri antrenabili.

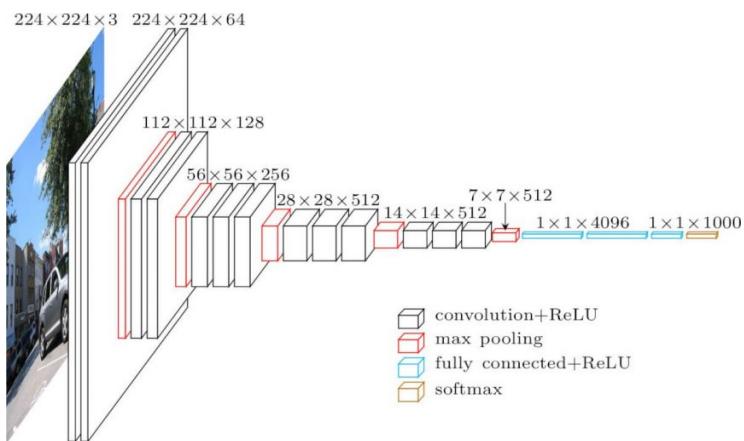


Fig. 5.4.5: Arhitectura modelului VGG16<sup>39</sup>

## ResNet50

Rețeaua ResNet50 este o varianta cu 50 de straturi a tipului de rețea ResNet. Conform [14] aceste rețele au apărut pentru a rezolva problema gradientului care dispare<sup>40</sup>, iar soluția a fost introducerea unui bloc rezidual, astfel încât la ieșirea dintr-un bloc de conoluții se adaugă și intrarea în bloc, acestea sunt “scurtături”, care permit realizarea unor rețele mai adânci. La fel ca în cazul VGG16, datele de intrare sunt reprezentate de o imagine cu dimensiunea de  $224 \times 224$ . Datele de ieșire, în

<sup>39</sup>[http://www.master-taid.ro/Cursuri/MLAV\\_files/Retele%20Convolutionale%20-%20Convolutional%20Neural%20Networks%20\(CNNs\).html](http://www.master-taid.ro/Cursuri/MLAV_files/Retele%20Convolutionale%20-%20Convolutional%20Neural%20Networks%20(CNNs).html)

<sup>40</sup> Scaderea puternică a gradientilor odată cu avansarea în rețea în etapa de propagare înapoi.

varianta originală sunt reprezentate de un tensor cu dimensiunea  $1000 \times 1$ , dar în cazul nostru dimensiunea va fi  $2 \times 1$ . O rețea de tipul ResNet50 are peste 23 de milioane de parametri antrenabili.

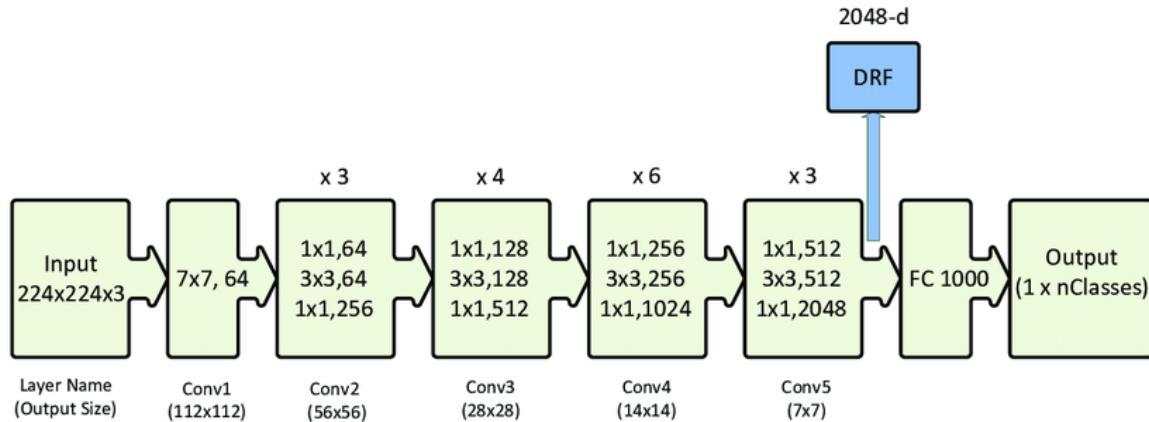


Fig. 5.4.6: Arhitectura modelului ResNet50<sup>41</sup>

### 5.4.3 Experimente

În continuare vom prezenta rezultate obținute în urma experimentelor prezentate în capitolul 3.3.3, atât pentru datele de antrenament, cât și pentru datele de test.

Valorile prezentate în continuare sunt obținute pentru antrenare utilizând grupuri de dimensiune 3 și 80 de epoci de antrenament. Aceste valori au fost alese în unei serii de experimente cu următoarele valori: 1, 3 și 5 pentru batch și 45, 60 și 80 pentru numărul de epoci de antrenament.

Funcția de cost, funcția care ne spune cât de bine sau cât de rău sunt clasificate datele, care va fi folosită pentru toate experimentele prezentate este funcția entropie încrucișată. Funcția entropie încrucișată este una din cele mai folosite funcții de cost și are valori cu atât mai mari cu cât diferența dintre două distribuții de probabilitate este mai mare.

Pentru evaluare modelelor vom analiza valorile acurateței și valorile funcției de cost, pentru mulțimea de date de antrenament, pentru mulțimea de date de validare și pentru cea de test.

#### **Model care clasifică cuvintele pronunțate corect, de cuvintele care conțin o anume eroare generală**

Prima variantă de evaluare a pronunției încercată presupune realizarea unui model pentru fiecare cuvânt regăsit în baza de date de antrenament, model capabil să clasifice cuvintele pronunțate

<sup>41</sup>[https://www.researchgate.net/figure/ResNet-50-architecture-26-shown-with-the-residual-units-the-size-of-the-filters-and-f\\_ig1\\_338603223](https://www.researchgate.net/figure/ResNet-50-architecture-26-shown-with-the-residual-units-the-size-of-the-filters-and-f_ig1_338603223)

corect, de cuvintele care conțin o anumită greșală generală. Aceste greșeli însotite de cuvintele care le conțin pot fi găsite în Fig. 5.4.2. Pentru fiecare cuvânt, din cele 31 păstrate din înregistrare, avem la dispoziție doar 50 de instanțe de antrenament, ceea ce înseamnă în medie 1 minut de înregistrări disponibile pentru fiecare cuvânt.

Rezultatele obținute sunt prezentate în continuare:

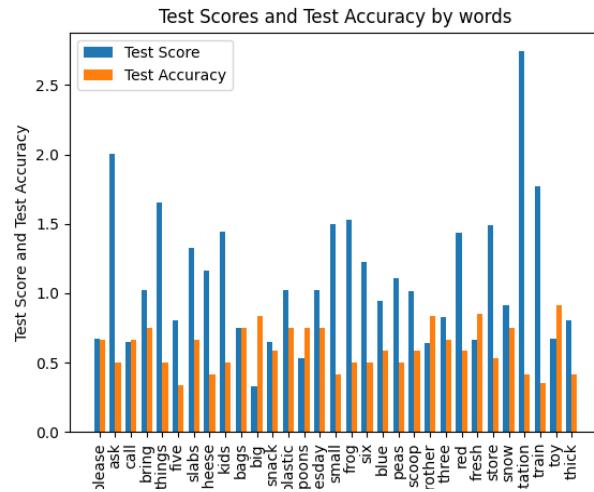


Fig. 5.4.7: Valorile pentru scorul și acuratețea modelelor pentru datele de test utilizând VGG16

