Report Group 42

Floris van Kuijen (2053097), Andra Rice (2049504),
Hannah van der Vlies (2054007) Loes van Voorden (2055036)

Data loading and processing

Pre-processing consisted of normalizing the dataset with the use of Mel-frequency Cepstral Coefficients (MFCC). This resulted in the audio files being transformed from amplitude signals to decibel scaled Mel spectrograms of 201 by 40 pixels (5). This technique was applied for its competence in measuring distances between different languages (Bonet-Solà & Alsina-Pagès, 2021). The targets were one hot encoded to allow for multi-class classification in combination with a Categorical Cross-Entropy loss function.

Architecture design

The basis for the model is a Convolutional Neural Network (CNN) as shown in Figure 1. It has repeated iterations of a 2D convolutional layer activated with ReLU, followed by a Batch Normalization and Max Pooling Layer which was added to decrease complexity. After a Global Pooling Layer comes a flattening layer and a linear layer activated on ReLU. The model ends with a softmax output activation function in order to calculate class probabilities, and ultimately returns the best prediction.

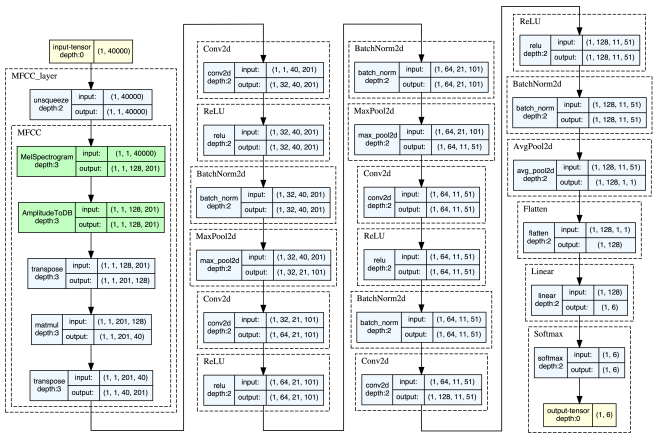

Figure 1: Architecture Diagram.

Experiments

The experiments were conducted by tuning model hyperparameters, which can be seen in Table 1. Models with an altered architecture were also tested, but showed negligible improvements and were therefore left out of the final results.

Results

In general, bigger batch sizes, more epochs, and higher learning rates achieved better results. The best overall results were achieved with b=200, e=50, lr=0.001, performing with an 83.7% accuracy on the test set, and being able to reach 92.3% accuracy on the competition set. The reason the model does not perform optimally is due to the similarity languages have with each other. Languages will share words with other languages or have similar accents and pronunciations. This will cause the model to reasonably make mistakes as to which language is used in a recording. Figure 2 illustrates the similarities between languages from the model's perspective, using PCA.

	Batch Size	Epochs	Learning Rate	Accuracy
1	64	3	0.001	0.510
2	64	3	0.001	0.550
3	100	20	0.001	0.710
4	200	20	0.001	0.825
5	200	20	0.0001	0.820
6	60	50	0.0001	0.759
7	**200**	**50**	**0.001**	**0.837**
8	200	50	0.0005	0.828
9	200	50	0.0001	0.772

Table 1: Table presenting model performance on different hyperparameters.

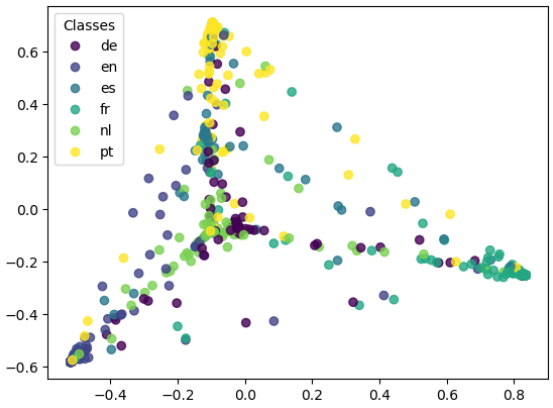

Figure 2: Principal Component Analysis on model.

Discussion and conclusions

- MFCC allowed us to use frequency instead of amplitude as a key feature, thus controlling for the difference in loudness between data points.
- Global Pooling reduced our parameters from around 2,4 million parameters to 130,950 parameters, which greatly improved our model by decreasing complexity and training time.
- We concluded that the model works better with larger batch sizes and more epochs.
- The architecture is a good architecture given that it performs the task with the accuracy rate of approximately 80%.

References

Bonet-Solà, D., & Alsina-Pagès, R. M. (2021). A comparative survey of feature extraction and machine learning methods in diverse acoustic environments. *Sensors*, *21*(4), 1274.

