Deep Learning for Speaker Characterization

Degree Thesis
submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by

Daniel Garriga Artieda

In partial fulfillment
of the requirements for the degree in
Bachelor's degree in Telecommunications Technologies and Services Engineering

Advisor: Dr Francisco Javier Hernando Pericas
Barcelona, Date 19/06/2022

Contents

List of Figures

List of Tables

Abbreviations

AI Artificial Intelligence

ASR Automatic Speech Recognition

CE Cross Entropy

CNN Convolutional Neural Network

DNN Deep Neural Network

FC Fully Connected

LSTM Long-Short Term Memory

MLP Multilayer Perceptron

ordinalreg Ordinal Regression Classification

ReLU Rectified Linear Unit

seq2seq Sequence to Sequence

STFT Short Time Fourier Transform

TALP Tecnologies i Aplicacions del Llenguatge i la Parla

TSC Departament de Teoria del Senyal i Comunicacions

UPC Universitat Politècnica de Catalunya

VGG3L Visual Geometry Group 3 Layers

VGG4L Visual Geometry Group 4 Layers

WCE Weighted Cross Entropy

Abstract

Speech characterization is one of the most relevant tasks in a lot of voice-related artificial intelligence applications. As these technologies thrive and the amount of data available increases, it is also salient their adaptation to different languages and contexts. In this project, a network to classify the gender, age and accent of a Catalan speaker through their voice is proposed. Different variations of the main model's blocks are going to be explored, including some innovative techniques as the Double Multi-Head Attention pooling mechanism. In addition, some data analysis- including the application of some state-of-art voice data augmentation techniques- will be done aiming for better results. Some results show strong promise, as they indicate improvement in comparison to some classical methods not based on machine learning.

Resum

La caracterització d'un locutor és una de les tasques més rellevants en moltes aplicacions d'intel·ligència artificial. Tan bon punt s'afinen aquestes tecnologies i augmenta la quantitat de dades disponibles, és important adaptar-les a diferents idiomes i contexts. En aquest projecte, es proposa una xarxa neuronal per classificar el gènere, l'edat i l'accent d'un parlant en català. Vàries variacions dels blocs més convencionals seran explorades, incloent algunes tècniques punteres com un mecanisme de pooling basat en *Double Multi-Head Attention*. Per altra banda, es durà a terme anàlisis de dades per tal de millorar els resultats obtinguts, incloent l'aplicació de tècniques estat de l'art d'augment de dades. Alguns dels resultats són força prometedors, al demostrar una considerable millora respecte altres tècniques clàssiques no basades en l'aprenentatge automàtic.

Resumen

La caracterización de un locutor es una de las tareas más relevantes en muchas aplicaciones de inteligencia artificial. Del mismo modo que estas tecnologías mejoran y se aumenta la cantidad de datos disponibles, es también importante adaptarlas a distintos idiomas y contextos. En este proyecto, se propone una red neuronal para clasificar el género, edad y acento de un locutor en catalán. Distintas variaciones serán exploradas, incluyendo algunas técnicas punteras como el mecanismo de pooling basado en *Double Multi-Head Attention*. Por otro lado, se llevará a cabo análisis de datos con tal de obtener los mejores resultados posibles, incluyendo la aplicación de técnicas nivel estado del arte de aumento de datos. Algunos de los resultados son prometedores, al demostrar una considerable mejora respecto otras técnicas clásicas no basadas en aprendizaje automático.

Acknowledgements

Firstly, I would like to start this report acknowledging the support given by my advisor Javier Hernando. Having an expert like him daily following the project, with his valuable recommendations and opinions, was key for its development. He gave me the opportunity to work on this project, which I am really grateful for.

Secondly, I would also like to acknowledge the support given by professor Ignasi Esquerra, who was also part of the project. Having a second expert's opinion in our meetings was also greatly appreciated.

Lastly, I would like to notice that I have been working on this project in cooperation with Daniel Navarrete, a UPC Master student who since the very beginning helped me to set everything I needed up for the project development. Having a more veteran presence like him by my side also had a really positive impact.

Thank you all, I really appreciate your trust and help.

Revision history and approval record

Revision	Date	Purpose
0	28/05/2022	Document creation
1	19/06/2022	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Daniel Garriga Artieda	daniel.garriga.artieda@estudiantat.upc.edu
Francisco Javier Hernando Pericas	javier.hernando@upc.edu

Written by: Daniel Garriga Artieda		Reviewed and approved by: Francisco Javier Hernando Pericas	
Date	28/05/2022	Date	19/06/2022
Name	Daniel Garriga	Name	Javier Hernando
Position	Project Author	Position	Project Supervisor

1 Introduction

In this introduction, the reader will be given some context on the project- starting with the idea, motivation and importance of it- and continuing with the procedures and organization followed, explaining the objectives aimed in between.

1.1 Motivation and relevance

If somebody asks a child which is the key factor that differentiates humans from animals, the answer would probably be the way we feel, think and communicate. As we were taught in school, communication can take place in a wide variety of ways: personal/impersonal, verbal/non-verbal,... The way we communicate usually indicates how we are and even where we come from.

As technology improves at an abysmal rate, if we take a look at our society, we can clearly see a trend towards impersonal/non face-to-face communication; indeed, COVID has truly emphasised this movement. We are changing conversations for phone messages and in-person meetings for telematic ones. In fact, as it is surely common now, most of this thesis' meetings have been done online.

However, there is one element from classical communication that has maintained- if not raised- its importance over time: our voice. The examples of how important of a role voice plays in communication using technological devices, just go on and on: there are about 7 billion WhatsApp voice messages sent around the world every single day, we can now ask our car to show us the directions to our destination and just image how little sense would it make to have an online meeting without hearing each other.

The amount of technological applications based on our voice, created and developed in recent years, is almost overwhelming. In particular, speaker characterization applications- as the one developed in this project- have gained a lot of importance lately in a wide variety of fields such as security, marketing and even medicine or forensics.

Nevertheless, one key- and often forgotten- factor is accessibility. One of the main characteristic that our voice has associated is its language. It would not be practical or useful to develop a voice application only in one language, as its scope might be severely reduced. Some basic applications and models can maybe be directly translated, but more complex ones- specially related with Machine Learning, where the Neural Network is not able to learn many dialects at once- need specific implementations for each language.

Keeping accessibility in mind, this project aims to create and adapt speaker characterization algorithms to one particular language: Catalan. Next chapter will develop on how new features have helped the investigation on this topics in Catalunya and in our department, while given some background information of them.

1.2 Project background

This project takes place with the context of the Catalan Common Voice database creation, powered by Mozilla. This database was created in 2018, but has seen a huge expanding thanks to the promotion it has received in the last year and the voice donations of the Catalan-speaking population, so there are now new releases every three months.

On a more global note, the creation of this Common Voice Catalan database and the need to adapt artificial intelligence technologies to the Catalan language, lead to the creation of the AINA program. This project- promoted by the *Polítiques Digitals i Administració Pública* department of the Catalan government and the Barcelona Supercomputing Center- aims to shape all these speech recognition techniques to Catalan [1]. This project aspires to make a small contribution to the AINA program and consequently to the modernization of the Catalan language.

This database and the whole project is relatively new, but some work has already been done, as UPC students Santiago Escuder and Miquel Angel India have already worked on this project and developed some of the algorithms needed [2][3]. Indeed, this project will serve as a continuation and application of these models. Although the department and these students have been working in this project and the topics related to it for quite some time, there are a lot of improvements to be made and a lot of goals to be accomplished. It is for that reasoning that the department was looking for students to continue doing research on this topic, as well as developing an improved classification with better results using the newest version of the database and possibly new approaches.

1.3 Objectives

The main goal of this project is to classify (and characterize) a given speaker by their gender, age (differentiating between classes separated by ten years) and which Catalan accent/dialect do they speak. To do so, there some key focal points that need to be tackled during its development:

1. To create an algorithm using supervised Deep Convolusional Neural Networks with state-of-art results.
2. To implement self-attention techniques for better results.
3. To improve past work done by the department with the newest version of the Common Voice dataset.
4. To do data analysis for better income training data, explore data augmentation techinques.

1.4 Requirements and specifications

Every successful project needs to translate its objectives into defined requirements and specifications.

The main project requirements are:

- Robust gender, age and dialect classification
- The final product needs to improve the results from previous similar projects
- Optimized input database

With the specifications for the requirements above being:

Requirement	Specifications
- Robust gender, age and dialect classification - The product will have improved results from previous similar projects	- Improvement on the Confusion matrices - Improvement on the Accuracy and F-Score measurements from previous systems by 5%/0.05
- Optimized input database	- The following dataset labels must not exceed this values for a precise training: - Gender: <65% men, >35% women - Age: >1% teens, >13% 20s, >15% 30s, >15% 40s, <25% 50s and 60s, >10% 70s

1.5 Project structure

In order to properly develop the project and complete its objectives, there have been defined four basic work-packages. Each work-package has its own internal tasks, which have some deadlines and milestones associated.

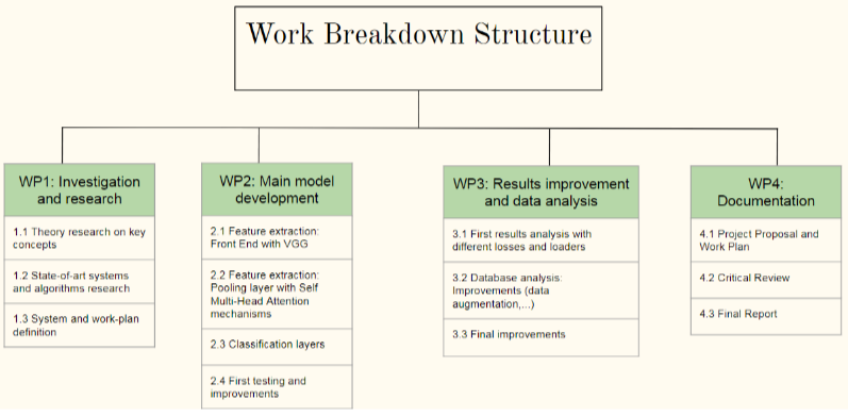


Figure 1: Work packages and their internal tasks

A more detailed description of each work-package can be found in Appendix **Work-Packages**.

All of these work-packages can be summarized in a Gantt diagram. This diagram was created at the start of the project, and has been slightly modified during the course of it. As seen, some of the tasks have been developed at the same time as some other ones. On the other hand, some of them -as the final improvements with data augmentation- were done once the whole model was established:

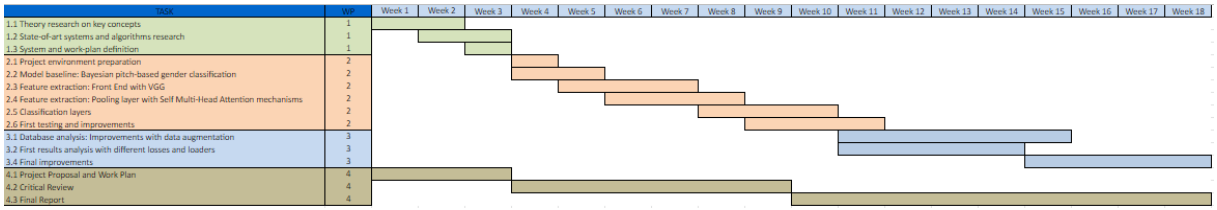


Figure 2: Project's Gantt diagram

2 State of the art

As seen on the project organization chapter, the first step taken in this project consisted in doing research on state-of-art techniques and other similar projects. Thanks to the popularization of Deep Learning in this type of tasks, there is a large variety of models and approaches related to them.

In this chapter, a general overview of the current work being done on gender, age and accent classification is going to be given, getting into detail on the different models and mechanisms and their associated results. First, some specific similar projects are going to be discussed, making extra emphasis on the different approaches and the range of results expected. Following that, a review on some specific aspects of the system- such as the different losses and loaders, the feature extraction method and some possible data augmentation techniques- is being done in order to decide which techniques could be interesting to try in our model.

2.1 Gender, age and accent classification using Deep Learning

Gender, age and even accent/dialect voice classification are some of the most commonly-tackled problems by voice specialists. The robustness to a variance in the speaker's population segment has been proven to be key to the success of ASR systems [4], currently used in a wide variety of applications. Depending on the scope of the project, each of these tasks can be approached differently from one to the others (regression/classification, different layers,...), as it is also relatively common to train the model on a multi-task approach (usually combining age or accent classification with gender).

Gender

For our gender classification task, we are constraining it to the biological gender, as it is the one which has influence on the speaker's voice. Although its complexity, this task is considered to be the most trivial one out of the three, as the network heavily relies on the speaker's pitch information. As humans, it is easy to understand why that is the case, as it might also be way more challenging to identify someone's age from their voice rather than their gender. Nevertheless, multiple investigations have tried to fully optimize their system's performance, in order to accomplish almost perfect results.

In this task, the complexity of the used model often has an impact on how the system performs. Some basic models, nonetheless, still perform with improved results compared to the other tasks ones- in fact, as seen in next chapters, a simple not Machine Learning-based Naive Bayes classifier already gets decent results. However, some special techniques may lead to fully optimized performances.

Some state-of-art result metrics will fluctuate between a 95-99.5% accuracy and 0.94-0.985 f-score. We can see this range of results through different approaches and models, such as LSTM networks, a MLP or a similar one (with attention mechanisms) as the one used on this project [*see* [5], [6] *and* [7]].

Age

While, as discussed in the previous section, two-classes gender classification usually achieves >0.9/0.95+ f-score measures, age classification can often be way more challenging due to the similarity between frequency components of two classes [8][9]. Going back at our human point of view, it is almost impossible for us to distinguish between the voices of someone in their fourties and someone in their fifties. If we could compare the spectrograms of someone saying the same sentence with a ten-year span, we could certainly see that they are almost identical.

There are some studies that also reveal how some characteristics- that usually make the classification more difficult- slowly appear in our voice as we get older. This factors go from lower speaking rate to some extra jitter [10][11], and add up to the age classification complexity.

Sometimes, the classification proceeds from regression. A regression, in this task, is often more convenient as it is more accurate to give an approximate age to the speaker rather than classifying them to a group of age ([12] is an example of directly using a regression layer). For instance, the difference between a thirteen year old's voice and an eighteen year one might result much more noticeable than the one between two fourty-five and fifty year old, despite the first two being in the same class and the second ones in separate ones in a classification.

As mentioned in this chapter's introduction, models based on multi-task learning with age and gender have also proven to be efficient, offering an f-score of between 0.6 and 0.8 [7].

Accent

Like the age task, accent classification could also represent a great challenge depending on the case. It is worth noticing the difference between accent and dialect; while different dialects might use specific words from their specific vocabulary, different accents only change the way the speakers pronounce the same words. In this project's case, it could be said that we are working with dialects, but, as we are no focusing on speech recognition tasks, we can use the term accent and assume that the different speakers are talking the exact same language.

Identifying the speaker's accent is also a salient task in today's voice analysis tasks. ASR relies a lot in trying to understand a particular language, so knowing the speaker's accent/dialect is mandatory. Indeed, a lot of this ASR sytems use the speaker's accent as another input to the network [13].

In this task, there are two important factors that have direct influence on the training success: how many classes/labels are we classifying and how different are them from one to another (prosody: intonation, rhythm,...) . Going back to the human analogy, if we tried to classify a speaker's accent between British or North-American, we would probably guess it most of the times. On the other hand, if we wanted to classify someone's accent by which USA state they were born, we would probably not get get it right.

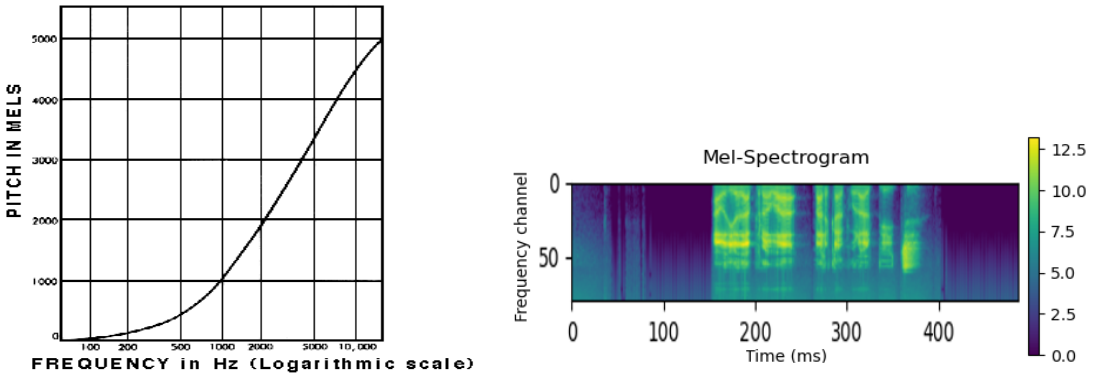
State of the art results can reach up to a 0.88 f-score for a three-classes case, but usually sees a severe drop-off as more classes are added to the classification [14][15].

2.2 Feature Extraction

Feature extraction is the first key aspect a classification system has to face. Finding the correct set of voice features, that will help our model correctly characterize the speaker, is a salient issue. Currently, there is a large variety of feature extraction methods and algorithms. For speaker's voice classification, spectral features are the most commonly used, including the Mel Frequency Cepstral Coefficients/Spectrogram (MFCC), the Linear Prediction Coefficients (LPC), Line Spectral Frequencies (LSF), the Discrete Wavelet Transform (DWT) and the Perceptual Linear Prediction (PLP). Indeed, the most used of them is the Mel/Log-Mel Spectrogram.

Mel-Spectrogram

Spectrograms are one of the most useful ways to represent a speech signal, as it shows its frequency components' evolution through time. The spectrogram is computed applying a STFT to a discrete signal, and shows in a variable darker tone the power of the transform depending on time (x-axis) and frecuency (y-axis). One way to adapt the frequencies' scale to a more human perception-oriented one is using the Mel scale. As humans, the measure related to how we perceive a sound is called the pitch, and it does not linearly correspond to the sound's frequency. To solve this problem, the Mel scale transforms the frequency range of the STFT/Spectrogram in Hertz into a new one [16].



(a) Pitch/Frequency relation

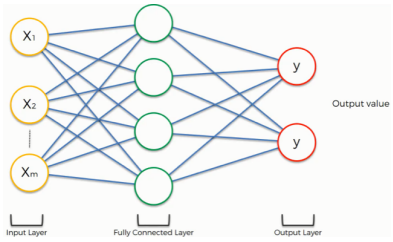
(b) Mel-Spectrogram example

Figure 3

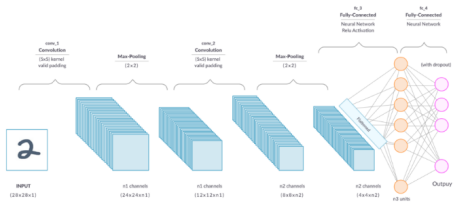
2.3 Deep Learning classifiers

Complexity of speech classification models can deeply change depending on the task at hand. While a simple model that computes some output values for a given input could be enough for basic classification tasks, it is worth taking a closer look to more advanced approaches that will lead to better results. Indeed, most state-of-art systems use a complete

CNN, which has at least a feature-learning block- composed by a CNN- and some fully-connected layers to learn and output the results (usually with a SoftMax layer). Other optional blocks- such as pre-processing or attention mechanisms ones- are often included to improve the system's performance.



(a) Basic classification model with only a FC layer



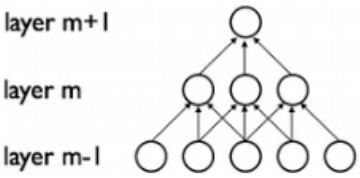
(b) Complete CNN model

Figure 4

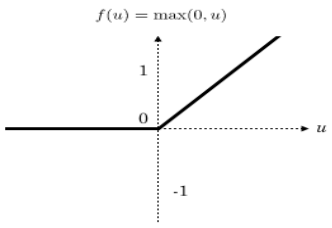
In this section- as well as in next one- some of the most used blocks in classification tasks will be detailed, while giving examples of state of art implementations.

Front End Block

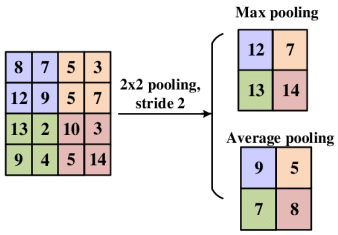
This Front End/Convolutional layer's function is to learn and extract high-level features from the input data, as well as reducing the dimensions of it using pooling layers. In this layer, some filters- also called kernels- are used to detect the main features of the input image. This convolution layers are also followed by a ReLu activation function [17][18].



(a) Basic Convolution example



(b) ReLu function



(c) Pooling layer

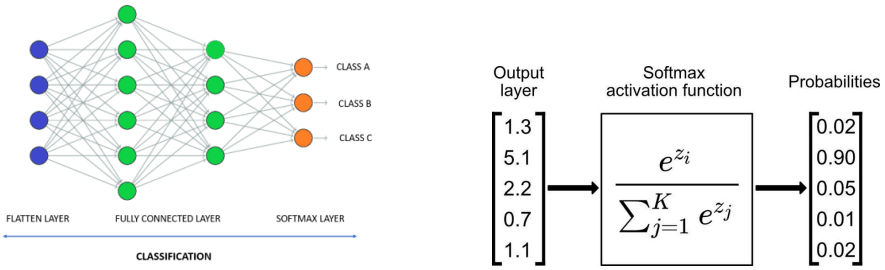
Figure 5

To emphasize the most important features, an attention-based pooling layer often follows this layer. This type of mechanisms will be explained in the next chapter, as it is not mandatory to make use of them in basic CNN models.

Fully-Connected Layers

As features are already extracted, our network has now to decide and classify which label better represents the output of that Front End layer. This task is done by the FC layer. This network will learn in order to adapt its weights and biases aiming to do the best classification.

As mentioned before, this layer's input- a one-dimensional sample vector- comes from the last Front-End/Convolutional layer (or the attention-based pooling one if used): the flattened layer. This vector will then go through as many fully-connected layers as we decide. Lastly, the output data from these FC layers will then go trough a SoftMax layer, which will compute the probability distribution for each class.



(a) Fully-Connected Layer scheme (b) Soft-Max Layer

Figure 6

2.4 Attention mechanisms

Between the two layers in a basic CNN system mentioned above, it is common to include an attention-based pooling mechanism. This new layer's goal- firstly used in seq2seq models [19][20]- is to emphasize the most salient high-level features of the input feature vector. To do so, a weight- calculated through the encoder and decoder sequences and representing how well the query and target match- is given to each input sequence of the encoder.

This attention-based pooling mechanisms usually result in better performances than statistical ones, like the one shown in 5c, as these basic ones do not take into account that elements of the sequence do not contribute equally in the classification process. In speech recognition/classification models, consequently, the addition of attention mechanisms also plays an important role in emphasising the most relevant features. For instance, if we are doing a gender classification, the attention layer would probably emphasize those features which contain pitch information- which will probably be the ones that make the most impact on the classification.

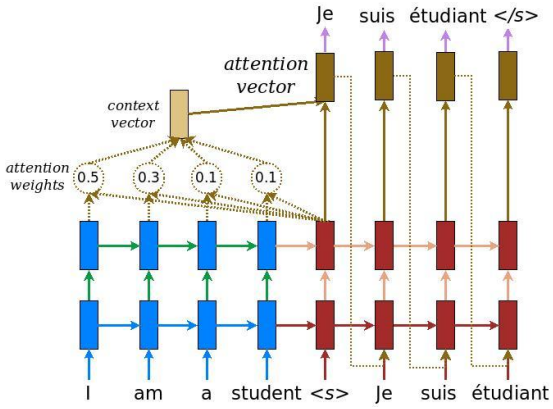
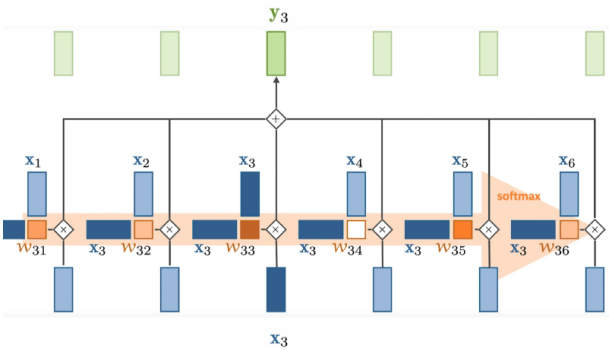


Figure 7: Language Seq2seq Encoder-decoder Attention example

Multiple approaches to the *attention* concept can be taken:

Self-Attention

Self-attention mechanisms, in contrast to basic attention mechanisms, are self-referential. This means that weights are computed only using the input states of the encoder. This type of mechanisms have proven to be successful in a large variety of tasks going from reading comprehension [21] to ASR [22].



(a) Self-Attention global scheme

$$y_i = \sum_j w_{i_j} x_j$$
$$\hat{w_{i_j}} = x_i^T x_j$$
$$w_{i_j} = \frac{exp\hat{w_{i_j}}}{\sum_j exp\hat{w_{i_j}}}$$

(b) Self-Attention mathematical expressions

Figure 8

Self Multi-Head Attention

A recent implementation of this method on Speaker Verification was developed by UPC's TALP Department [3], which demonstrated that it leads to positive results when implemented on these type of tasks.

This approach consists in dividing the encoded sequence (with dimension D) into smaller sub-sequences (*heads*, with dimension 1xD/k, with k being the number of heads). When we have this heads divided, we have to compute their weights with trainable parameters $u_j \in R^{\frac{d}{k}}$ with the same method as the self-attention ones:

$$w_{t_j} = \frac{exp(h_{t_j} \intercal u_j)}{\sum_t^N exp(h_{t_j} \intercal u_j)}$$
$$c_j = \sum_T^N h_{t_j} \intercal w_{t_j}$$

Figure 9: Multi-Head Attention Formulas

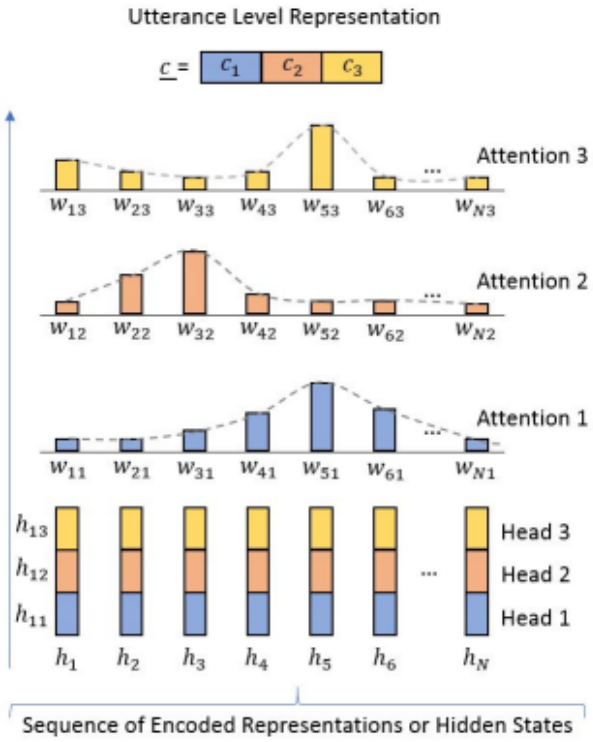


Figure 10: Multi-Head Attention global scheme

Another approach stem from the Self Multi-Head Attention one is the **Double Multi-Head Attention** one. In this approach, also developed by the TALP department, a self-attention layer is concatenated to a multi-head attention one [23]. With this method, the speakers embeddings obtained are even more discriminating, as well as giving more importance to the most valuable heads (with Multi-Head Attention any discrimination between heads was done). For that task, an improvement of 6.09% and 5.23% in terms of EER compared to Self Attention pooling and Self Multi-Head Attention, respectively can be found

2.5 Losses and Loaders

Losses

When working with DNN/CNN one of the key aspects that will have a direct influence to our training success is the loss function we choose. Indeed, this function is one of the key elements of the back-propagation training step.

First, the network computes an scalar loss value comparing the output/predicted class with the original/true one using the loss function. Then it adjusts its parameters (weights and biases) recursively iterating through them and finding the set which results in a minimum loss (gradient descent).

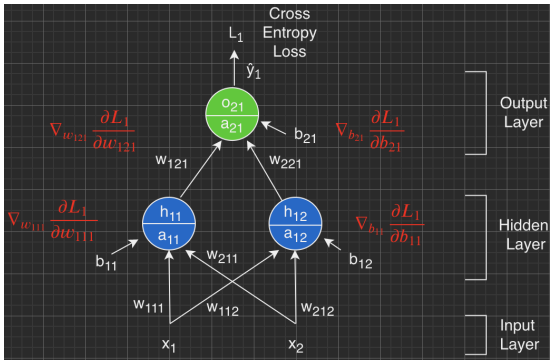


Figure 11: Basic network loss diagram [24]

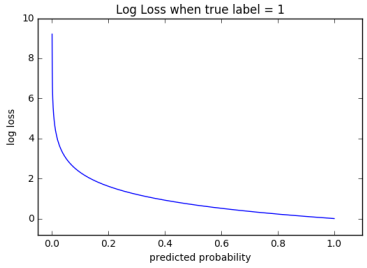
Here is a list of the most used lost functions:

- **Cross-entropy**:

  With cross-entropy, we are using the parallelisms with information and entropy to evaluate the probability of the system giving as an output the correct class. When the network has a probability of resulting in the correct class near 1, the loss tends to be 0. On the other hand, when that probability is low (near 0), the loss exponentially rises:

$$\mathbf{CE}(y, \hat{y}) = -\sum_{i=1}^{M} y_i \log(\hat{y}_i)$$

(a) Cross entropy loss function



(b) Lost-Output probability dependence

Figure 12

- **Weighted Cross-Entropy**:

  This lost function is based on the same principles as the simple cross-entropy one. The difference comes from the weights added to each of the values entropy-calculation to help to reduce the negative effects of imbalanced sets:

$$\mathbf{WCE}(y, \hat{y}) = -\sum_{i=1}^{M} w_i y_i \log(\hat{y}_i)$$

Figure 13: Weighted Cross-Entropy loss function

- **Mean-Squared Error**:

  This loss function, one of the most basic but also effective in some cases, is usually used in regressions. It simply computes the difference between the predicted and the original label. As the difference increases, the loss exponentially does so too.

$$\mathbf{MSE}(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{D}(\hat{y}_i - y_i)^2$$

Figure 14: Mean Square Error loss function

- **Kullback-Leibler Divergence Loss**:

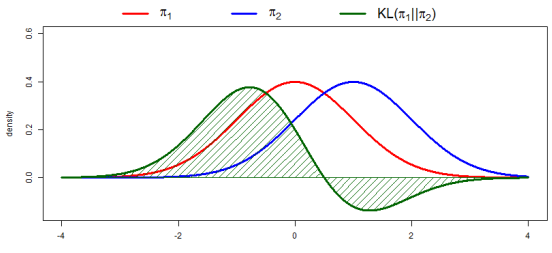
  This loss function, probably the most sophisticated one on the list, aims to represent the distance between two probability distributions.

  When working on classification models, a Softmax layer is usually used at the network. This layer, gives us the output distribution for all classes, which we can compare to the input distribution.

  This loss function is also used in regressions, where the output distribution can also be compared to the output.

$$\mathbf{KLD}(\hat{y}||y) = \sum_{c=1}^{M} \hat{y}_c \log \frac{\hat{y}_c}{y_c}$$

(a) Kullback-Leibler Divergence loss function



(b) Kullback-Leibler Divergence between two normal distributions[25]

Figure 15

Loaders

The loader of our system corresponds to the way we introduce data when training it. The basic approach consists in dividing our input dataset into small packages called *batches* or *mini-batches*.

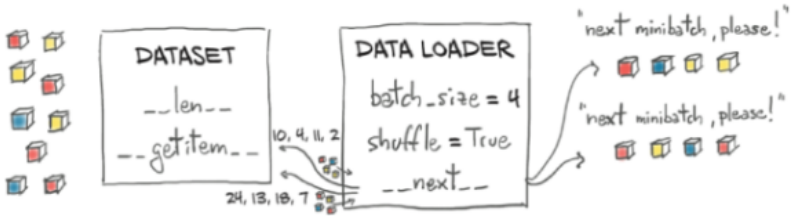


Figure 16: Data Loader scheme [26]

Depending on the way we input these packages, we can differentiate two loaders [2]:

- **Random Loader**

  The data is loaded randomly with no special distinction between classes. Each data sample is used one time and the epoch (training cycle) is declared as finished when the system has gone through all of them.

- **Weighted Random Sampler**

  The functionality is similar to the previous one but, in this case, the data sample can be loaded more than once in order to compensate imbalance between classes. Those classes with less sample are designated with a higher weight so they have a higher chance to be loaded.

2.6 Data Augmentation techniques

In order to increase the generalizability of a data model, data augmentation is one of the most used techniques in supervised/unsupervised learning [27]. The main idea is to create replicas from the already existing training data, modifying some of that original's data characteristics. This is particularly interesting to avoid overfitting, which usually appears as a consequence of imbalanced classes with not enough samples. Overfitting occurs when our network is not trained with enough data and specifically learns each sample's case, consequently not being able to properly classify new data from the same class.

Depending on our application, the techniques applied to create that artificial data might change. For speech classification, these data augmentation techniques need to be to applied directly to the voice sample or the voice spectrogram- which will later have an impact in our model's learning.

In order to apply these data augmentation techniques, it is really important that we apply it to all the classes- as if we apply it only to the ones with few samples, the network might end up classifying between augmented/not-augmented instead of their original classes. A study of the quantity of data to be augmented for each class- as well as understanding

that the new synthetic data needs to be different enough from the original one, but also being recognizable to the system- is mandatory in order to have successful results.

Some of the most common voice/speech data augmentation techniques are:

- **Noise injection**

  This method is one of the most simple. It consists in adding some random noise- the randomness is a key factor, as adding always the same noise value would not allow to use the same data multiple times- to the original audio or to the spectrogram. The noise characteristics could change depending on the environment the system is asked to perform on. For instance, in [28] a signal simulating the noise outside of a car is used in the augmentation process.

- **Time shifting and time stretching**

  This two transformations, applied in the time domain, consist in shifting the audio forward or backwards and stretching it so it has a longer or shorter duration.

- **Vocal Tract Length Perturbation**

  This technique, proposed in [29], is a similar technique as using the Mel-scale. It is based on changing the frequency scale with a *wrap factor*, which consequently changes the speaker's pitch. It differentiates from the time stretching method as it does not change the track's duration-

$$f' = \begin{cases} f\alpha & f \leq F_{hi}\frac{\min(\alpha,1)}{\alpha} \\ S/2 - \frac{S/2-F_{hi}\min(\alpha,1)}{S/2-F_{hi}\frac{\min(\alpha,1)}{\alpha}}(S/2-f) & \text{otherwise} \end{cases}$$

Figure 17: Vocal Track Length Perturbation formula

- **Time and Frequency masking**

  This technique- described in [30]- works either in the time domain, the frequency domain or in both. In the track's spectrogram, we apply a mask to a random number of channels (in the case of frequency masking) or on a random length of time samples (in the case of time masking). Indeed, this method offers the best results out of all the mentioned above.

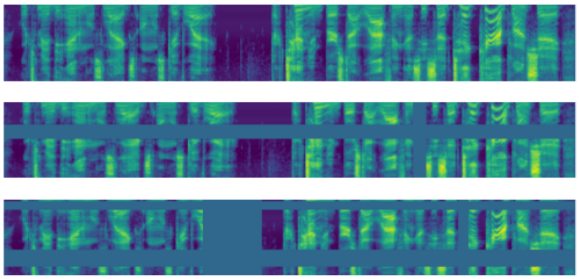


Figure 18: Time, frequency and time+frequency masking [30]

3 Project Development

In this chapter, the methodology followed through the project is going to be described. First, the dataset used with its distribution and features are going to be detailed. Secondly, the chosen model is going to be explained, going into detail with the different blocks and layers used. Lastly, a list of the software used- including the data augmentation techniques chosen- will be provided.

3.1 Common Voice Dataset

As explained in section 1.2, the improvement of the Catalan Common Voice dataset is one of the main motivations of this project. Currently, the last version of the dataset is v8.0, which was released in January, 2022. This dataset has been created thanks to the voice donation of the Catalan-speaking population. When going into the Common Voice website, one can either donate their voice- reading some short text- or validate other audios. This second step is actually even more important than the first one, as only validated samples are being used for this project. In this last release, 607.601 validated audios are available, almost 80.000 more than last year's one.

Note: Some audios do not have all the classes' information, for instance, some might have its gender and age but no its accent, so the total number of audios from one class may not add up to 607.601.