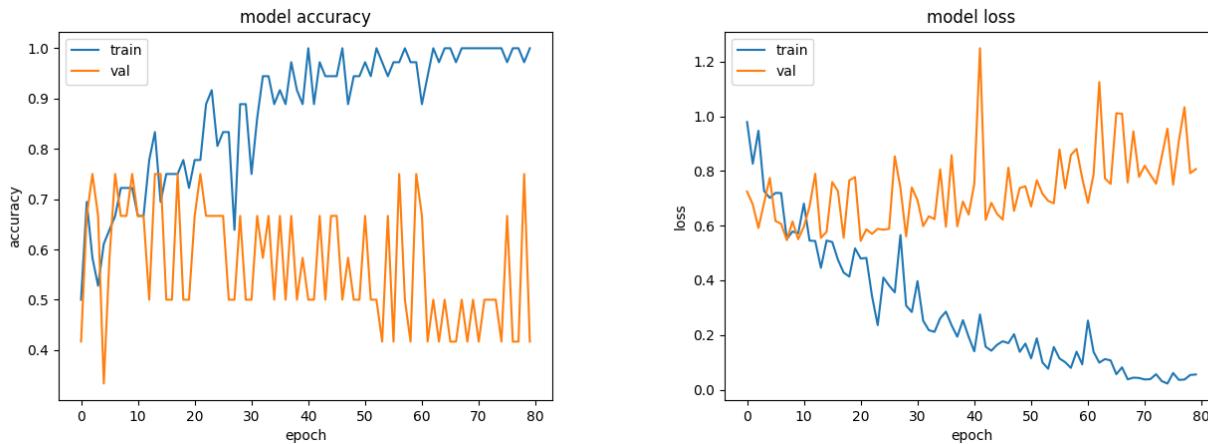


Fig. 5.4.8: Valorile pentru scorul și acuratețea modelelor pentru datele de antrenament și de validare utilizând VGG16 pentru cuvantul thick

Din cale se poate observa din Fig. 5.4.7 acest model se confruntă cu fenomenul de overfitting, rezultatele obținute pe mulțimea de date de antrenament sunt semnificativ mai bune, decât cele obținute utilizând mulțimea de date de validare și cea de test. Acest lucru se datorează numărului mic de instanțe de antrenament pe care le avem la dispoziție, 50 pentru fiecare cuvânt, care nu îi permit modelului să

ajungă în punctul în care să facă o generalizare cu privire la datele oferite la intrare. Acest fenomen se manifestă în felul următor: în primul rând acuratețea calculată utilizând datele de validare rămâne constant între anumite valori, fără a avea un trend ascendent, în timp ce valoarea funcției de cost, valoare care trebuie minimizată, crește. Același fenomen observat în cazul mulțimii de date de validare, se observă și în cazul datelor de antrenament, unde observăm valori mari ale funcției de cost și o diferență de acuratețe de aproximativ 0.4 între mulțimea de date de antrenament și cea de test.

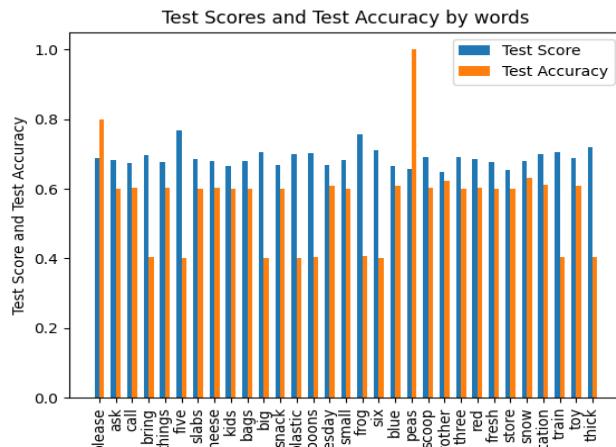


Fig. 5.4.9: Valorile pentru scorul și acuratețea modelelor pentru datele de test utilizând ResNet50

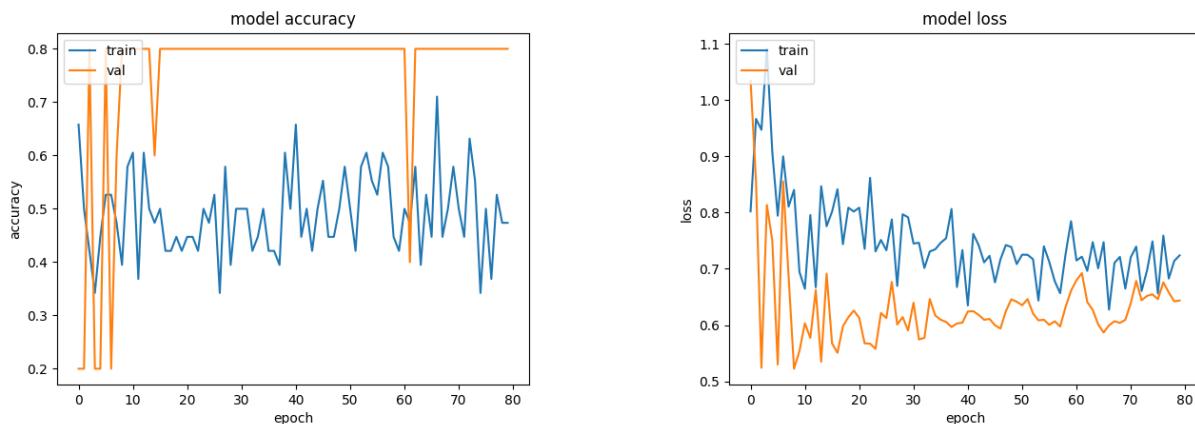


Fig. 5.4.10: Valorile pentru scorul și acuratețea modelelor pentru datele de antrenament și de validare utilizând ResNet50 pentru cuvantul thick

În ceea ce privește varianta de antrenare a unui model pornind de la o rețea de tipul ResNet50, se poate observa în Fig. 5.4.9 fenomenul opus fenomenului de overfit, mai exact, avem de-a face cu fenomenul de underfit, lucru care poate fi dedus din faptul că în graficul pentru acuratețe, avem valori mai mari pentru acuratețea mulțimii de date de validare, comparativ cu acuratețea obținută pe mulțimea

de date de antrenament. Apariția acestui fenomen ne spune că modelul nu reușește să surprindă suficient de bine relația dintre datele de intrare și datele de ieșire. Acest fenomen apare în momentul în care utilizăm ResNet50, deoarece comparativ cu rețeaua VGG16, care are 138 de milioane de parametrii antrenabili, rețeaua ResNet50 are doar puțin peste 23 de milioane de parametrii și drept urmare nu reușește să surprindă complexitatea datelor primite. Problema numărului de parametrii este dublată, ca în cazul rețelei VGG16, de problema numărului mic de instanțe de antrenament. În cazul anterior nu aveam suficiente date pentru a putea realiza o generalizare, în cazul de față nu avem suficiente date pentru a realiza corelații între datele de intrare și cele de ieșire, deși modelul încearcă să învețe aceste legături.

În ceea ce privește rezultatele obținute pe mulțimea de date de test, prezentate în figura 5.4.8, se observă același fenomen de underfit, prin faptul că avem, pentru o mare parte din cuvinte, valori ale acurateței, mai mari decât cele obținute în cazul mulțimii de date de antrenament, dar și deoarece chiar dacă valorile acurateței sunt ridicate, valorile funcției de cost rămân în continuare și ele în jurul nivelului de 0.7. Se observă, de exemplu, pentru cuvântul *peas*, chiar o valoare de 1 pentru acuratețe, care poate fi motivată de faptul că având 50 de instanțe de antrenament pentru fiecare cuvânt, setului de date de test care reprezintă 10% din aceste instanțe, îi sunt asociate doar 5 instanțe.

### **Model care clasifică cuvintele pronunțate corect, de cele pronunțate greșit utilizând drept date de antrenare cuvinte extrase din înregistrări**

Cea de-a treia variantă va încerca să identifice cuvintele pronunțate corect și cele pronunțate greșit, utilizând cuvintele extrase pentru antrenarea modelelor prezentate anterior. Având în vedere că avem 31 de cuvinte și 50 de înregistrări pentru fiecare, vom avea 1550 de instanțe de antrenament, care sunt echivalentul a aproximativ 30 de minute de înregistrare.

