**Deep Learning Project**

Name: Kyriaki Bei

AM: : 2022202004015

email: [sandybei058@gmail.com](mailto:sandybei058@gmail.com)

# **Subject:** Speaker Identification by Voice.

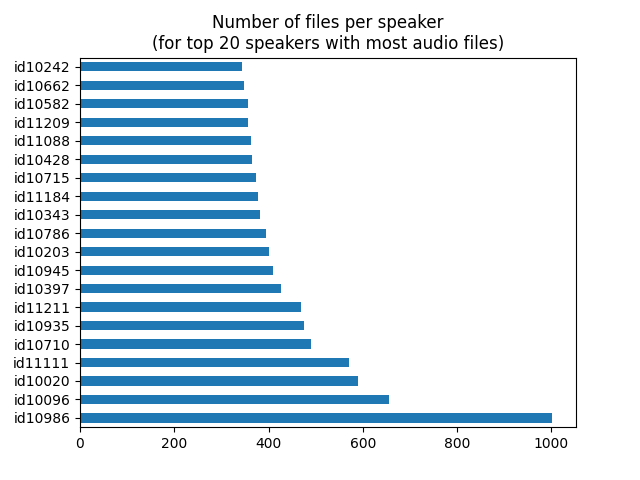
# **The Original Dataset**

The dataset that was used for this project is [VoxCeleb](https://www.robots.ox.ac.uk/~vgg/data/voxceleb/), an audio-visual dataset consisting of short clips of human speech, extracted from interview videos uploaded to YouTube. The dataset consists of two versions, VoxCeleb1 and VoxCeleb2. For this project **VoxCeleb1** dataset was selected. **VoxCeleb1** contains 2 sets of files, **Dev files** and **Test files**. **Test files** contains speakers’ ids that are not included in the **Dev files**, so they could not be used as a test set for our classification task. For this reason, only **DEV files** were used.

|  |  |
| --- | --- |
|  | **DEV files** |
| **# Speakers** | 1,211 |
| **# Files** | 148,642 |

# **The Identification Dataset**

From the1,251 speakers of the **DEV files**, only 10 were selected for identification. To select a balanced subset of **VoxCeleb1**, the speaker ids were sorted by their total number of audio files. Then, 10 speakers were chosen out of the speakers with the most audio files. The speakers were selected so that the resulting dataset would be balanced with respect to the number of audio files per speaker. The resulting dataset consists of **10** speakers and **4,195** audio files in total.



**VoxCeleb1 ID VGGFace1 ID**

id10203 David Attenborough

id10343 Gloria Steinem

id10397 J.J. Abrams

id10710 Louis C.K.

id10715 Lucie Arnaz

id10786 Meat Loaf

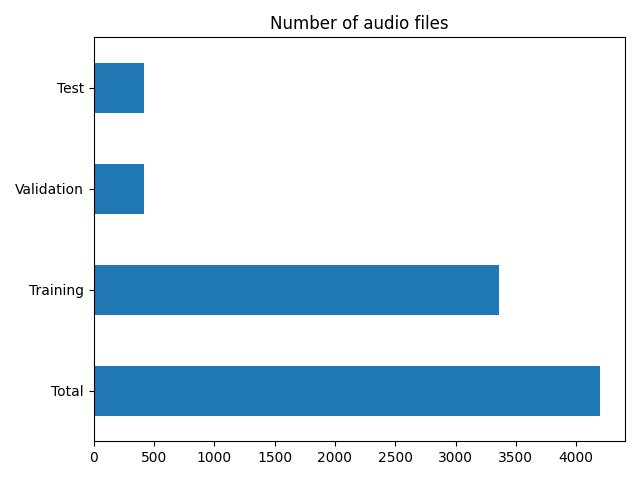
id10935 Peter Jackson

id10945 Quentin Tarantino

id11184 Tom Hooper

id11211 Vince Gilligan

After the generation of the dataset, it then gets split it into a training, validation, and a test set. The training set consists of the 80% of the total audio files, while the test and the validation sets consist of the 10% of files each. The number of files per dataset is presented in the figure below:

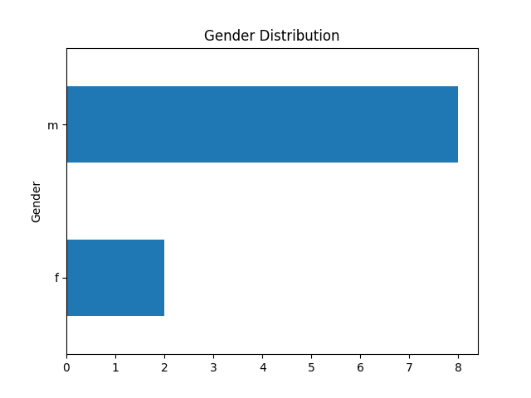
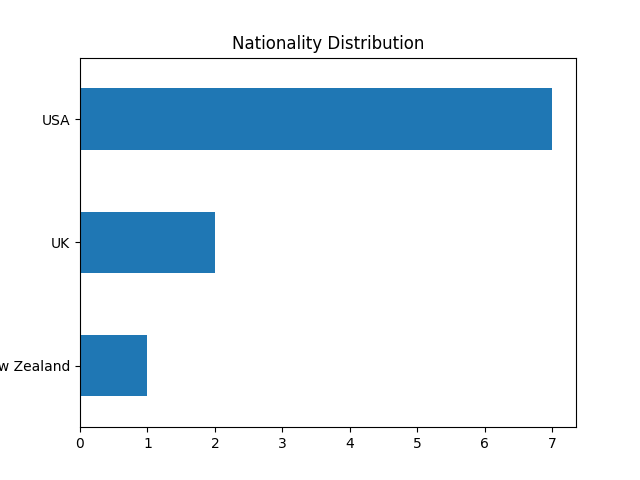


# **Data Exploration**

By running the **explore\_data.py** file, we visualize information about the dataset. In particular, we get 2 plots of the speakers’ gender and nationality distribution. By observing these plots, we see that:

* 80% of the speakers are males and 20% are females
* 10 % are from New Zealand, 30% from UK and 70% from USA

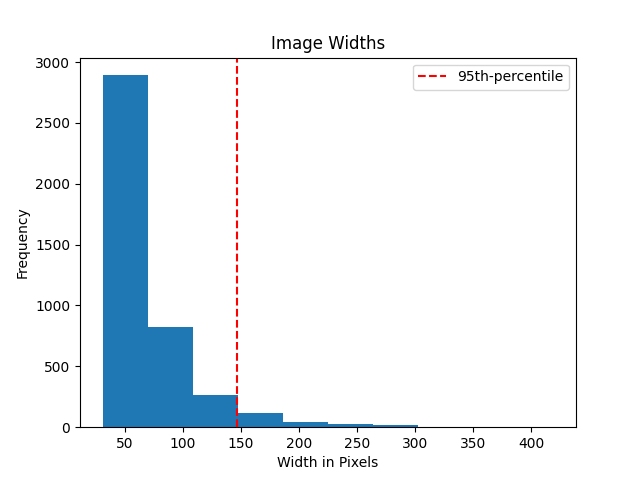
We conclude that our dataset is balanced with respect to the number of audio files per user, but it is unbalanced with respect to the gender and nationality of the speakers.



# **Audio Preprocessing**

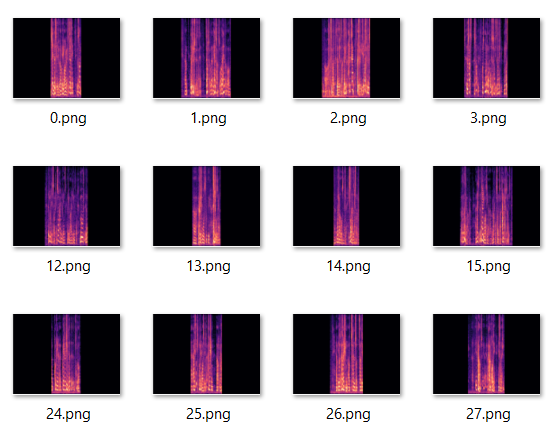
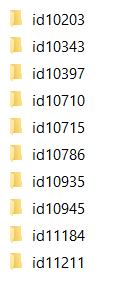
The file **audio\_analysis.py** includes functions for the preprocessing of the audio files. All audio signals have the same sample rate (16,000 kHz) so they do not need to be resampled. The audio preprocessing consists of the following steps:

1. The audio signals get converted to mono so that they have the same number of channels.
2. the processed signals get converted to a Mel spectrogram.
3. The spectrograms get resized to have the same height, width to be fed to the model. The audio clips have different duration, so they get cropped or padded to have equal width. The 95th-Percentile of all spectrograms’ widths was selected to be the optimal image width since it 95% of all images have width less than this number… and information is not lost.