Our full code and process can be found here on GitHub.

Work distribution

All:

- Went to meetings (every Monday and Friday for about two hours every meeting) to discuss our plan for the week and to help each other out with errors or struggles
- Discussed findings and results yet to be obtained
- Coded together

Floris:

- Worked on model created by Loes and Andy and finalized it, managing to decrease model parameters from 2,4 million down to 131 thousand
- Did bugfixing, error solving, and made sure model was usable by the tester code
- Trained and tested many models with different hyperparamters using RTX 2060

Andra:

- Did research on CNN architectures, layers, and the purpose of those layers
- Built the basic CNN model
- Organized scrum master rotation, project timeline, and weekly plans
- Wrote initial Architecture Design, Error Analysis, and parts of the Discussion and Conclusions
- Created Table 1

Hannah:

- Built the training loops for the models
- Researched and helped implement Global Pooling, Multi-classification and MFCC
- Generated the architecture diagram, confusion matrix, MFCC and loss graphs
- General troubleshooting and editing

Loes:

- Did research on the possible implementation of an RNN instead of a CNN
- Got the binary model to work and resolved errors
- Did research on which loss function to use
- Tested multiple versions of the model to see which layers worked best
- Wrote parts of the report
- Set up the architecture for the final model

Appendix

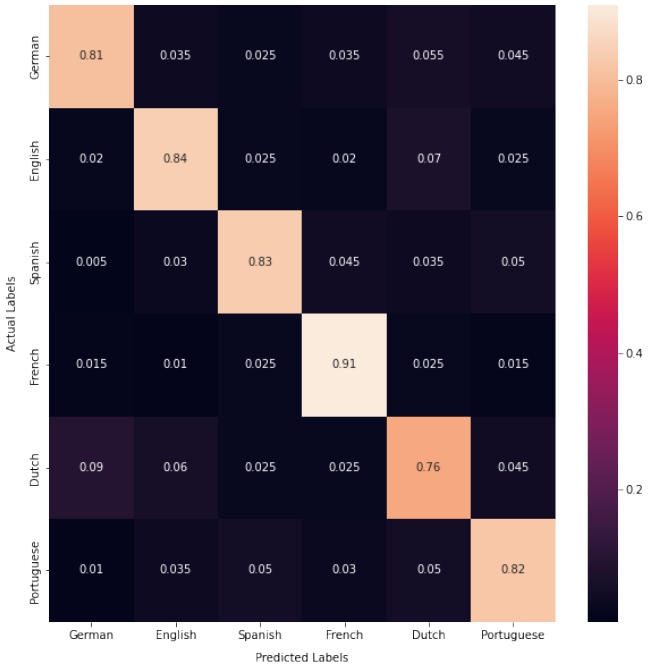

Figure 3: Confusion Matrix on model.

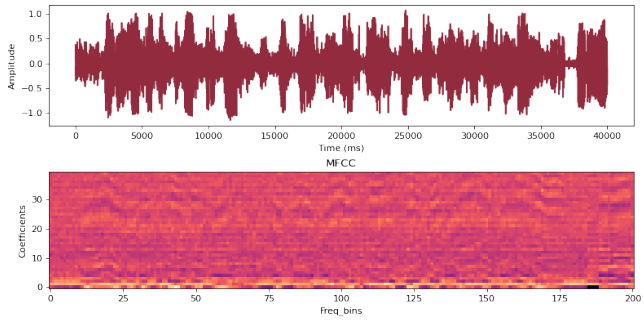

Figure 4: Transformation from waveform to MFCC.

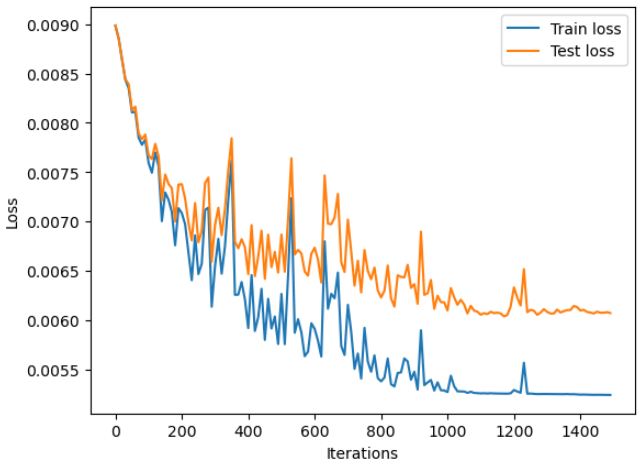

Figure 5: Loss graph for b=200, e=50, lr=0.001, accuracy=83,7%.