Rezultatele obținute sunt prezentate în continuare:

Test accuracy	Test score
0.8053	0.4641

Fig.5.4.11: Valorile pentru scorul și acuratețea modelelor pentru datele de test utilizând VGG16

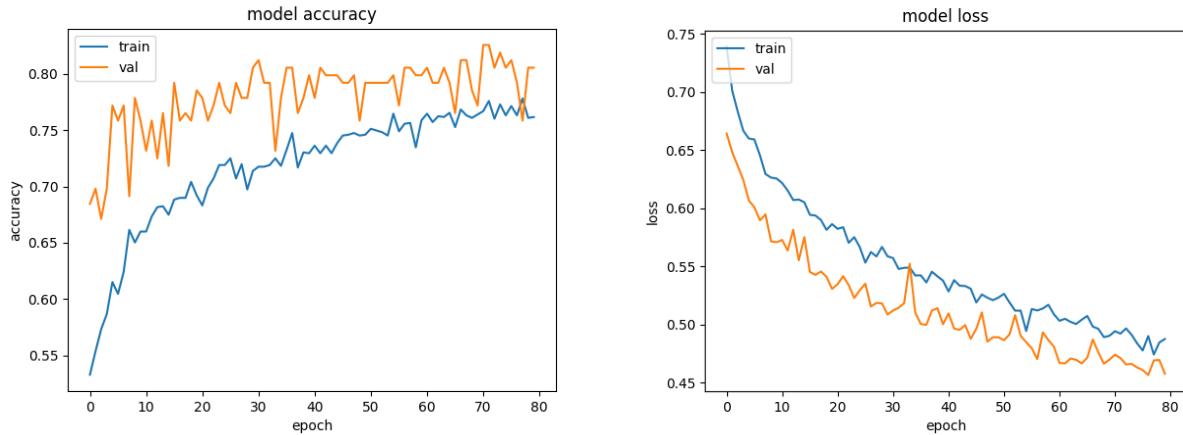


Fig. 5.4.12: Evoluția acurateții obținute pentru mulțimea de date de antrenament și pentru cel de validare utilizând VGG16

Din Fig. 5.4.11 în care este reprezentată evoluția acurateței pentru mulțimea de date de antrenament și cea de validare, se poate observa în continuare un fenomen de underfit, valori cu aproximativ 0.05 mai mari pentru mulțimea de date de validare comparativ cu mulțimea de date de antrenament. În cazul de față această diferență, mică comparativ cu diferența prezentă în Fig. 5.4.9, poate fi justificată de faptul că mulțimea de date de validare reprezintă doar 10% din numărul total de date disponibile. Apariția acestui fenomen poate fi motivată în continuare de numărul instanțe de antrenament, 1550, număr de instanțe mic comparativ cu nivelul de complexitate al problematici abordate. Se observă, totuși, o corectare a acestui fenomen raportat la metoda abordată anterior, remediere motivată de cele 1500 instanțe de antrenament suplimentare.

Analizând graficul evoluției valorii funcției de cost, care prezintă un trend descendent, putem afirma că modelul evoluează pe parcursul procesului de învățare și surprinde parametrii care realizează legătura dintre datele de intrare și datele de ieșire.

Această corectare a fenomenului de underfit, raportat la metoda abordată anterior se observă și la nivelul valorilor acurateței și a funcției de cost pentru mulțimea de date de test, unde valoarea de 0.8053 pentru acuratețe nu este semnificativ mai mare decât valorile obținute pentru mulțimea de date de antrenament, iar valoarea funcției de cost se află sub pragul de 0.5.

Test accuracy	Test score
0.6510	0.6719

Fig.5.4.13: Valorile pentru scorul și acuratețea modelelor pentru datele de test utilizând ResNet50

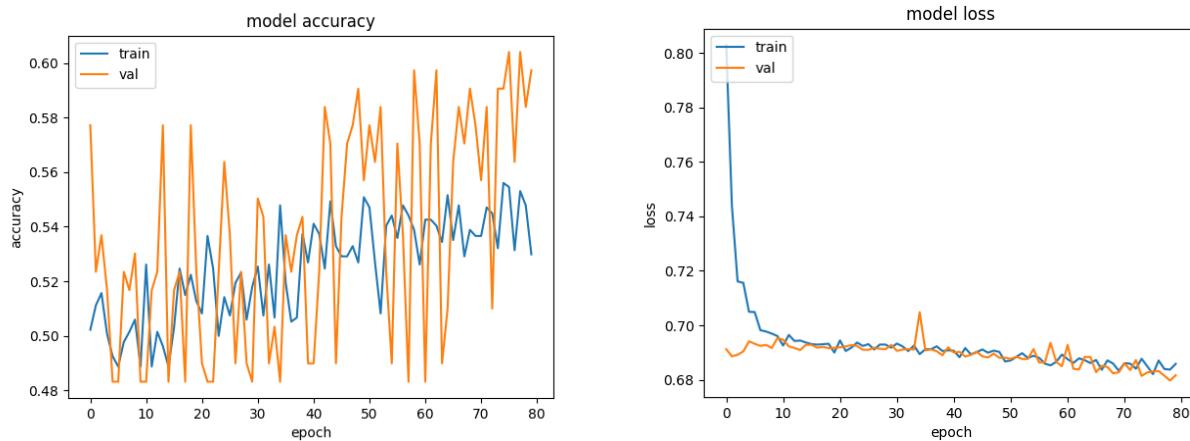


Fig. 5.4.14: Evoluția acurateței obținute pentru mulțimea de date de antrenament și pentru cel de validare utilizând ResNet50

În ceea ce privește varianta care utilizează un modelul pre-instruit ResNet50, observăm din Fig.5.4.12 că valorile acurateței, alternează în intervalul 0.48 - 0.58, pentru mulțimea de date de validare și între 0.48 - 0.55 pentru mulțimea de date de antrenament, fără a fi prezent un trend evident ascendent. Având în vedere acest lucru putem afirma că modelul nu reușește să surprindă caracteristicile datelor oferite la intrare. Acest lucru este susținut și de faptul că după un număr de 10 epocii, valoarea funcției de cost se situează între aceleași valori, atât pentru mulțimea de date de antrenament, cât și pentru mulțimea de date de validare și drept urmare putem concluziona că modelul stagnează în ceea ce privește procesul de învățare.

La fel ca în cazul modelului prezentat anterior, rezultatele mai slabe (cu 0.2 mai puțin pentru acuratețea mulțimii de date de test) obținute utilizând ResNet50 se datorează numărului mai mic de parametrii antrenabili, comparativ cu VGG16, utilizați de acest tip de rețea.

Un alt motiv care ar putea conduce la aceste rezultate este dat de faptul că, analizând Fig. 5.4.4 și Fig. 5.4.5, în care sunt descrise arhitectura rețelei VGG16, respectiv ResNet50, observăm că rețeaua VGG16 aplică convoluții de dimensiuni mici, 3 x 3, chiar asupra imaginii date la intrare, pe când în cazul ResNet50, se realizează un proces de agregare și se aplică convoluții de dimensiune mai mare 7 x

7, anterior aplicării mai multor convoluții de dimensiunea 3 x 3. Aceste caracteristici ale arhitecturii celor două rețele au două consecințe, pe de-o parte procesul de antrenare al rețelei VGG16 va dura mai mult comparativ cu antrenarea rețelei ResNet50, dar în ceea ce privește sarcina curentă, analiza unei spectrograme, convoluțiile de dimensiunea 3 x 3, aplicate imaginii originale vor surprinde mai bine particularitățile acesteia.

### **Model care clasifică cuvintele pronunțate corect, de cea pronunțate greșit utilizând drept date de antrenament fragmente de câte o secundă din înregistrări**

Cea de-a treia variantă va încerca să clasifice cuvintele pronunțate greșit de cele pronunțate corect, utilizând drept date de antrenament fragmente cu dimensiune de câte o secundă din înregistrări. Împărțind cele 50 de înregistrări, vom obține 1200 de înregistrări folosite ca date de antrenament, echivalentul a 20 minute de înregistrare.

Avantajul adus de această abordare, antrenare utilizând câte o secundă de înregistrare, este reprezentat de faptul că datele sunt consistente în ceea ce privește dimensiunea. Pentru acest model avem mai puține date de antrenament 1200, comparativ cu cele 1550 disponibile în cazul anterior, iar datele analizate sunt mai puțin complexe, drept urmare a micșorării ferestrei de timp analizate.