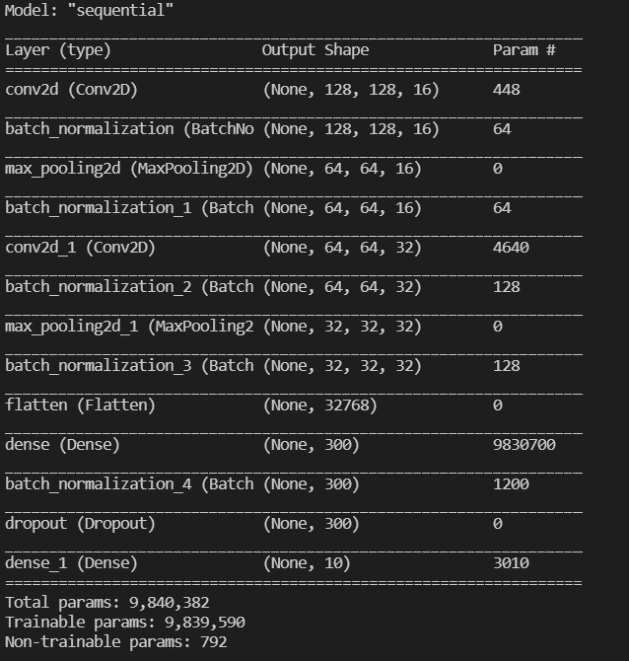
# **Spectrograms Directories Creation**

The file **create\_directories.py** creates 3 directories (train, val, test) with the spectrogram images of the audios of each speaker. Each directory contains 10 folders each one named after the speaker ids and containing spectrogram images of the respective speaker. The folders of the directories and its containing files can be seen in the following figure:

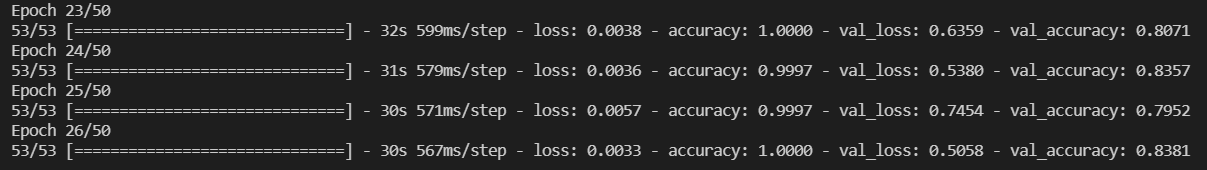


# **Training**

The model that was used for classification is a CNN model that consists of 2 convolutional layers, both containing a **Convolutional** and a **Maxpool** layer, followed by 2 **Fully Connected** layers and a **SoftMax** output layer. To reduce overfitting and improve generalization error **Batch Normalization** layers were added between layers and a **Dropout** layer that drops 50% of the data was added before the last **Fully Connected** layer. The architecture of the network can be viewed in the following figure:



By running **cnn.py** we train the model, and we get the following results:

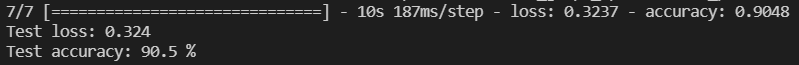


**Train loss:** 0.003 **Train accuracy:** 100%

**Val loss:** 0.505 **Val accuracy:** 83.8%

From the plot of training and validation accuracy, we can see that the model overfits until the first … epochs, but after that the loss keeps falling.

We can evaluate the model’s performance on the test set by running the **evaluate.py** file. The results we get are the following:



Then we perform some prediction on 3 spectrogram images of the test directory. The speakers have different gender and nationality:

* Gloria Steinem (female, UK)
* Tom Hooper (male, New Zealand)
* Quentin Tarantino (male, UK)

As we can see from the figure below, the model identifies correctly all 3 speakers:

