Report Group # 30

Laurynas Jagutis (2054970), Nin Khodorivsko (2061810),
Katarzyna Kleczek (2059098), Bram Meijer (2057266)

Data loading and processing

The dataset used in this project consists of waveforms of audioclips, spanning 5 seconds each. Each audioclip corresponds to a German, English, Spanish, French, Dutch, or Portuguese speaker. We loaded the data using `NumPy`. Min-max normalization is performed on the training data to rescale each signal between 0 and 1. This is done to rule out differences caused by a person's pitch. As part of our data exploration process, we defined a function to plot five random samples (see Figure 3).

Architecture design

The architecture of our model is depicted in Figure 1. The model normalizes input data using min-max normalization and applies MFCC spectrogram computation. It includes four convolutional layers (32 output neurons, kernel size 3), followed by ReLU activation and 2D max pooling. An RNN layer (32 hidden features) captures temporal dependencies. A linear layer (64 input, 32 output features) is followed by batch normalization, ReLU activation, and a final linear output layer. The CNN layers were chosen because they are good at extracting local features. The RNN layers model sequential dependencies which are crucial for language.

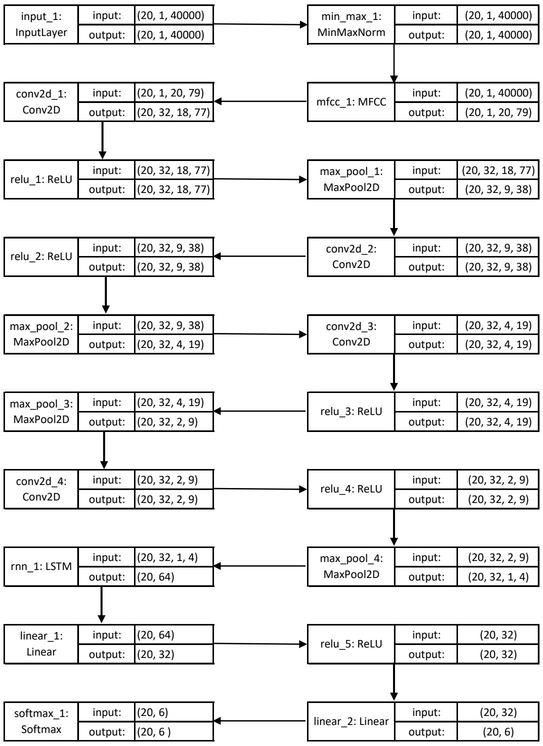

Figure 1: Diagram summarizing the proposed architecture.

Experiments

We first tested the model using only two classes and then we expanded it to encompass all classes. A 0.83 train/test ratio was used. A dropout rate of 0.5 was added to the RNN layer to reduce overfitting. Hyperparameter tuning was performed on the batch size and the learning rate. We used cross entropy as a loss function and Adam as an optimizer.

Results

We found that a learning rate of 0.0001 and a batch size of 20 resulted in the highest test accuracy (0.61; see Table 1). The binary model had a higher accuracy but the hyperparameters used for that model performed poorly for the multi class one. This was unsurprising, however, as the multiclass model needs to learn more subtle differences. From Figure 2 we can see that the highest number of false predictions was caused by Dutch. This is the case for all languages except French where the most false predictions corresponded to German.

model	batch_size	lr	train_acc	test_acc
1	32	0.00001	0.39	0.35
2	32	0.0001	0.78	0.56
3	32	0.001	0.94	0.48
4	**20**	**0.0001**	**0.88**	**0.61**
5	20	0.00001	0.40	0.40
6	20	0.001	0.12	0.20

Table 1: Table presenting the results

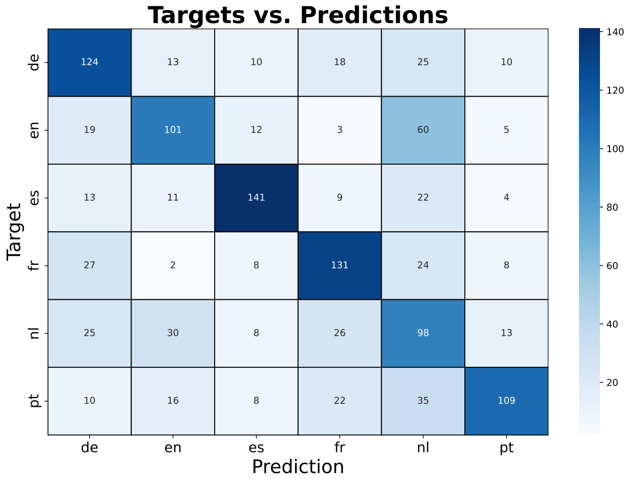

Figure 2: Heatmap visualizing the count of predictions for each label for the best-performing model

Conclusions

Our model obtained a final accuracy on the test set of 0.61. This means that the model performs 4 times better than a random guess, which would yield an accuracy of 0.16. It also means that the layers and hyperparameters were correctly selected. For the error analysis, we are not sure why Dutch was more frequently falsely predicted to be the target in comparison to other target languages. It can be due to linguistic similarities to other Germanic languages as well as complex spellings and pronunciations, making it harder for the classifier to correctly match them with their language category.

Work distribution

- Bram - data normalization, writing the report, and making the architecture design visualization
- Kasia - developing spectrogram functions, writing the report, and creating tables
- Laurynas - adjusting the binary model to 6 classes, experimenting with additional dropout and batch normalization layers, hyperparameter tuning, documenting the results
- Nin - building the binary classification model and the training function, binary model selection

Appendix

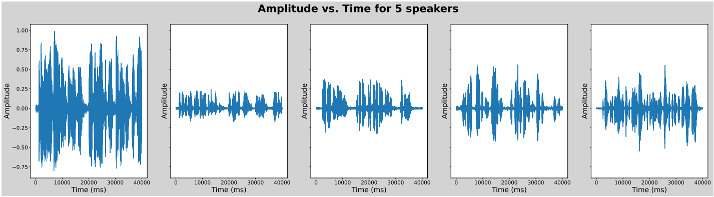

Figure 3: Figure depicting 5 random samples within the original training data. The horizontal axis represents the time in milliseconds while the vertical axis corresponds to the amplitude

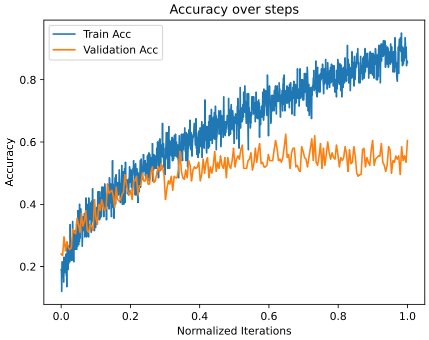

Figure 4: Figure visualizing the development of the training and validation accuracy over all the epochs for the multiclass model using spectrograms

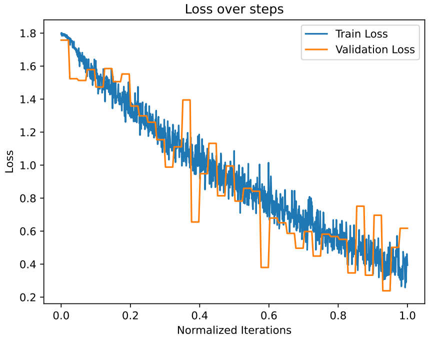

Figure 5: Figure visualizing the development of the training and validation loss over all the epochs for the multiclass model using spectrograms