Rezultatele obținute sunt prezentate în continuare:

Test accuracy	Test score
0.5369	3.419

Fig.5.4.15: Valorile pentru scorul și acuratețea modelelor pentru datele de test utilizând VGG16

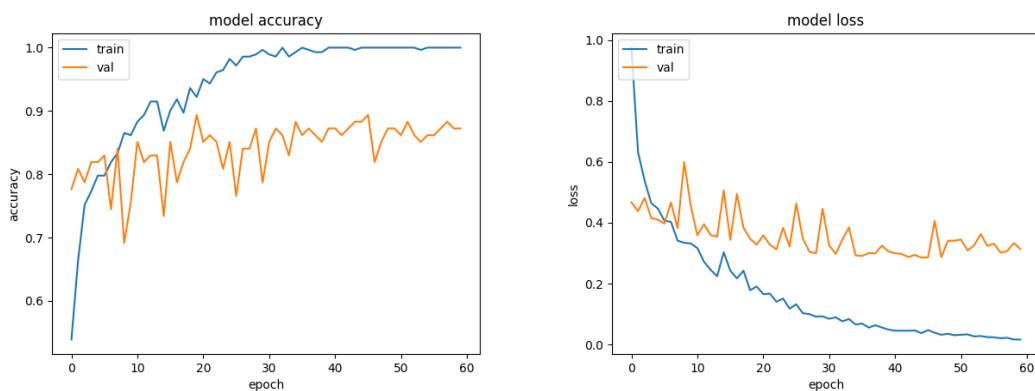


Fig. 5.4.16: Evoluția acurateței obținute pentru mulțimea de date de antrenament și pentru cel de validare utilizând VGG16

Din Fig. 5.4.15 și 5.4.14 putem observa că avem de-a face, din nou, cu fenomenul de overfitting, manifestat prin faptul că valoarea pentru acuratețea mulțimii de date de antrenament ajunge la 1, iar valoarea pentru funcția de cost, pe aceeași mulțime de date atinge valoare 0. Drept urmare modelul clasifică perfect mulțimea de date de antrenament, dar în ceea ce privește mulțimea de date de validare și cea de test nu se poate afirma același lucru, având valori de 0.85 și respectiv 0.53 pentru acuratețe, iar pentru valoarea funcției de cost, 0.4 și 3.419.

Un motiv care conduce la apariția acestui fenomen este faptul că rețeaua VGG16 are prea mulți parametri, raportat la complexitatea datelor analizate, comparativ cu situația anterioară în care erau analizate segmente mai lungi de timp, între 3 și 4 secunde. Această problemă este dublată și de reducerea numărului de instanțe de antrenament, de la 1550, la 1200.

Test accuracy	Test score
0.5503	0.9227

Fig. 5.4.17: Valorile pentru scorul și acuratețea modelelor pentru datele de test utilizând ResNet50

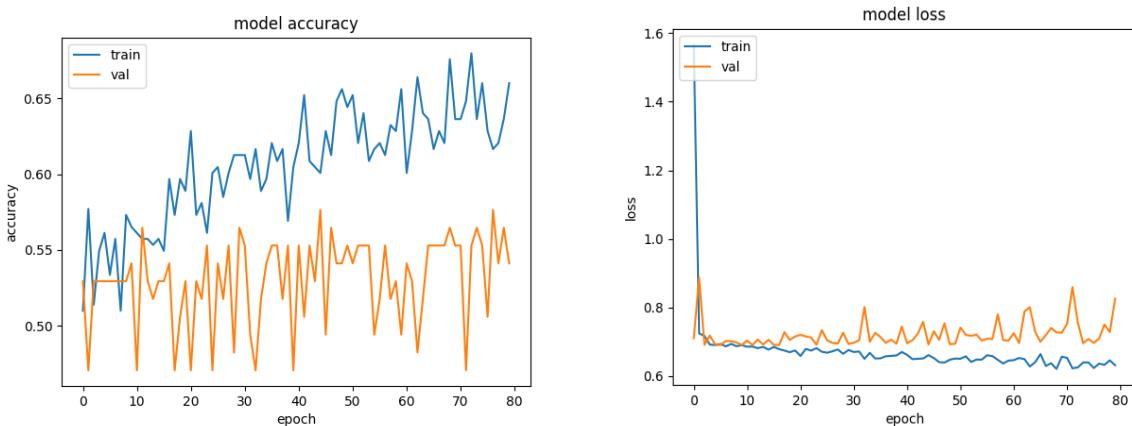


Fig. 5.4.18: Evoluția acurateței obținute pentru mulțimea de date de antrenament și pentru cel de validare utilizând ResNet50

Analizând tabelul din Fig. 5.4.16 și graficele din Fig. 5.4.17, observăm că pentru cazul antrenării rețelei de tipul ResNet50, se manifestă același fenomen, ca în cazul modelului prezentat anterior, mai exact, valorile acurateței atât pentru mulțimea de date de antrenament, cât și pentru

mulțimea de date de validate, alternează între valorile, 0.5 - 0.65 și 0.45 - 0.55, iar valorile funcției de cost nu scad sub valoarea de 0.6 și oscilează foarte puțin.

Drept urmare concluzionăm că modelul nu evoluează în ceea ce privește sarcina de găsire a caracteristicilor care fac legătura între datele de intrare și eticheta acestora. Justificarea este aceeași ca în cazul modelului prezentat anterior, pentru antrenarea rețelei de tipul ResNet50.

#### 5.4.4 Concluzii

Având în vedere rezultatele experimentelor și interpretarea acestora, putem trage o serie de concluzii generale în ceea ce privește antrenarea unei rețele neuronale convoluționale pentru evaluarea pronunției.

În primul rând, putem observa că rețeaua VGG16, converge în mai puține iterații. Pe parcursul celor 80 de iterații observate, rețeaua VGG16 converge în cazul tuturor celor 3 modele antrenate, comparativ cu rețeaua ResNet50, unde în toate situațiile ajunge în punctul în care fluctuează într-un anume interval, fără a prezenta un trend evident ascendent în ceea ce privește acuratețea. Deci, putem afirma că rețeaua VGG16 extrage informații mai relevante din datele de intrare, comparativ cu ResNet50, iar acest lucru se datorează în special complexității rețelei și numărului mare de parametrii antrenabili disponibili. Aspectul negativ adus de această complexitate este dat de faptul că după cum am observat, atât în cazul primului model, cât și în cazul celui de-al treilea, apare fenomenul de overfitting.

În al doilea rând, un aspect care merită menționat este faptul că în cazul rețelei VGG16 antrenarea poate dura aproximativ două ore pentru 80 de epoci cu 1550 de instanțe în mulțimea de date de antrenament, iar pentru aceiași parametri antrenarea rețelei ResNet50 durează aproximativ o oră și 20 de minute. Totuși această diferență în ceea ce privește timpul se reflectă la nivelul milisecundelor în cazul predicției, diferență neglijabilă pentru utilizator, deci drept urmare nu este un motiv suficient pentru a renunța la alegerea rețelei de tipul VGG16.

În concluzie, având în vedere atât rezultatele în ceea ce privește acuratețea modelului, cât și diferențele de timp observate, vom alege să folosim cel de-al doilea model, antrenat utilizând înregistrările cuvintelor și rețeaua neuronală convoluțională VGG16, care ajunge la valoarea de 0.8 pentru acuratețea mulțimii de date de antrenament.

Pe lângă acuratețea modelului, un alt avantaj al acestei abordări este dat de faptul că fragmentele de înregistrare trimise de utilizator, nu necesită o procesare, înaintea realizării

spectogramelor, spre deosebire de cea de-a treia abordare unde este necesară fragmentarea în segmente de câte 1 secundă. De asemenea, comparativ cu primul model prezentat, cel care realizează un model pentru fiecare cuvânt, modelul ales are avantajul de a permite aplicației să extindă baza de date cu exerciții oricât de mult, fără a depinde de datele de antrenament disponibile.

## 6. Evaluare și feedback din partea utilizatorilor

Pentru a colecta părerea utilizatorilor atât cu privire la utilitatea aplicațiilor de învățare a unei limbi străine asistată de calculator, cât și cu privire la aplicația propriu-zisă prezentată în cadrul acestei lucrări am realizat un test de utilizabilitate. Un grup de 6 persoane au luat parte la acest test, test care a decurs în următorul mod. Persoanele, toate cu experiență în utilizarea unui telefon intelligent, au primit în prealabil instrucțiuni legate de modul de utilizare a aplicației, dar și o descriere a funcționalităților acestora, iar ulterior au fost îndrumate să folosească aplicație în ritmul propriu.

După ce utilizatorii au considerat că au reușit să parcurgă toate funcționalitățile aplicației, aceștia au completat un formular care cuprinde 19 întrebări, întrebări grupate în 4 categorii: întrebări legate de profilul utilizatorului, întrebări legate de experiența și interesul acestora față de învățarea asistată de mobil, întrebări cu privire la aplicația testată în etapa descrisă anterior și întrebări legate de posibile viitoare îmbunătățiri ale acestei aplicații.