The audios on the dataset are divided in the following tasks and classes:

Gender

For the gender classification, three classes are considered: *Male*, *Female* and *Other*. These classes have the following statistics:

Class	Samples	Percentage
Male	385.061	76,5%
Female	117.666	23,4%
Other	418	0.1%

Table 1: Gender Sample Distribution
(additional graphic can be found at Appendix **Common Voice Dataset Distribution**)

In this task, a significant imbalance towards the *Male* class can be observed. In addition, the *Other* class has not been used for the sake of this project, as it would be nearly impossible to train the model with such few samples.

Age

For the age task, eight classes- each class representing people from the same age with a ten year span, ranging from *teens* to *seventies*- are considered. This *teens* class, however, due to minor privacy conditions, only has voice samples from eighteen and nineteen year's old. The distribution is the following:

Class	Samples	Percentage
Teens	3.946	0,01%
Twenties	40.724	8,0%
Thirties	66.860	13,1%
Fourties	78.070	15,4%
Fifties	131.579	25,9%
Sixties	165.861	32,6%
Seventies	21.375	4,2%
Eighties	50	0.00%

Table 2: Age Sample Distribution
(additional graphic can be found at Appendix **Common Voice Dataset Distribution**)

As shown in the tables above, the *teens* and *eighties* classes have not a lot of audio samples. Depending on the experiment, they might be considered as *twenties* and *seventies*, respectively, or completely dismissed. This consideration, nonetheless, will be specified in each experiment's details in 4.3.

Accent

For the accents task, the dataset distinguishes between all the dialects spread through the Catalan-speaking territory. Despite having even more detailed information in some of the classes- such as the *aprenent (recent, desde altres llengües)*, which indicates that the speaker recently started learning Catalan- as these classes have quite limited samples, they will not be taken into account as explained in the 4.4 section. An extensive imbalance towards the *central* class can also be seen, which needs to be addressed in order to have a proper training. The classes' distribution is the following:

Class	Samples	Percentage
Central	431.166	88,04%
Valencià	28.432	5,81%
Nord-occidental	18.971	3,87%
Septentrional	5.556	1,13%
Balear	5.089	1,04%
Septentrional-central	188	0,04%
Central-Girona	188	0,04%
Aprenent-castellà	122	0,02%
Balear-mallorquí	81	0,02%
Tortosí	6	0,00%
Aprenent-altres	5	0,00%

Table 3: Accents Sample Distribution
(additional graphic can be found at Appendix **Common Voice Dataset Distribution**)

3.2 Used Model

In this section, the used model and its characteristics is going to be detailed. This model was partially inherited from [2] and [23], as the author of these projects and this one's tutor generously gave me access to it.

The model is composed by three main blocks: the Front End block, an attention-based Pooling layer and classification Fully-Connected layers (with a Softmax following them). As explained in sections 2.3 and 2.4, the first will extract the input's features, the second is going to emphasize the most important ones and concatenate the matrices (reducing its dimensions) and the last one will make the classification.

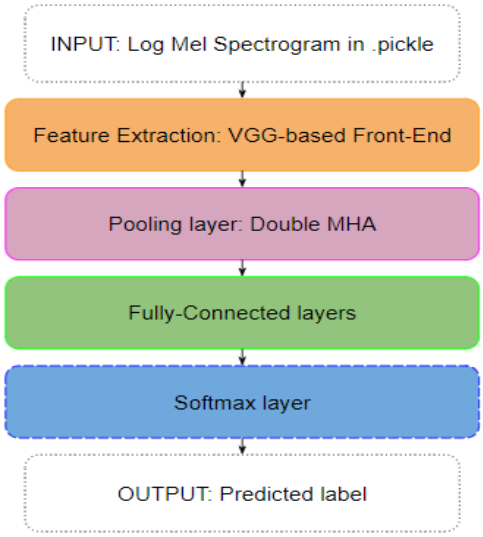


Figure 19: Used Model Architecture

In next sections, each of the main blocks are going to be detailed, going through their architectures and most relevant characteristics.

Input format: Log Mel Spectrogram in .pickle

As explained in section 2.2, Mel/Log-Mel spectrograms are usually the best input format for speech classification. Therefore, it was decided to use Log-Mel spectrograms as the input information of the model. In this case, 80 mel-channels and N frames are going to be used, so the input data will have a size of 80xN.

All this information will be handled to the system using the .pickle format. A pickle, is the product of converting Python objects into byte streams. This Python objects are serialized/de-serialized, in order to be processed in an effective way by multiple Python files [31].

Front End Layers

Once all the information is handled to the system, it first goes trough the Front End/Feature Extractor layers. These Front End layers, are adapted from the one described in [3]. In this model's case, while some of the experiments were also done using the VGG3L architecture as a basis, it was mainly adapted to a VGG4L one. Moreover, to try to avoid overfitting, VGG2L has been also adapted and unsuccessfully tested. This architectures, based on [32], has proven to outperform previous models on the image processing field.

In each case (VGG3L/VGG4L), each block is composed by two convolutional layers and a MaxPooling one with a 2x2 stride. Consequently, for every layer that our CNN system has, the number of frames will be divided by 2 (at the end of the block, N=$\frac{1}{2^3}$ for VGG3L and N=$\frac{1}{2^4}$ for VGG4L).

The output of this block, will be a $(64 * 2^3, \frac{80}{2^3}, \frac{N}{2^3})$ matrix for the VGG3L case and a $(64 * 2^4, \frac{80}{2^4}, \frac{N}{2^4})$ matrix for the VGG4L one.

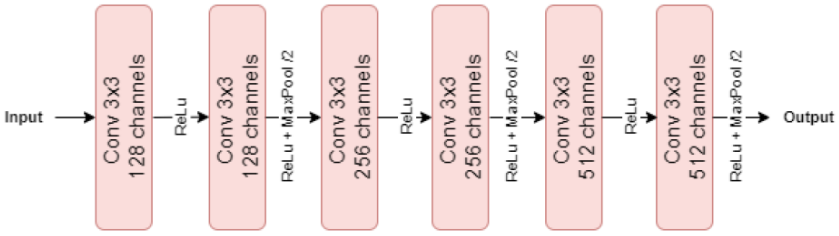


Figure 20: VGG3L Architecture

Pooling Layer with Attention mechanisms

For this next pooling layer, the three main state-of-art attention mechanisms described at 2.4 (self-attention, self multi-head attention and double multi-head attention) have been implemented. Basic temporal/statistical ones have been dismissed as they have been proven to offer worse results [23] for this layer (still used in the VGG ones).

The first step in every pooling mechanism consists in a flattening stage where all the channels (512 or 1024 depending of the VGG model chosen) and one of the other dimensions (the mel-scale frequency one, now decimated to 10 or 5) are concatenated in one.

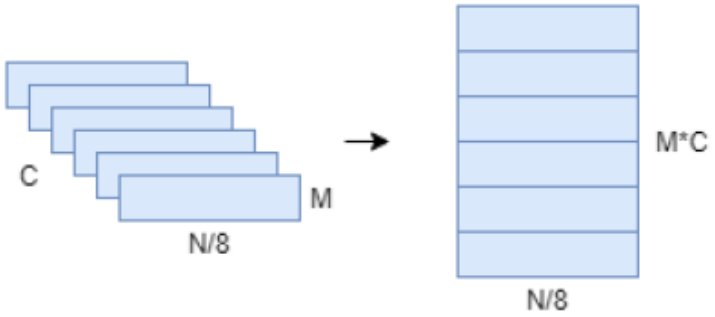


Figure 21: Flattening Stage Scheme

Self-Attention

To apply the self-attention mechanism to the system, we need to apply the method explained in 2.4 to the 512/1024x(N/8 or N/16) flattened matrix. Once we compute the weights, we can apply them to the matrix's values, so end up with the 1x5120 vector that will be the input of the classification layer.

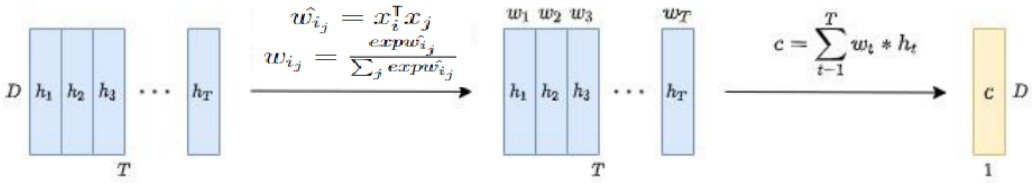


Figure 22: Self-Attention Mechanism Application (D = 5120, T = N/8 or N/16)

Self Multi-Head Attention

To use self multi-head attention in the model, the mechanism explained in 2.4 was also applied to the flattened vector. In this case, the dimension of each head will be 5120/k, with k being the number of heads. Applying Fig.9, the resulting output is again a 1xD vector, as after applying the weights to the each set of heads, the outcome are k 1xD/k vectors, which are concatenated into the 1xD one:

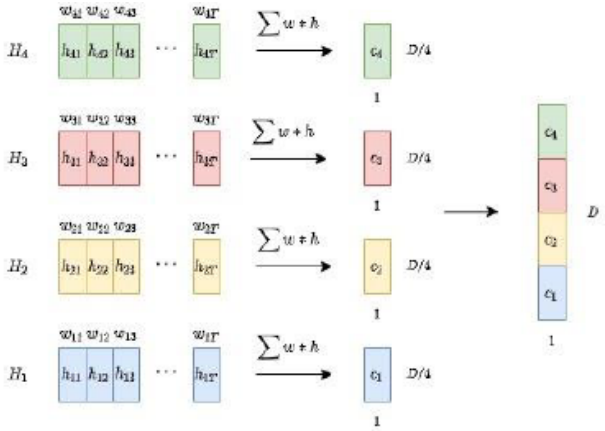


Figure 23: Self Multi-Head Attention Mechanism Application (D = 5120, T = N/8 or N/16, in this case k = 4) [18]

Double Multi-Head Attention

The improvement of double multi-head attention in relation to the multi-head one comes from the discrimination done on the heads, as the most relevant ones are outlined with the second self-attention layer. In this case- as explained in 2.4- a multi-head attention pooling is firstly applied, which is followed by a self-attention one.

Last section showed how, after applying the computed weights to the sets of heads, it resulted in k 1xD/k vectors. Here, instead of concatenating them, they will go through that other self-attention layer- so weights can be applied to the most relevant ones. The resulting vector will now have 1xD/k dimensions, as it will be a weighted average of the k input ones (which also had 1xD/k dimensions).

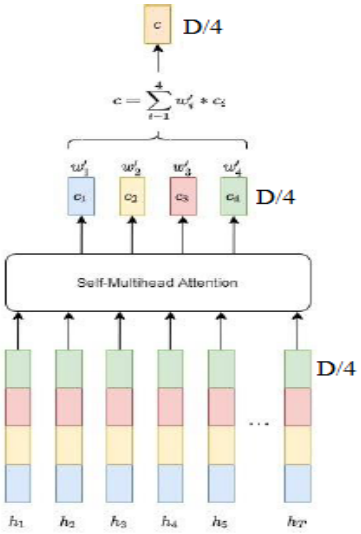


Figure 24: Double Multi-Head Attention Mechanism Application (D = 5120, T = N/8 or N/16, in this case k = 4) [18]

Classification Layers

Once the data has gone through the attention pooling layer and a one-dimensional 1x5120 or 1x5120/k vector is resulted, the classification step using FC layers proceeds. This block- as explained in section 2.3- followed by a SoftMax layer, will output some probabilities for each label, finally choosing for the most probable label.

For this block's configuration, two FC 400-dimensional layers are each followed by a batch normalization step [33] and a ReLU function one. A third layer- also with a dimension of 400- is added without the other two steps. Finally, a fourth layer- previous to the SoftMax function- performs the last classification, so it has as many dimensions as classes the network is classifying.

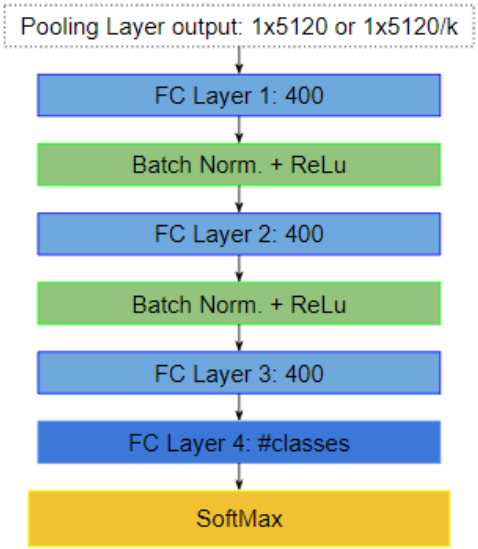


Figure 25: Classification Layers Configuration

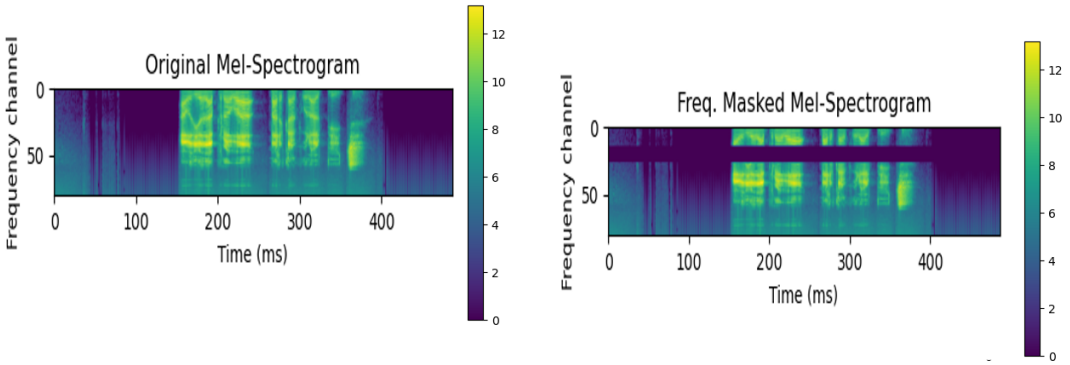
Furthermore, the CE and WCE and losses, described in section 2.5, were also implemented. In addition, in order to improve the performance in the age task, a system based on [34][35] was implemented. With this method, the model outputs a fake regression scores instead of a pure classification ones (if the result was class "fourties"/"[0.1,0.2,0.23,0.93,0.56,0.21,0.09]", now it would be something like "[0.9,0.9,0.9,0.83,0.1,0.1,0.1]"/"fourties", so it simulates a regression). For this new approach, a binary cross-entropy was used, and was globally designed as a new loss called *ordinalreg* loss.

3.3 Improvements with Data Augmenation

As seen in section 2.6, there is a large range of data augmentation techniques to be applied in spectrogram-based classification CNNs. In this project, the first approach consisted in trying the most basic techniques, as the software was easier to implement. To start it off, noise injection-based data augmentation was tried. After some experiments, this technique ended up being dismissed, as the results- despite showing some performance improvement- were not the expected one.

After the verification that noise injection was not the most effective technique for this specific task, the research developed in section 2.6 was done. As explained, the mechanism that showed the better results was frequency or frequency+time masking. For its implementation, the SpecAugment article [30]- developed by a Google Research group- was followed.

The Python library PyTorch- really used through the development of this project- actually provides some methods to implement the techniques described in the SpecAugment paper [36], so it was more simple to fit it in the rest of the code. Both the channels where the mask was applied and its duration, were randomized (with the channel number being between 0 and 80 and the duration from 4 to 16).



(a) Spectrogram without Frequency Masking

(b) Spectrogram with Frequency Masking

Figure 26

3.4 Software

This project was mainly developed using the UPC/TSC Calcula server. Using the ssh protocol, I was able to access this server's resources remotely at any time. The specific tools used through the project's development are the following:

Development software

- Windows 10: Local Operating system
- Ubuntu 18.04.6 LTS: Calcula remote server Operating system
- Python/Python 3: Programming language used
- Visual Studio Code: Local Text editor
- nano and Vim: Server's text editors

Remote Connection

- ssh protocol: Protocol used to access the Calcula server
- Open VPN (GUI): Application which connects your local computer to a VPN network, in this case the TSC one
- byobu: Really useful tool which multiplex your sessions in the Calcula server, allowing to have multiple windows open from you terminal and leave a session open even with your local computer closed

Python Packages/Environment

- Anaconda: Python package environment administrator
- Torch: Essential package for CNN's architecture development and data augmentation techinques
- Matplotlib: Library with many plotting options used in many statistical charts such as the ones in this report (histograms, confusion matrices,...)
- wandb: Library which saves the loss and accuracy evolution graphics during training in your wandb.com account
- sklearn: Library with multiple Deep Learning tools

4 Experiments and Results

4.1 Preliminary Experiments

Preliminary Experiment: Naive Bayes pitch-based gender classifier

Before the final experiments are set, it is needed to get some baseline results from some classical methods. In this case, a naive pitch-based Bayes gender classifier will provide us some preliminary metrics that our Deep Learning system will later have to improve. This classification method consists in obtaining the posterior probabilities of the pitch values for each of the gender classes (Male and Female).

The first step in this application, consists in computing the train and test databases audios' pitches. To do so, it has been used the Praat software, which provides a high pitch accuracy [37].

With the training set pitch values, we proceed with the computation of the two gaussians, with their associated classes. Only audios with a length above 2.5 seconds were used, as the conditions needed to be the same as the final gender experiments (see section 4.2). With these two distributions defined and their posterior probabilities, we sweep through all the possible pitch-decision thresholds, looking for the one that leads to the best Accuracy measure.

The final step consists in computing again the Accuracy and f-score measurements for the obtained threshold but, in this case, using the distribution of the two classes on the train dataset:

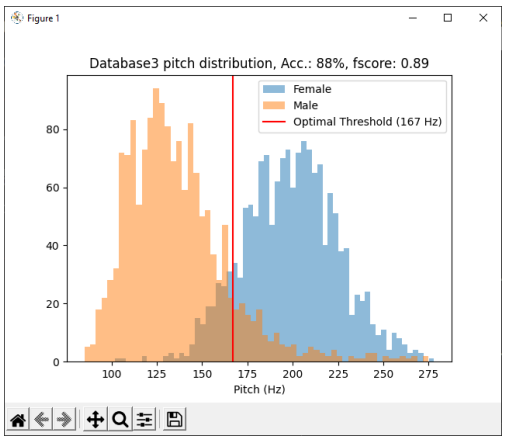


Figure 27: Train-dataset gender classes distribution and experiment results

The final results of this first baseline experiment were:

	Accuracy	F-Score
Gender	88%	0.89

Table 4: Baseline Gender Experiment Results

4.2 Gender Experiments

Experiment sets and settings

As explained in previous chapters, for the gender classification task, only the *male* and *female* classes have been considered. For this task's experiment, as the number of samples from each class in the dataset was large enough, almost completely balanced classes were used for the training set (around 70.000 samples from each class).

One important filter applied, relates to the length of the audios used. The length (in seconds) of this task's classes are represented in the Appendix **Gender Classes Audio Length** histograms. These distributions are crucial to be taken into account, as the system's "Window size" parameter (the length of the window we apply to the signal) has a strong correlation with better or worse results. A wider window will represent more information, but also represents more dismissed audios (if the signal's length is shorter than the window's one, that audio will be dismissed), so a compromise between the amount of information taken from each audio and the number of dismissed ones is shown. Eventually, that window's length was defined at 2.5s.

Experiment results

As the classes were already almost perfectly balanced, for this testing experiment, using the weighted loss did not represent any improvements. The training, validation and testing metrics, along with the validation and test confusion matrices (using the weighted loss) are:

	Train		**Validation**		**Test**	
	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
Cross-Entropy	99,11%	0.99	97,14%	0,95	95.6%	0,96
WCE	99,15%	0.99	97,29%	0,955	95,8%	0,96

Table 5: Gender Experiment Results

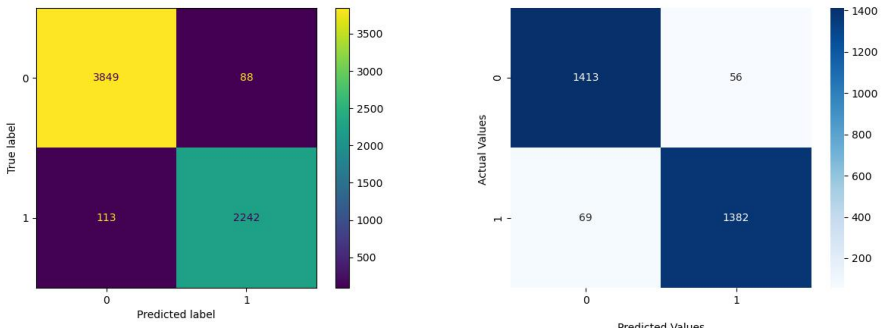


Figure 28: Gender Validation and Test Confusion Matrices, Male = 0; Female = 1

Improvements with Data Augmentation

For the gender classification task, as the training set without data augmentation was already balanced, data augmentation was only used to increase the number of samples from each class (x2, each classes' samples rise to 140000). This approach was just to check if the data augmentation techniques were properly working, as augmenting the number of samples in an already-balanced set which also has a large amount of samples should not represent almost any improvement.

The results were the following:

	Train		**Validation**		**Test**	
	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
Cross-Entropy	99,20%	0.99	97,24%	0,95	96.1%	0,96
WCE	99,23%	0.99	97,48%	0,96	96.3%	0,96

Table 6: Gender Experiment Results with Data Augmentation

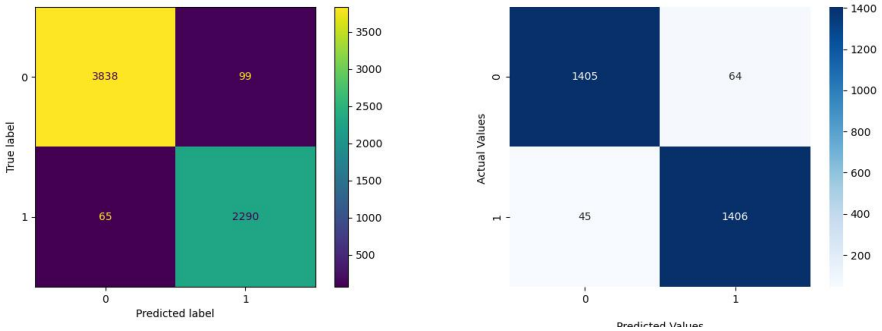


Figure 29: Gender Data Augmentation Validation and Test Confusion Matrices, Male Label = 0; Female Label = 1

4.3 Age Experiments

For the age task, multiple approaches were set out. One of the main constraints from this task's classification, was the strong imbalance to be faced. The *eighties* class was dismissed from the beginning, as the 50 samples available were not enough to develop any proper training. The *teens* class was included as another class in the training but, despite being one of the most distinctive classes from a feature stand point, the low number of samples ended up hurting the system.

As in gender's experiments, only audios shorter than 2.5 seconds were used. See Appendix **Age Classes Audio Length** for the classes audios' length distributions.

Experiment 1: Classifying Three Age Classes

Experiment sets and settings

In this first age experiment, a more simple approach, previous to the all-classes classification one, was taken. The idea was to re-classify all the classes into only three:

- Youngsters: 18-39 y.o.
- Adults: 40-59 y.o.
- Elderly: 60-79 y.o.

The train set distribution for this experiment was the following:

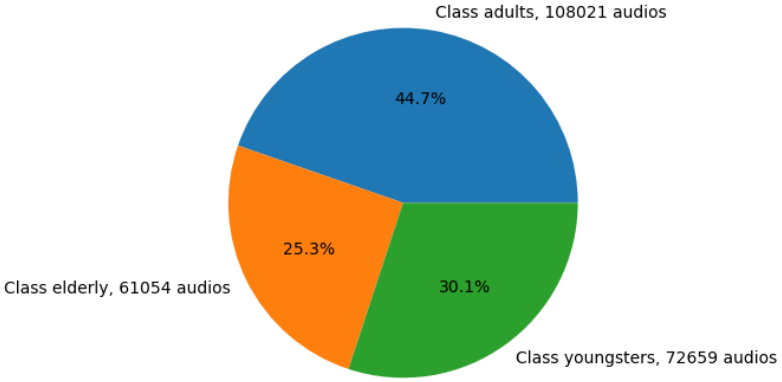


Figure 30: Three Age Classes Training Set Distribution

Results

In this case, as explained in section 3.2, the ordinalreg concept was implemented and used, as well as the other basic losses. This approach showed slightly better results for this task than CE and WCE, so it was chosen as the main model/loss:

	Train		**Validation**		**Test**	
	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
Ordinalreg	98%	0,99	59%	0,49	57%	0,47
Weighted	98%	0,98	55%	0,46	55%	0.45

Table 7: Baseline Age Experiment Results

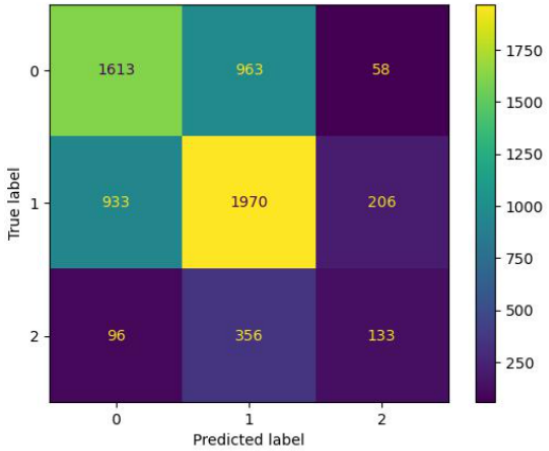


Figure 31: Age Validation Confusion Matrix, ordinalreg,
Youngsters = 0; Adults = 1; Elderly = 2

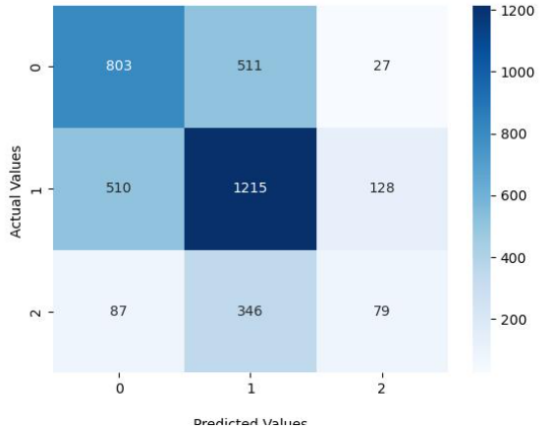


Figure 32: Age Test Confusion Matrix, ordinalreg
Youngsters = 0; Adults = 1; Elderly = 2

Experiment 2: Age Classification with 7 Classes

For this age experiment, all the classes from *teens* to *seventies* (both included) participated. To goal was to see how the system performed classifying all the classes at once, without any data augmentation or balancing techniques.

As in the other age classification experiment, ordinalreg was also declared as the main model/loss. WCE also showed a small improvement compared to the simple CE one, as this last one was quickly dismissed due to not compensating the significant imbalance.

The results obtained from this experiment were underwhelming, as- despite having really good training metrics (97% Accuracy and 0.98 F-Score)- the network was not able to properly generalize so it could not recognize the less-represented classes in the validation and testing steps:

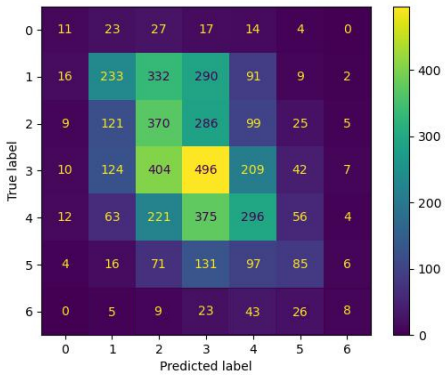


Figure 33: Age Validation Confusion Matrix, ordinalreg,
teens = 0; twenties = 1; thirties = 2;
fourties = 3; fifties = 4; sixties = 5; seventies = 6

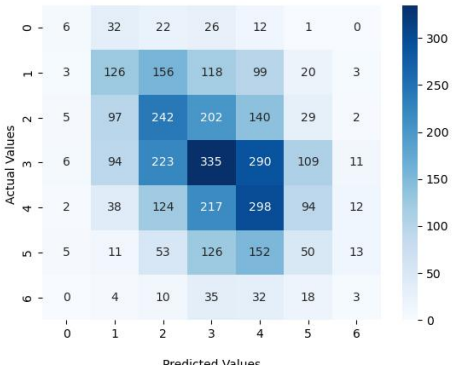


Figure 34: Age Test Confusion Matrix, ordinalreg,
teens = 0; twenties = 1; thirties = 2;
fourties = 3; fifties = 4; sixties = 5; seventies = 6

Improvements with Data Augmentation

Unlike the gender experiments, for this imbalanced age classification task, data augmentation had some room for improvement. In this case, some of the classes- like *teens*- had very few samples, in contrast to *fifties*, which had by far the most.

The data augmentation criteria was then to partially compensate the less-represented classes, creating more replicas from their samples. Frequency masking was also applied to the rest of the classes as, if that was not the case, the system could end up classifying between mask/no-mask.

Ultimately, this data augmentation techniques improved the result metrics by around 2% (absolute Accuracy score) and 0.02 (f-score).

4.4 Accent Experiments

For this last task, many of the classes were dismissed for the two experiments set below, as they did not have enough samples. Like in the other tasks, the audios shorter than 2.5 seconds were also dismissed. The length histograms of the different accent classes can be seen in Appendix **Accent Classes Audio Length**.

Accent classification can be a really hard challenge, as all the accents come from the same language and have a lot of similar prosody/phonetic features. For that reasoning, two experiments were set: a more simple-binary one, and a complete one with the five most representative classes.

Experiment 1: Binary Accent Classification (central-like/oriental, valencian-like/occidental)

For this first experiment, the five most represented classes were merged into only two: central-like/oriental and valencian-like/occidental. This is a really common Catalan dialect classification, as the subclasses included in these two more global ones share a lot of phonetic characteristics (as well as lexical, but those are not as relevant for the task at hand) [38].

Experiment sets and settings

For this experiment, both classes had nearly the same number of samples, around 32.000. Indeed, this was an important requirement, as one of the main goals of this experiment was to analyze how the system performed under the most simple settings (classifying two imbalanced classes).

Results

For this experiment, only the CE and WCE were used. As the two classes were already almost perfectly balanced, the difference between this losses was minimum. The results for the WCE were:

	Train		**Validation**		**Test**	
	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
Cross-Entropy	92%	0,92	74%	0,57	72%	0,62
WCE	92%	0,92	75%	0,58	73%	0,64

Table 8: Baseline Accents Experiment Results

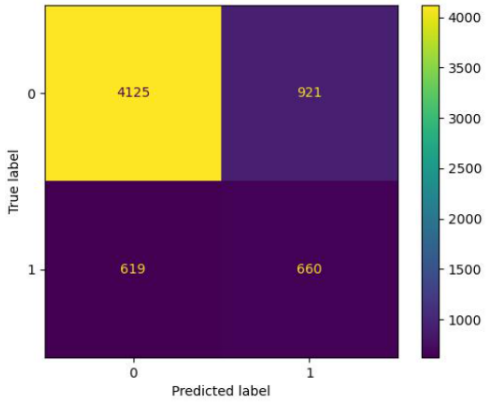


Figure 35: Accents Validation Confusion Matrix,
Oriental = 0; Occidental = 1

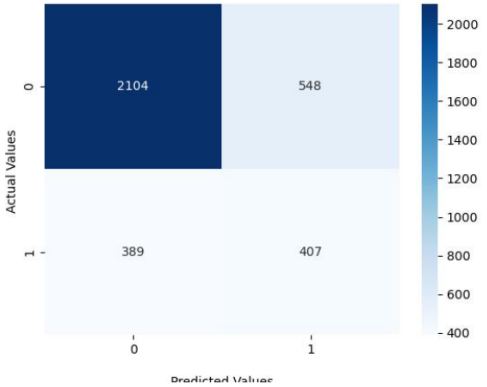


Figure 36: Accents Test Confusion Matrix,
Oriental = 0; Occidental = 1

Experiment 2: Accents Classification with 5 Classes

For this experiment, the five classes with the most samples were used. As in the second age experiment, the goal was to see how the network performed in classifying all the classes at once, without applying any data augmentation or imbalance-compensation techniques. As in the two-classes experiment, the WCE and CE were used. In this case, as the classes were strongly imbalanced, the WCE showed better performance.

The results of this experiment were also quite poor. As expected, the system classified most of the samples as *central* due to the severe imbalance in the training set towards this class. This behaviour was reflected in the testing metrics, as the Accuracy reached a decent 72% (the test set was also slightly imbalanced towards the *central* class, so most of the samples were also predicted as *central* and consequently boosted this metric) and a 0.23 f-score.

Improvements with Data Augmentation

As mentioned in last section, for this task, imbalance between classes was even more present, as the *central* class- despite already pre-filtering out many samples in the last experiment- still had 40% of the total samples. On the other hand, *balear* and *northern* only represented around the 4% of the train set's samples.

Similar to the procedure in the data augmentation section of the age task, the distribution of the classes of last section's experiment was compensated, so the influence on the training set of the less-represented classes was increased. In any case, frequency masks were also applied to all samples.

The results showed a better performance in the classification of the less-represented classes. This can be seen in the F-Score metrics of the validation and test steps, where it improved 0.04 and 0.06 points, respectively, in comparison to the ones without data augmentation. However, as the network did not predict as much the *central* class, the test accuracy actually dropped (from 72% to 63%).

5 Budget

This project doesn't aim to create any product or prototype, as it is purely an investigation one. Consequently, the cost can be calculated as a summation of the engineers involved salaries and the cost of the used resources.

For the first one, the cost of a junior engineer can be calculated if we take into account that this thesis represents a total of 18 ECTS (each ECTS equals 30h of work) and a junior engineer's wage is around 12€/hour. As two senior engineers also participated in this project, their associated cost needs to be included too. We can consider that we did a 2-hour meeting each week for fourteen weeks, with a wage of 22€/hour. This would add up to:

$$C_1 = 18ECTS * 30\frac{hours}{ECTS} * 12\frac{€}{hour} + 2 * 14weeks * 2\frac{hours}{week} * 22\frac{€}{hour} = 7712€$$

For the second one, it can be computed as the cost of the laptops used in addition to the cost of the servers. For the laptops, we can consider that three 600€ laptops were used. As for the server, while the UPC's Calcula was used, it can be considered that some remote Cloud service, like the Google one [39], was used. This server costs around 2.50€/hour. If we consider that around 50 experiments were done, with an approximate duration of 2 hours each:

$$C_2 = 3lap * 600\frac{€}{lap} + 50exp * 2\frac{hours}{exp} * 2,50\frac{€}{hour} = 2050€$$

Finally, the total cost of the project will be:

$$C_total = C_1 + C_2 = 9762€$$

6 Conclusions

Through the development of this project, a complete model capable of characterizing a Catalan speaker using their voice spectrogram was established. Overall, both the research done and the project development itself, proved why Deep Learning-based approaches can offer great results in these tasks.

The Catalan Common Voice dataset- as well as being the main motivation for this project's creation- was an essential tool for its development. The amount of quality voice data in it will surely lead to many other projects in the AI field. As the most-represented classes already have enough samples, the donations of the less-represented ones should be promoted.

The enhancements in the model and the used data improved all the department's previous results in this task but, despite the advances in the model explained in this report, it still showed the same behaviour (high binary-gender classification accuracy, poor age/accent classification with more classes).

As mentioned, the model results were ultimately diverse. It proved to be really efficient in binary classification, specially in the gender task- were it reached state-of-art results. The binary accent classification experiment also verifies this theory, as it showed a massive improvement in comparison to classifying with all the classes.

Related to the previous point, the system performed really poorly when it was trained with hardly-imbalanced sets. When classifying all the age and accent classes, those with limited number of samples ended up being almost unrecognizable in the test and validation steps.

With the system not being able to recognize some of the classes, the accuracy scores were exponentially affected in negative way. Looking at some of the confusion matrices- for instance the multi-class ages ones- a clear tendency towards the desirable diagonal line can be seen. The system did not differentiate the classes with a small number of samples and sometimes got confused with the better represented ones.

The system also showed some overfitting tendencies. In most of the experiments with poor result metrics, it can be seen how the training step was performed with high accuracy but then the network was not able to correctly generalize in the validation and testing step.

Data augmentation proved to be partially successful, as it improved the results in almost every experiment it was applied, but ultimately did not solve in a bigger scale any of the network's weaknesses caused by imbalanced sets.

It is also vital to choose the correct approach for the brought up system. For instance, for the age experiment, a regression is almost mandatory- as classifying some of the classes was a really challenging task for the network due to their similarity. The model's characteristics and layers should address the task's demands, not the opposite.

Finally, on a more personal note, despite the results not being as impressive as I would liked, I am really satisfied with how much I have learned through the development of this

project. Being able to develop this kind of complex network with the help of experts on the field was such a great experience.

7 Future Work

As the classification on some of the tasks did not go as well as expected, there are some improvements that could be interesting to apply.

The major future work-line I would like to bring up is to reconsider the whole system. While it showed good performance on some of the tasks, others showed strong overfitting and overall underwhelming results. As mentioned in this report, some of the network characteristics were changed and tested out- for instance, the front end block was tried with 2, 3 and 4 VGG layers to avoid overfitting, but the results were always similar. Therefore, some reconsiderations on the main blocks of the network could improve a exponentially its performance.

Another interesting improvement I would advocate for would be applying multi-task learning. As explained in the state-of-art section, many networks used for similar tasks apply multi-task learning, specially merging the training for gender and age. In this project I could not develop the system to do so, but it could definitely improve some of the network's performance.

Finally, the last point that would also help improving the system's performance would be improving the dataset. As mentioned in last section, the Catalan Common Voice dataset has been crucial for this project's development, but the low volume of samples for some of the classes ended up hurting the training a lot. Balanced classes with a large amount of samples will surely lead to state-of-art results in the future.

References

[1] Paul Boersma and David Weenink. *AINA: La nostra llengua és la teva veu*. December 2020. URL: https://smartcatalonia.gencat.cat/ca/projectes/tecnologies/detalls/article/AINA.

[2] Santiago Escuder. "Catalan age, accent and gender classification using Common Voice database". In: *Universitat Politècnica de Catalunya* (2021).

[3] Pooyan Safari Miquel India and Javier Hernando. "Self Multi-Head Attention for Speaker Verification". In: *Universitat Politècnica de Catalunya* (2019).

[4] Martin Russell Maryam Najafian. "Modelling Accents for Automatic Speech Recognition". In: (2015), p. 1568.

[5] Fatih Ertam. "An effective gender recognition approach using voice data via deeper LSTM networks". In: *Applied Acoustics* 156 (Aug. 2019), pp. 351–358. DOI: 10.1016/j.apacoust.2019.07.033.

[6] Mucahit Buyukyilmaz and Ali Osman Cibikdiken. "Voice Gender Recognition Using Deep Learning". In: (2016).

[7] J.Y. Tursunov A.; Mustaqeem: Choeh and S Kwon. "Age and Gender Recognition Using a Convolutional Neural Network with a Specially Designed Multi-Attention Module through Speech Spectrograms". In: (2015).

[8] Matthieu J. Guitton Catherine L. Lortie Mélanie Thibeault and Pascale Tremblay. "Effects of age on the amplitude, frequency and perceived quality of voice". In: (2015).

[9] R.R. Deshmukh Maheshkumar B. Landge and P.P. Shrishrimal. "Analysis of Variations in Speech in Different Age Groups using Prosody Technique". In: (2015).

[10] S Schotz. "Perception, analysis and synthesis of speaker age". In: (2006).

[11] A. Kelly F.; Drygajlo and N Harte. "Speaker verification in score-ageing-quality classification space". In: (2013).

[12] Mohamad Hasan Baharia; Mitchell McLarenb; Hugo Van hammea and DavidA. van Leeuwenb. "Speaker age estimation using i-vectors". In: (2014), pp. 99–108.

[13] H.-J. Na and J.-S. Park. "Accented Speech Recognition Based on End-to-End Domain Adversarial Training of Neural Networks". In: (2021).

[14] Keven Chionh; Maoyuan Song and Yue Yin. "Application of Convolutional Neural Networks in Accent Identification". In: (2020).

[15] Mok Wei Xiong Edmund Leon Mak An Sheng. "Accented Speech Recognition Based on End-to-End Domain Adversarial Training of Neural Networks". In: (2019).

[16] Leland Roberts. *Understanding the Mel Spectrogram*. 2020. URL: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53.

[17] URL: https://databricks.com/glossary/convolutional-layer.

[18] Daniel Aromí Leaverton. "Predicting emotion in speech: a Deep Learning approach using Attention mechanisms". In: *Universitat Politècnica de Catalunya* (2021).

[19] A. Vaswani; N. Shazeer; N. Parmar; J. Uszkoreit; L. Jones; A. N. Gomez; Ł. Kaiser and I. Polosukhin. "Attention is all you need". In: *Advances in Neural Information Processing Systems* (2017).

[20] K. Rao C.-C. Chiu; T. N. Sainath; Y. Wu; R. Prabhavalkar; P. Nguyen; Z. Chen A. Kannan; R. J.Weis; and E. Gonina. "State-of- the-art speech recognition with

sequence-to-sequence models". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), pp. 4774–4778.

[21] Jianpeng Cheng; Li Dong; and Mirella Lapata. "Long short-term memory-networks for machine reading". In: (2016).

[22] Daniel Povey et al. "A Time-Restricted Self-Attention Layer for ASR". In: (2018), pp. 5874–5878. DOI: 10.1109/ICASSP.2018.8462497.

[23] Pooyan Safari Miquel India and Javier Hernando. "Double Multi-Head Attention for Speaker Verification". In: *Universitat Politècnica de Catalunya* (2019).

[24] afb labs. *Understanding Backpropagation(Gradient Descent) in Neural Networks for Binary Classification*. 2020. URL: https://medium.com/@afblabs7/understanding-backpropagation-gradient-descent-in-neural-networks-for-binary-classification-d5305f0765e8.

[25] Naomi Schalken. *KL divergence between two normal distributions*. 2017. URL: https://www.researchgate.net/figure/KL-divergences-between-two-normal-distributions-In-this-example-p-1-is-a-standard-normal_fig1_319662351.

[26] Manning Publications Co. *Data Loaders, Deep Learning with PyTorch*. 2020. URL: https://livebook.manning.com/concept/deep-learning/dataloader.

[27] Pragya Soni. *Data augmentation: Techniques, Benefits and Applications*. 2022. URL: https://www.analyticssteps.com/blogs/data-augmentation-techniques-benefits-and-applications.

[28] Victor Emilio Hernández Leal. "Emociones en señales de voz: reconocimiento con redes neuronales profundas". In: *Universitat Politècnica de Catalunya* (2021), p. 32.

[29] Navdeep Jaitly and E. Hinton. "Vocal Tract Length Perturbation (VTLP) improves speech recognition". In: (2013).

[30] Daniel S. Park; William Chan; Yu Zhang; Chung-Cheng Chiu; Barret Zoph; Ekin D. Cubuk and Quoc V. Le. "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition". In: (2019).

[31] Python Documentation. *pickle — Python object serialization*. 2021. URL: https://docs.python.org/3/library/pickle.html.

[32] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2015), pp. 1–14.

[33] Jason Brownlee. *A Gentle Introduction to Batch Normalization for Deep Neural Networks*. 2019. URL: https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/.

[34] Jianlin Cheng; Zheng Wang and Gianluca Pollastri. "A neural network approach to ordinal regression". In: (2008).

[35] Mathias Gruber. *How to Perform Ordinal Regression / Classification in PyTorch*. 2021. URL: https://towardsdatascience.com/how-to-perform-ordinal-regression-classification-in-pytorch-361a2a095a99.

[36] Torchaudio Contributors. *Audio Feature Augmentation*. 2022. URL: https://pytorch.org/audio/main/tutorials/audio_feature_augmentation_tutorial.html.

[37] Paul Boersma and David Weenink. *Praat: Doing phonetics by computer*. 2021. URL: https://www.fon.hum.uva.nl/praat/.

[38] Mònica López Bages. *LES VARIETATS GEOGRÀFIQUES O DIALECTALS: CATALÀ ORIENTAL I OCCIDENTAL*. URL: https://blogs.cpnl.cat/nivelldtarragona/files/2011/11/oriental-occidental1.pdf.

[39] *Google Cloud server*. URL: https://cloud.google.com/compute/gpus-pricing.

Appendices

Work-Packages

Project: Investigation and research	WP ref: IR	
Major constituent: Document	Sheet 1 of 4	
Short description: In this Work Package I am going to be studying all the state-of-art techniques that I might need for the project. Concepts like Convolutional Neural Network, losses and loaders or self multi-head attention should be clear and dominated after this section.	Planned start date: 14/02/2022 Planned end date: 07/03/2022	
	Start event: First contact with project's content End event: System and work-plan definition	
Internal task T1: Theory research on key concepts Internal task T2: State-of-art systems and algorithms research Internal task T3: System and work-plan definition	Deliverables: - Work Plan	Dates: (specified in the Gantt diagram)

Figure 37: Work package 1

Project: Main model development	WP ref: DVPM	
Major constituent: Software	Sheet 2 of 4	
Short description: In this Work Package I am going to focus on developing the main model and CNN, focusing on its main parts one by one.	Planned start date: 07/03/2022 Planned end date: 18/04/2022	
	Start event: Env. preparation End event: Testing and first improvements	
Internal task T1: Project environment preparation Internal task T2: Model baseline: Bayesian pitch-based gender classification Internal task T3: Feature extraction: Front End with VGG Internal task T4: Feature extraction: Pooling layer with Self Multi-Head Attention mechanisms Internal task T5: Classification layers Internal task T6: First testing and improvements	Deliverables: - Software - Baseline experiment report	Dates: (specified in the Gantt diagram)

Figure 38: Work package 2

Project: Results improvement and data analysis	WP ref: RI & DA	
Major constituent: Statistical results	Sheet 3 of 4	
Short description: In this Work Package I am going to evaluate the designed system with different methods and trying to do some improvements on the database so we achieve the most optimal training.	Planned start date: 18/04/2022 Planned end date: Project ending	
	Start event: Testing designed software with main database End event: Final results presentation	
Internal task T1: Database analysis: Improvements with data augmentation Internal task T2: First results analysis with different losses and loaders Internal task T3: System improvement with multi-task learning Internal task T4: Final improvements	Deliverables: - Final software - Final training database - Final results	Dates: (specified in the Gantt diagram)

Figure 39: Work package 3

Project: Documentation	WP ref: DC	
Major constituent: Documentation	Sheet 4 of 4	
Short description: In this Work Package I am going to document all the procedure I have followed and all the learning so that the whole system can be understood in the future.	Planned start date: 14/02/2022 Planned end date: Project ending	
	Start event: Project Proposal and Work Plan End event: Final results presentation	
Internal task T1: Project Proposal and Work Plan Internal task T2: Critical Review Internal task T3: Final Report	Deliverables: - Final software - Critical Review - Final Report	Dates: (specified in the Gantt diagram)