### 6.1 Profilul utilizatorilor

Aplicația prezentată, și în general procesul de învățare a unei limbi străine asistată de mobil, este un proces care poate fi realizat de persoane din toate categoriile de vârstă, indiferent de ocupația acestora și de cunoștințele ulterioare în acest domeniu.

Aceste întrebări au scopul de a înțelege mai bine utilizatorul obișnuit al acestui tip de aplicații.

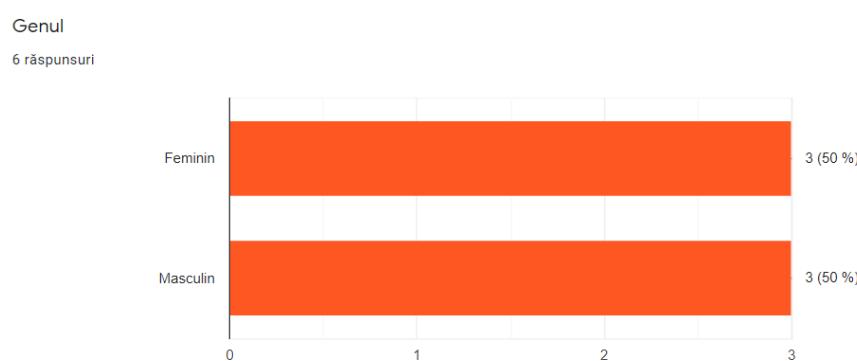


Fig. 6.1.1: Grafic privind genul persoanelor care au testat aplicația



Fig. 6.1.2: Grafic privind vârsta persoanelor care au testat aplicația

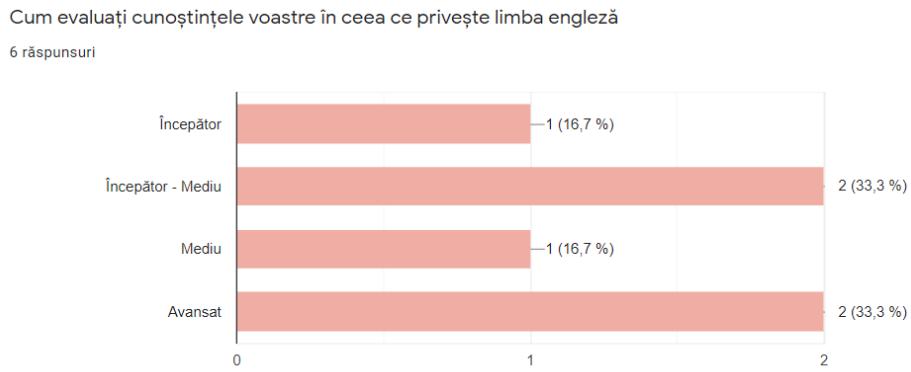


Fig. 6.1.3: Grafic privind autoevaluarea nivelului de cunoștințe în ceea ce privește limba engleză a persoanelor care au testat aplicația

Din căte se observă utilizatorii care au încercat aplicația sunt în egală măsură de sex feminin și masculin. În ceea ce privește vârsta sunt acoperite toate intervalele cu excepția persoanelor cu vârstă între 0 și 12 ani. Pentru nivelul de cunoștințe în limba engleză există câte o persoană pentru fiecare nivel, începând cu începători și încheind cu utilizatorii experimentați. Toți utilizatorii au drept limbă maternă limba română.

## 6.2. Interesul față de învățarea unei limbii străine asistată de mobil

În această categorie este analizat interesul persoanelor care iau parte la acest studiu față de aplicațiile de învățare a asistată de mobil a unei limbi străine. Întrebările surprind modul în care

utilizatorii privesc procesul de învățare a unei limbi străine, dar și experiența acestora în ceea ce privește utilizarea altor aplicații de învățarea unei limbi străine disponibile la momentul actual.

Când învățați o limbă străină care este metoda la care aplelați în mod obișnuit?

6 răspunsuri

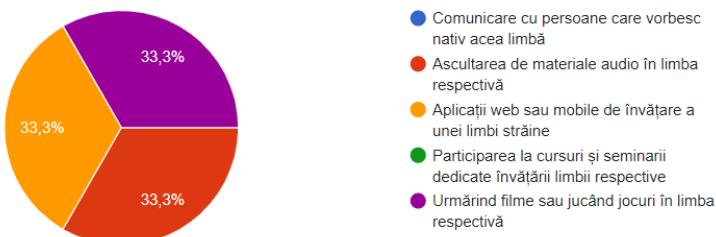


Fig. 6.2.1: Grafic privind modul în care persoanele care au testat aplicația învață o limbă nouă

Când dorîți să învățați o limbă străină care este aspectul pe care dorîți să îl dezvoltați în special?

6 răspunsuri

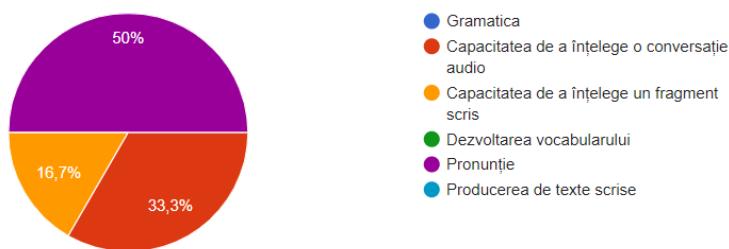


Fig. 6.2.1: Grafic privind aspectul pe care persoanele care au testat aplicația doresc să îl dezvolte în mod special

Aceste două grafice prezintă abordarea obișnuită a persoanelor care au testat aplicația în ceea ce privește învățarea unei limbi străine. Observăm că atât abordările, cât și aspectele de interes sunt foarte diverse. Metodele de învățare utilizate sunt în special legate de utilizarea tehnologiei, fie prin intermediul materialelor audio-video, a jocurilor sau a aplicațiilor dedicate acestei sarcini. De asemenea, observăm că jumătate din persoanele care au răspuns la chestionar doresc să-și îmbunătățească pronunția într-o limbă străină.

Considerați că aplicațiile web sau mobile pot fi introduse în procesul de învățare?

6 răspunsuri

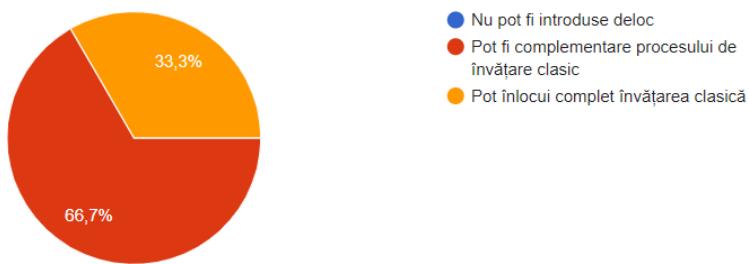


Fig. 6.2.3: Grafic privind opinia persoanelor care au testat aplicația cu privire la utilitatea aplicațiilor de învățare a limbilor străine

Pe o scară de la 1 la 5, unde 1 înseamnă deloc util și 2 înseamnă foarte util, cât de eficiente considerați că poate fi o aplicație mobilă de învățare a limbilor străine în ceea ce privește următoarele abilități:

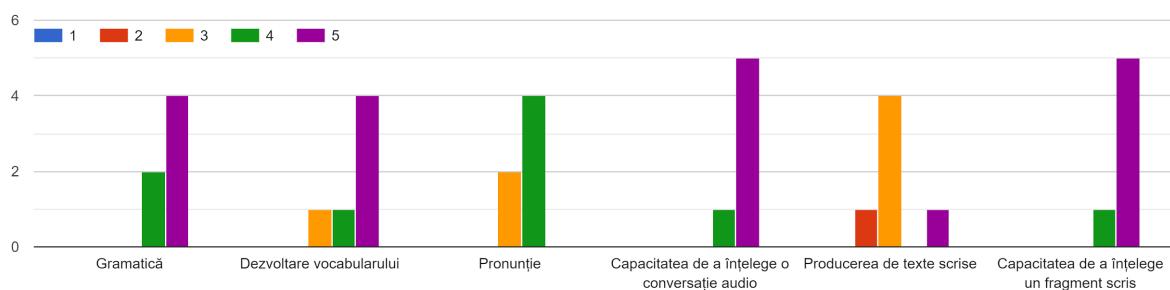


Fig. 6.2.4: Grafic privind opinia persoanelor care au testat aplicația cu privire la modul în care componentele implicate în învățarea unei limbi străine pot fi dezvoltate cu ajutorul unei aplicații

Din cîte putem observa în Fig. 6.2.3 și 6.2.4, persoanele care au testat aplicația sunt deschise în ceea ce privește aplicațiile de învățare a limbilor străine, aceștia considerând că acestea pot veni în ajutorul procesului clasic de învățare sau chiar să îl înlocuiască. De asemenea, aceștia consideră că acest tip de aplicații pot fi utile în special pentru dezvoltarea următorului tip de abilități: gramatică, dezvoltare vocabular, capacitatea de a înțelege o conversație audio și capacitatea de a înțelege un fragment scris. Aceștia consideră că și pronunția poate fi îmbunătățită în acest mod, dar ar putea exista o reținere ca urmare a faptului că această componentă nu este una cu tradiție în ceea ce privește aplicațiile de învățarea a limbilor străine, întrucât nu este la fel de ușor de integrat precum componente care nu necesită implicarea directă a utilizatorului.

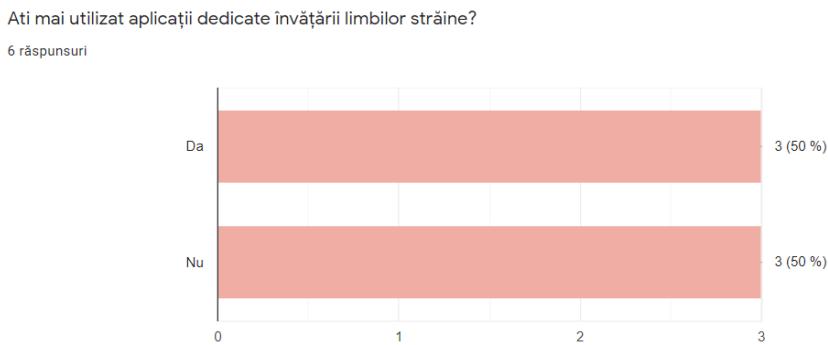


Fig. 6.1.8: Grafic privind experiența anterioară cu aplicații de învățare a limbilor străine

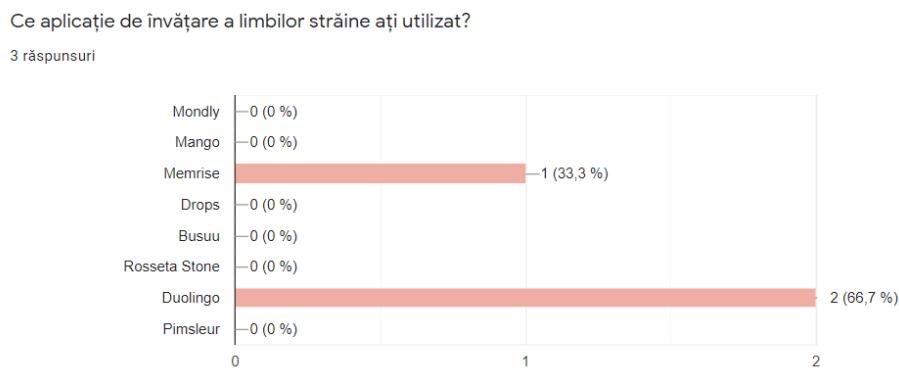


Fig. 6.1.9: Grafic privind aplicațiile de învățare a limbilor străine utilizate anterior

Din Fig. 6.1.8 și 6.1.9, observăm 50% din cei implicați în procesul de evaluare au cel puțin o experiență anterioară în ceea ce privește utilizarea de aplicații web sau mobile pentru învățarea limbilor străine. Preponderent a fost utilizată aplicația Duolingo, aplicație prezentată și în cadrul acestei lucrări în capitolul 2.4.

### 6.3 Experiența din timpul utilizării aplicației propuse

Această mulțime de întrebări urmărește să evaluateze experiența utilizatorilor supuși examinării cu privire la aplicația prezentată în această lucrare. Întrebările sunt menite să evaluateze atât utilitatea aplicației în ceea ce privește învățarea unei limbi străine, dar și modul în care s-a realizat interacțiunea cu aceasta, cât de ușor de utilizat și de intuitivă este aceasta.

Contul creat în cadrul aplicației a fost creat cu rolul de student sau având rolul de profesor?  
6 răspunsuri

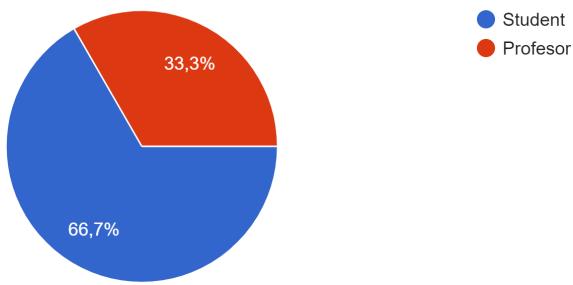


Fig.6.1.10: Grafic privind tipul de utilizator ales de persoana care a încercat aplicația

În care din zonele aplicației, zona destinată începătorilor și zona destinată utilizatorilor avansați de limba engleză, ati petrecut cel mai mult timp?  
6 răspunsuri

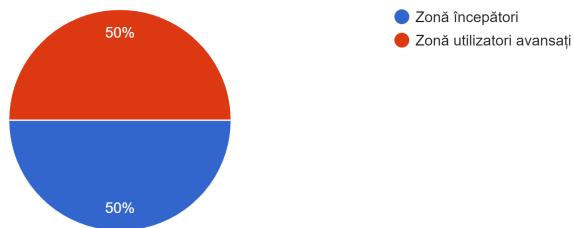


Fig.6.1.11: Grafic privind zona din aplicație în care utilizatorul a petrecut cel mai mult timp

Primele două întrebări, a căror răspuns este prezentat în Fig. 6.1.10 și 6.1.11 prezintă tipul de utilizator ales de fiecare persoană implicată în acest proces de evaluare a aplicației. Utilizatorii au ales tipul utilizatorului, student sau profesor, în funcție de necesitățile și de interese fiecărui. În funcție de tipul de utilizator ales aceștia au putut realiza activități diferite în procesul de rezolvare și de asignare a temelor. Observăm că mai mulți utilizatori au ales să utilizeze aplicația cu funcția de student. De asemenea, în ceea ce privește timpul petrecut în aplicație și respectiv tipul de exerciții asupra cărora a fost îndreptat focusul, exerciții pentru începători și exerciții pentru utilizatori experimentați, avem o distribuție egală între utilizatori. Utilizatorii au avut acces la ambele tipuri de exerciții, dar în funcție de experiența acestora au ales să petreacă mai mult timp într-o zonă de aplicație sau în alta.

Pe o scară de la 1 la 5, unde 1 deloc util și 5 foarte util, cum evaluați următoarele componente ale aplicației?

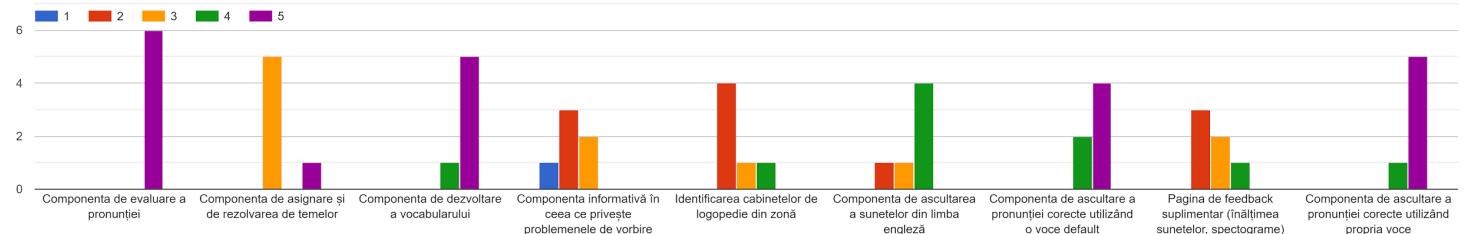


Fig.6.1.12: Grafic care prezintă cât de utile consideră utilizatorii fiecare componentă a aplicației

Pe o scară de la 1 la 5, unde 1 înseamnă foarte ușor și 5 înseamnă foarte dificil, cum evaluați dificultatea exercițiilor?