Figure 40: Work package 4

Common Voice Dataset Distribution

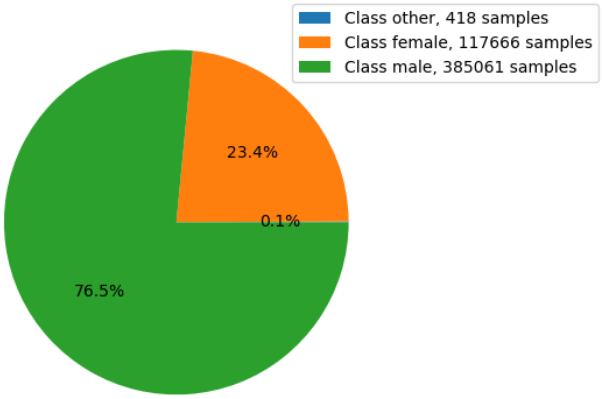


Figure 41: Gender Sample Distribution

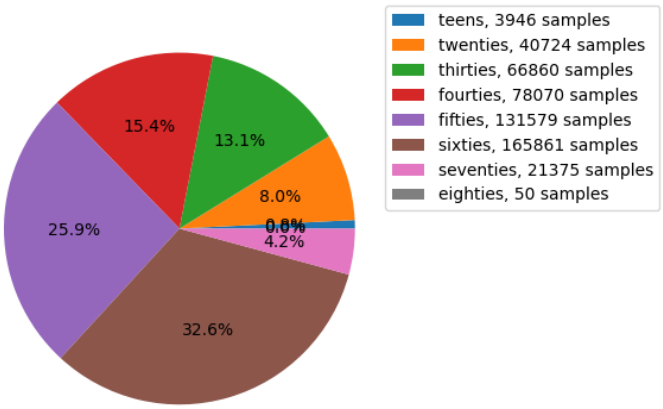


Figure 42: Ages Sample Distribution

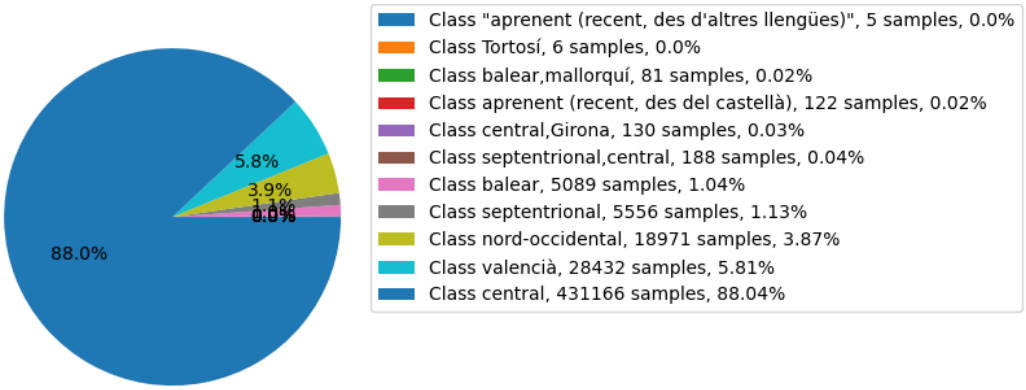


Figure 43: Accents Sample Distribution

Gender Classes Audio Length

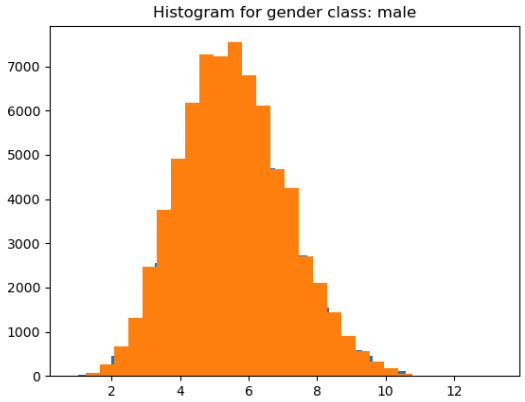


Figure 44: *Male* Gender Audio Length (s) Histogram

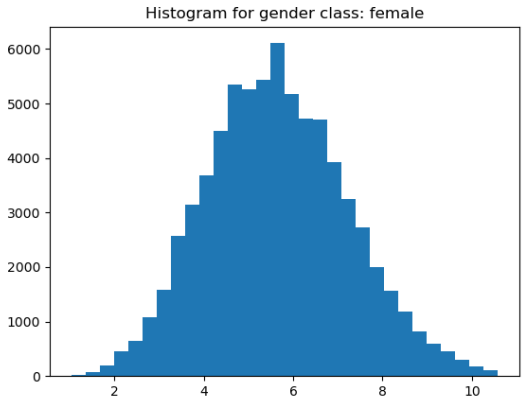


Figure 45: *Female* Gender Audio Length (s) Histogram

Age Classes Audio Length

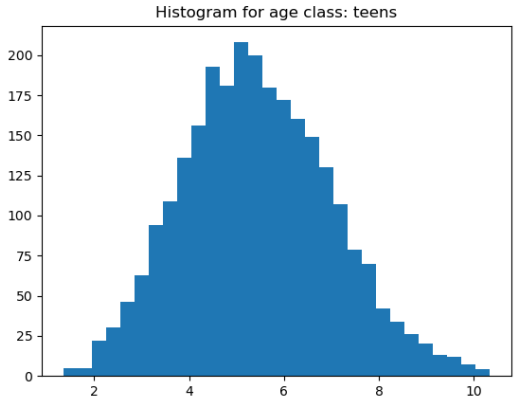


Figure 46: *teens* Age Audio Length (s) Histogram

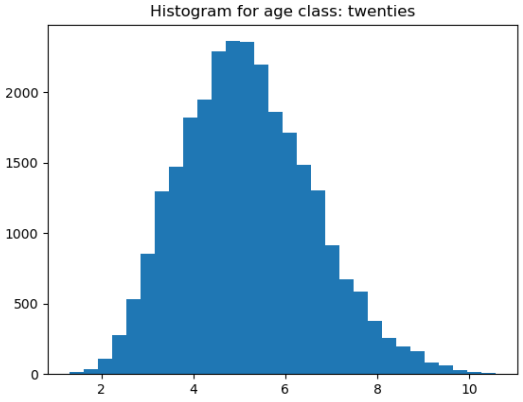


Figure 47: *twenties* Age Audio Length (s) Histogram

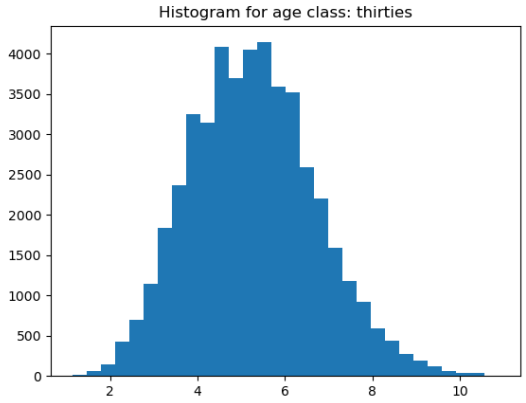


Figure 48: *thirties* Age Audio Length (s) Histogram

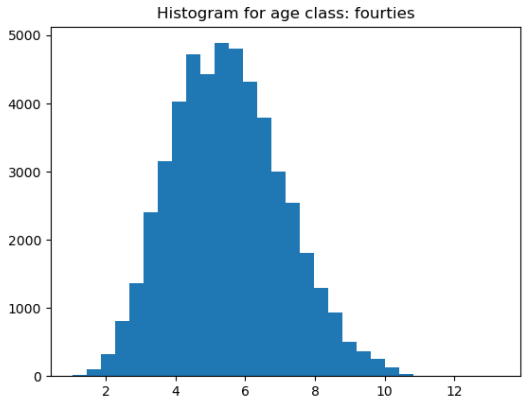


Figure 49: *fourties* Age Audio Length (s) Histogram

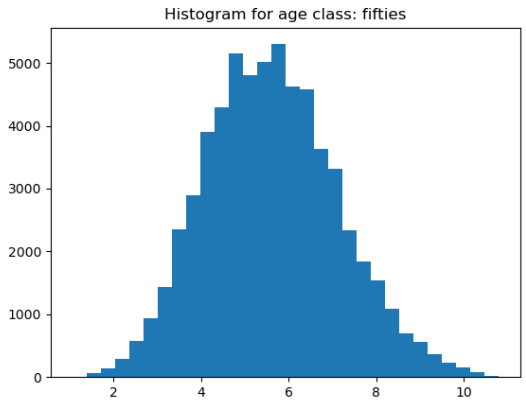


Figure 50: *fifties* Age Audio Length (s) Histogram

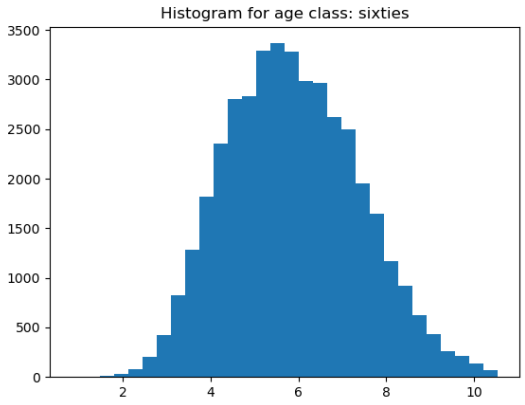


Figure 51: *sixties* Age Audio Length (s) Histogram

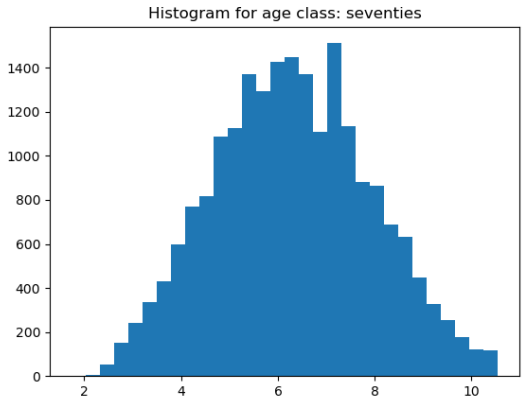


Figure 52: *seventies* Age Audio Length (s) Histogram

Accent Classes Audio Length

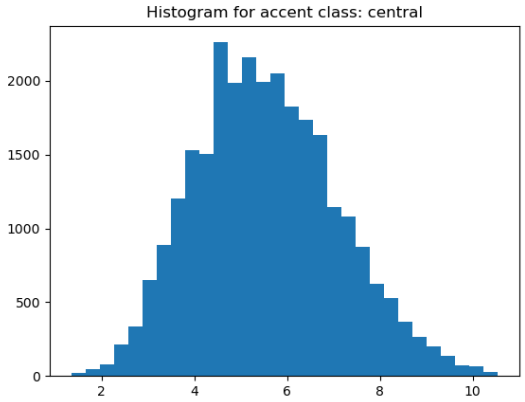


Figure 53: *central* Accent Audio Length (s) Histogram

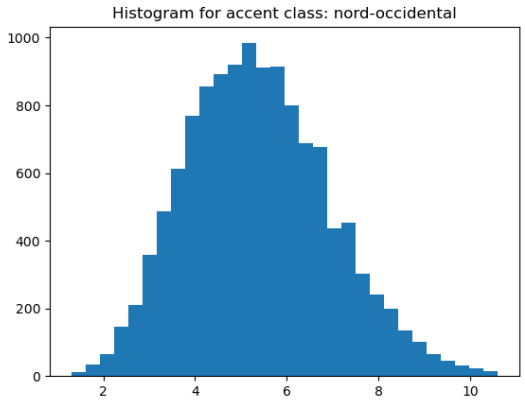


Figure 54: *nord-occidental* Accent Audio Length (s) Histogram

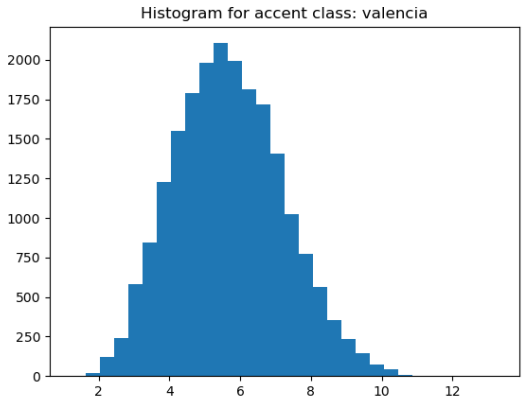


Figure 55: *valencià* Accent Audio Length (s) Histogram

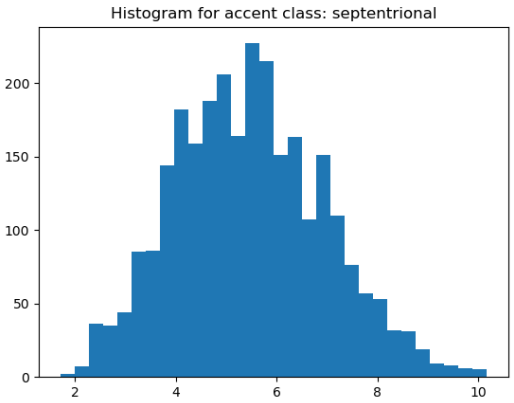


Figure 56: *septentrional* Accent Audio Length (s) Histogram

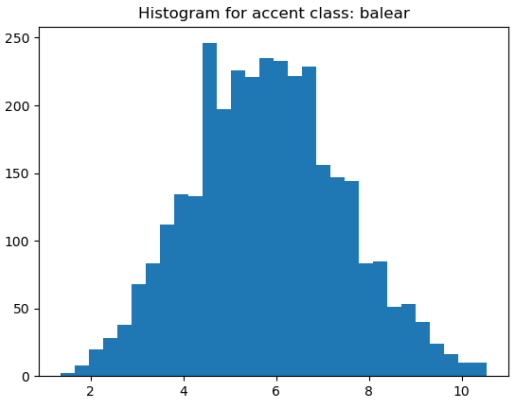


Figure 57: *balear* Accent Audio Length (s) Histogram