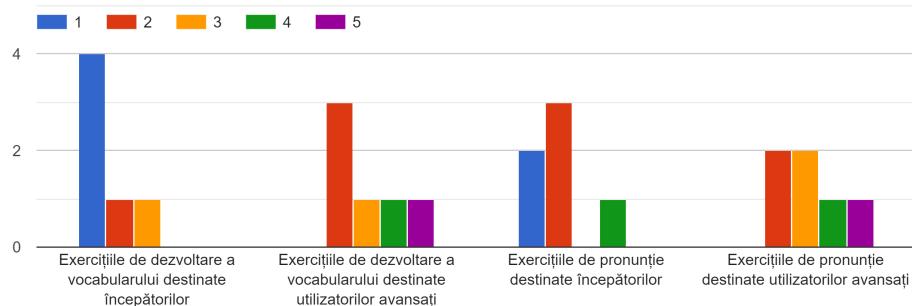


Fig. 6.1.13: Grafic care prezintă cât de dificile au considerat utilizatorii exercițiile de fiecare tip prezentate

Graficele prezentate anterior în Fig.6.1.12 și 6.1.13 ilustrează răspunsurile utilizatorului la întrebările cu privire la funcția de învățare a unei limbi străine. Sunt analizate două aspecte, cât de utilă este fiecare componentă cu care a interacționat utilizatorul, dar și nivelul de dificultate a exercițiilor rezolvate de utilizatori. Din Fig. 6.1.12 putem trage următoarele concluzii: utilizatorii au considerat mai utile componente precum evaluare pronunției, dezvoltarea vocabularului sau ascultarea textului propus ca exercițiu de pronunție, comparativ cu alte componente, cum ar fi, componenta de ascultare a sunetelor din limba engleză sau pagina de feedback suplimentar. Acest lucru poate fi motivat de faptul că aceste componente sunt interactive și antrenează atenția utilizatorului.

Dintre cele două metode de evaluare a pronunției care considerați că oferă răspunsul cel mai util?  
6 răspunsuri

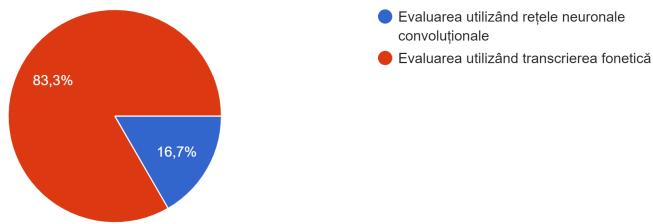


Fig. 6.1.14: Grafic care prezintă care metodă de evaluare a pronunției a fost considerată mai utilă de către utilizator

Tot în ceea ce privește funcția de învățare a unei limbi străine, și având în vedere că lucrarea de față prezintă două metode diferite de evaluare a pronunției, metoda utilizând rețele neuronale convoluționale și metoda care utilizează transcrierea fonetică, utilizatorii au fost întrebați care din cele două metode a fost preferată de aceștia. Răspunsurile la această întrebare sunt prezentate în Fig.6.1.14, din care deducem că utilizatorii au preferat preponderent metoda care utilizează transcrierea fonetică.

Pe o scară de la 1 la 5, unde 1 înseamnă foarte puțin intuitiv și 5 înseamnă foarte intuitiv, cum evaluați următoarele componente ale aplicației?

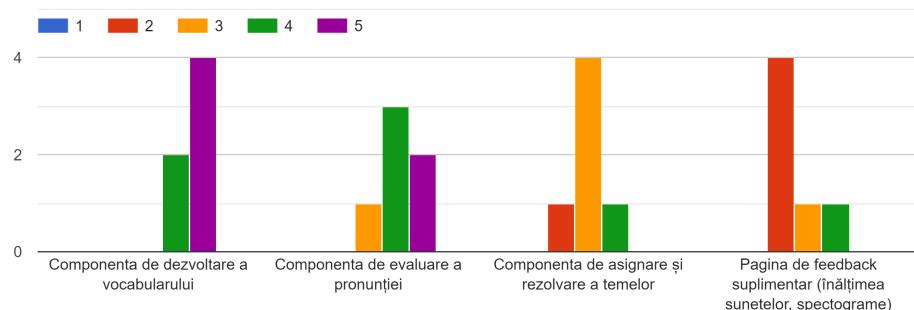


Fig. 6.1.15: Grafic care prezintă cât de intuitive și ușor de utilizat consideră utilizatorii diferite componente ale aplicației.

Pe o scară de la 1 la 5, unde 1 înseamnă nu sunt mulțumit deloc de respectivul aspect și 5 înseamnă sunt foarte mulțumiti cu evaluările următoarele părți ale aplicației?

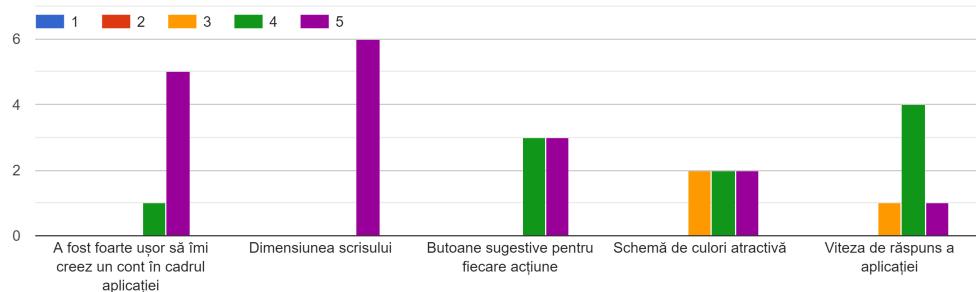


Fig. 6.1.16: Grafic care prezintă cât de mulțumiți sunt utilizatorii de interfața aplicației și de interacțiunea cu aplicația

Cele două grafice prezentate anterior surprind evaluare utilizatorilor în ceea ce privește interacțiunea cu aplicația și interfața grafică a acesteia.

În ceea ce privește experiența utilizatorilor, din Fig.6.1.15 observăm că pentru majoritatea utilizatorilor, modulul de asignare și rezolvare a temelor, dar și modulul de feedback au fost cel mai dificil de utilizat, iar utilizatorii au avut dificultăți în a le accesa, dar și în a interpreta rezultatele returnate de acestea.

Evaluarea interfeței grafice este prezentată în Fig. 6.1.16. Observăm că utilizatorii ai fost în general mulțumiți de aceasta, iar singurele aspecte care pot fi îmbunătățite sunt schema de culori și viteza de răspuns.

## 6.4 Îmbunătățiri

Ce tip de exerciții considerați ați dori să vedeați într-o variantă ulterioară a aplicației încercate?

6 răspunsuri



Fig. 6.1.17: Grafic care prezintă tipul de exerciții pe care ar dori utilizatorii să le vadă într-o versiune viitoare

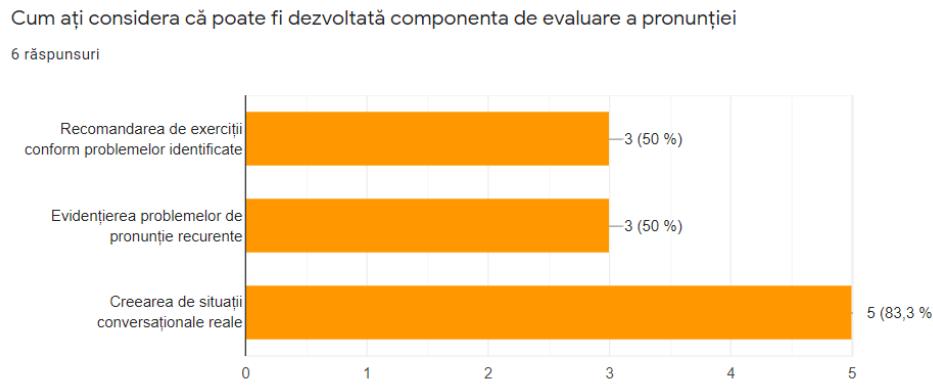


Fig. 6.1.18: Grafic care prezintă funcționalități despre care utilizatorii consideră că ar putea îmbunătăți componenta de evaluare a pronunției

Întrebările din secțiunea de îmbunătățiri urmăresc să surprindă care sunt funcționalitățile pe care utilizatorii sunt interesați să le vadă într-o viitoare versiune a aplicației. Astfel, observăm că există un interes în ceea ce privește realizarea de componente care să contribuie la dezvoltarea altor abilități care fac parte din procesul de învățare a unei limbi străine, pe lângă cele deja prezente în aplicație, cum ar fi exerciții de realizare a unor texte scrise în limba învățată sau dezvoltarea vocabularului printr-o altă metodă decât cea curentă.

De asemenea, în ceea ce privește componenta de evaluare și îmbunătățire a pronunției, observăm că utilizatorii sunt interesați în special de integrarea cunoștințelor dobândite într-o situație conversațională din viața reală.

## 6.5 Concluzii

În timpul realizării experimentului utilizatorii au avut libertate de a explora aplicația în ritmul propriu, condiționat de experiența acestora în ceea ce privește aplicații mobile. De asemenea, persoanele care au testat aplicația au utilizat-o conform așteptărilor proprii legate de o aplicație de învățare a unei limbii străine. Aceste aspecte au fost importante pentru ca experiența în ansamblu să fie cât se poate de apropiată de realitate, iar răspunsurile oferite în timpul completării formularului să surprindă opinia sinceră a acestora cu privire la experiența avută și cum ar putea fi aceasta îmbunătățită.

Drept urmare analizând răspunsurile, putem observa că utilizatorii au fost preponderent mulțumiți de experiență și există un interes în ceea ce privește acest tip de aplicații și în special în ceea ce privește dezvoltarea pronunției cu ajutorul acestora.

## **7. Concluzii finale**

### **7.1. Concluzii**

Urmărind atât opiniile utilizatorilor obținute în urma realizării testului de utilizabilitate, cât și în având în vedere scopul lucrării și problema pe care urmărește să o rezolve putem trage o serie de concluzii.

În primul rând, putem afirma că domeniul învățării unei limbi străine asistată de mobil este un domeniu care merită dezvoltat și integrat în procesul clasic de învățare. Posibilitate de a învăța o limbă nouă în orice moment din zi și în orice locație, fără a fi necesară prezența unei persoane avizate în acest domeniu, condiționată doar de accesul la un dispozitiv mobil, se apropie la momentul actual din ce în ce mai mult de o realitate.

Scopul aplicației realizată în această lucrare este acela de a oferi un exemplu legat de cum se poate realiza o astfel de aplicație și funcționalitățile diverse pe care le poate conține.

Funcționalitatea principală, în jurul căreia gravitează întreaga aplicație este componența de evaluare a pronunției, deoarece este un domeniu mai puțin explorat și pentru care nu există la momentul actual o variantă standard de realizare a unei astfel de evaluări. Lipsa unei metode standard pentru realizare acestei sarcini este unul din motivele principale care au condus la realizarea acestei lucrări în care sunt comparate două metode, transcrierea fonetică și rețelele neuronale convoluționale. Cele două metode au fost inițial descrise teoretic, ca mai apoi să se realizeze implementările amândurora și respectiv o serie de experimente care să pună în evidență avantajele și dezavantajele acestora. Pentru realizare acestui lucru a fost necesar, în cazul amândurora, un pas de obținere și procesare de date utilizate în funcție de cerințele fiecărei. Ulterior aceste metode au fost incluse în cadrul aplicației pentru a putea prezenta modul în care pot fi integrate într-un scenariu real.

Aplicația cuprinde și alte funcționalități pe lângă cele legate de evaluare pronunție, cum ar fi dezvoltarea vocabularului sau asignare și rezolvarea de teme, pentru a exemplifica numeroasele aspecte implicate în procesul de învățare pe care o astfel de aplicație le poate îmbunătăți.

Îmbunătățirile se pot împărți în două categorii: adăugarea de funcționalități noi și îmbunătățirea celor deja existente.

În ceea ce privește adăugarea de funcționalități noi acestea presupun în special dezvoltarea unor alte abilități cu ajutorul aplicației, cum ar fi, realizarea de texte scrise, înțelegerea unor fragmente audio sau dobândirea de informații cu privire la cultura și tradițiile țării de proveniență a limbii învățate. De asemenea, se poate lua în considerare și adăugarea posibilității de a învăța mai multe limbi, nu doar limba engleză, cu ajutorul acestei aplicații.

În ceea ce privește îmbunătățirea funcționalităților deja existente, putem îmbunătăți următoarele aspecte:

- Pentru partea de evaluare a pronunției, pentru ambele metode, o posibilă îmbunătățire constă în îmbogățirea bazei de date cu fragmente audio deja existente, în principal prin colectarea de înregistrări ale unor vorbitori de limba română pronunțând cuvinte în limba engleză. De asemenea, o altă îmbunătățire poate fi colaborarea cu o persoană care are cunoștințe în domeniul lingvistic și poate să grupeze greșelile tipice ale celor care vorbesc limba română în momentul în care vorbesc engleză. Realizând această grupare, se poate realiza un sistem de recomandări astfel încât utilizatorii să primească exerciții focusate pe problemele identificate, în același timp cu ajutorul acestei funcționalități s-ar putea identifica și diverse probleme de vorbire ale utilizatorului.
- O altă îmbunătățire poate fi reprezentată și de adăugare mai multor exerciții atât pentru partea de pronunție, cât și pentru partea de dezvoltare a gramaticii.
- De asemenea, având în vedere că aplicația se adresează și copiilor, iar scopul acesteia este de a face procesul de învățare a unei limbii străine cât mai interactiv, putem lua în considerare integrarea unor jocuri atractive care să mențină atenția și să stimuleze interesul utilizatorului față de acest proces.

## 8. Bibliografie

- [1] Georgiev, Georgieva, Smrikarov (2004), **M-learning - a new stage of e-learning.** In Conference: *Proceedings of the 5th international conference on Computer systems and technologies.* DOI: 10.1145/1050330.1050437
- [2] Steve, O'Hear. (2010) **Babbel introduces speech recognition to aid language learning.** *Babel Techcrunch.* <https://techcrunch.com/2010/06/23/babbel-introduces-speech-recognition-to-aid-language-learning/>.
- [3] Hincks, R. (2003) **Speech technologies for pronunciation feedback and evaluation.** *ReCALL.* 15. 3-20. 10.1017/S0958344003000211.
- [4] Renlog, Ai (2015), **Automatic pronunciation error detection and feedback generation for call applications.** In *International Conference on Learning and Collaboration Technologies LCT 2015: Learning and Collaboration Technologies*, pp 175-186.
- [5] David Berdin (2016), **PARLA: Mobile Application for English Pronunciation. A supervised machine learning approach**
- [6] Keven Chionh, Maoyuan Song, Yue Yin **Application of Convolutional Neural Networks in Accent Identification**
- [7] Aleksandr Diment, Eemi Fagerlund, Adrian Benfield, Tuomas Virtanen (2019) **Detection of Typical Pronunciation Errors in Non-native English Speech Using Convolutional Recurrent Neural Networks**
- [8] Meryam Telmem, Youssef Ghanou (2018) **Amazigh Speech Recognition System Based on CMUSphinx**
- [9] Mark Gales, Steve Young (Vol. 1, No. 3 (2007) ) **The Application of Hidden Markov Models in Speech Recognition**
- [10] Stanford University, Spring 2021, **Convolutional Neural Networks for Visual.** <https://cs231n.github.io/convolutional-networks/>
- [11] Boyang Zhang, Jared Leitner, Sam Thornton, **Audio Recognition using Mel Spectrograms and Convolution Neural Networks**
- [12] Kamalesh Palanisamy, Dipika Singhania, Angela Yao, **Rethinking CNN Models for Audio Classification**
- [13] **Adapting the default acoustic model.** <https://cmusphinx.github.io/wiki/tutorialadapt/>
- [14] **Machine Learning pentru Aplicații Vizuale**  
[http://www.master-tajd.ro/Cursuri/MLAV\\_files/Retele%20Convolutionale%20-%20Convolutional%20Neural%20Networks%20\(CNNs\).html](http://www.master-tajd.ro/Cursuri/MLAV_files/Retele%20Convolutionale%20-%20Convolutional%20Neural%20Networks%20(CNNs).html)
- [15] Leon Mak An Sheng, Mok Wei Xiong Edmund, **Deep Learning Approach to Accent Classification**
- [16] **Audio classification using CNN**  
<https://medium.com/x8-the-ai-community/audio-classification-using-cnn-coding-example-f9cbd272269e>
- [17] **Real-Time Voice Cloning**  
<https://github.com/CorentinJ/Real-Time-Voice-Cloning>