Speaker-Independent Spoken Digit Recognition

Enosh Peter Ponraj
7012171
enpe00001@
stud.uni-saarland.de

Priya George
7043712
prge00001@
stud.uni-saarland.de

Zenith Biswas
7010713
zebi00001@
stud.uni-saarland.de

Abstract

In this project, we focus on developing a SDR system in a speaker-independent setting. That is, the speakers in the evaluation set are disjoint from the training set speakers. We do so because we expect real-world ASR systems to generalize to different speakers than those we have data for. Moreover, for many languages that are under-resourced, we have have (limited) annotated speech data from a single speaker, but we would still want the system to be deployed to work on any speaker of that language. We tackle the problem of spoken digit recognition as a sequence classification task. Concretely, the inputs are short audio clips of a specific digit (in the range 0-9), then the goal is to build deep neural network models to classify a short audio clip and predict the digit that was spoken.

1 Introduction

Speaker-independent spoken digit recognition is a challenging task in the field of speech processing, where the goal is to recognize the digits spoken by different speakers without any prior knowledge of the speaker.

The code for this project is available on Github, which follows a structured approach to implementing the models and evaluating their performance using the evaluation metrics mentioned above. By comparing the performance of the two models, we can determine which model is better suited for the speaker-independent spoken digit recognition task based on the mel-spectrogram input.

The main elements of the project are explained below and the usage can differ for different tasks:

Input: Input Audio data *.wav* which are mapped based on the speaker.
Output: Classified Number.
Evaluation Metric: The Following evaluation metrics are used:
Precision: Precision is a measure of the model's ability to correctly identify positive samples, i.e., the proportion of true positive predictions out of all positive predictions.
Recall: Recall, also known as sensitivity, is a measure of the model's ability to correctly identify all positive samples, i.e., the proportion of true positive predictions out of all actual positive samples.
F1-score: F1 score is a weighted harmonic mean of precision and recall, which takes both metrics into account. It is a commonly used metric in machine learning for imbalanced datasets, where the number of positive and negative samples is not equal.
Confusion Matrix: The confusion matrix is a table that summarizes the true and predicted labels for a given classification task.

2 Task 1: Baseline

The goal of Task 1 was to create a baseline model for speech classification and evaluate its performance using various evaluation metrics.
Observation:
The evaluation of the linear classifier for speech classification shows that the overall accuracy of the model is 40 percentage on the test data, which suggests that the model does not perform well in correctly classifying the input speech signals. From the classification report, we can observe that the model has the highest precision for class 1 (89 percent) and the highest recall for class 0 (79 percent). This suggests that the model is relatively better at identifying class 1 than other classes, but it struggles to identify class 2, 3, 5, and 6, with low

precision and recall scores for these classes.
We can see that the F1-score is the highest for class 8 (0.46) and the lowest for class 2 (0.14). This suggests that the model performs relatively better in classifying class 8 and struggles to classify class 2. The macro-average F1-score (0.38) and weighted-average F1-score (0.39) are lower than the accuracy, indicating that the model has difficulty in classifying some classes, and the overall performance of the model is not balanced across all classes.

3 Task 2: Neural Network Models

In Task 2, we focus on creating a better model than the given baseline. The Goal is to have more accuracy than the baseline and to calculate the t-sne to analyze how the different models seperate the different classes.

CNN was the first choice as it relatively easy to train and fine-tune. Audio transformers are a good choice for spoken digit recognition when dealing with large datasets.

We concluded to go ahead with LSTM-RNN and Deep CNN. LSTM-RNN was chosen as the second choice because it seemed more challenging and as a group we were ready for challenges in LSTM-RNN.

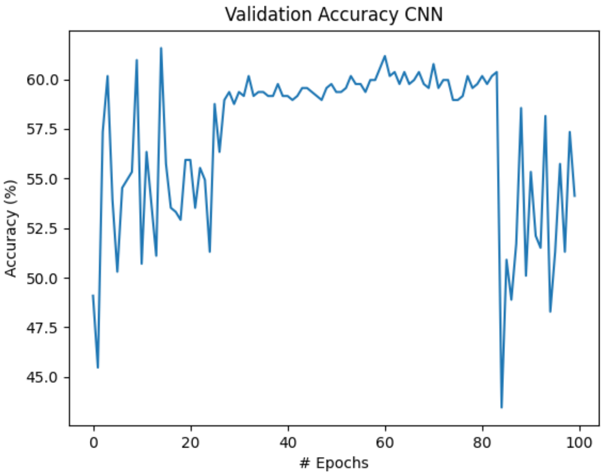

Figure 1: Accuracy of CNN model on validation set for 100 epochs.

CNN Observation:

The CNN model for spoken digit recognition has an overall test accuracy of 0.59, which indicates that it correctly classified 296 out of 503 spoken digit samples. The confusion matrix shows the number of true and false positive and negative predictions for each digit class.

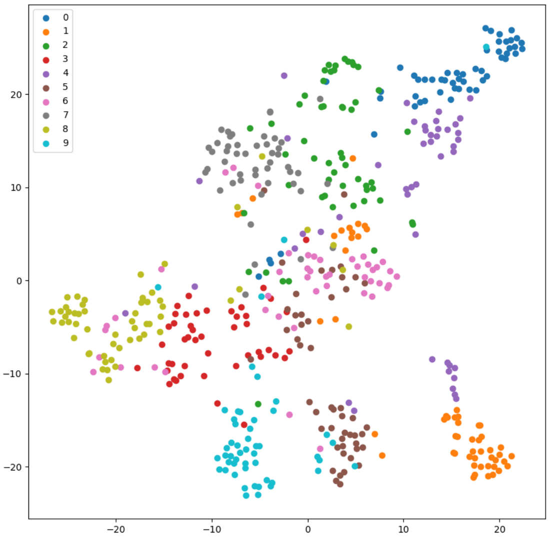

Figure 2: T-SNE evaluation of the last layer of CNN model without non-linearity.

The highest precision was achieved for digit 1, with .88, meaning that 88 percent of the predicted 1's were actually 1's. The highest recall was achieved for digit 7, with .91, indicating that 91percent of the actual 7's were correctly identified as such. The lowest precision and recall were both achieved for digit 6, with .32 and .63, respectively, suggesting that the model struggled to correctly classify this digit.

The f1-score, which is a harmonic mean of precision and recall, ranges from 0.35 to 0.80, indicating that the model performs well for some classes and poorly for others. The weighted average f1-score of 0.58 indicates that the model's performance is generally moderate.

RNN-LSTM Observation:

Based on the confusion matrix, the LSTM-RNN model achieved an accuracy of .65 on the test set. The model performed relatively well in classifying digits 0, 1, and 2, achieving precision and recall scores above 0.7. However, it struggled with classifying digits 3 and 4, achieving low recall scores of 0.57 and 0.33, respectively.

Compared to the CNN model, which achieved an accuracy of .58 on the same task, the CNN model appears to be much less accurate. The CNN model also achieved much higher precision and re-

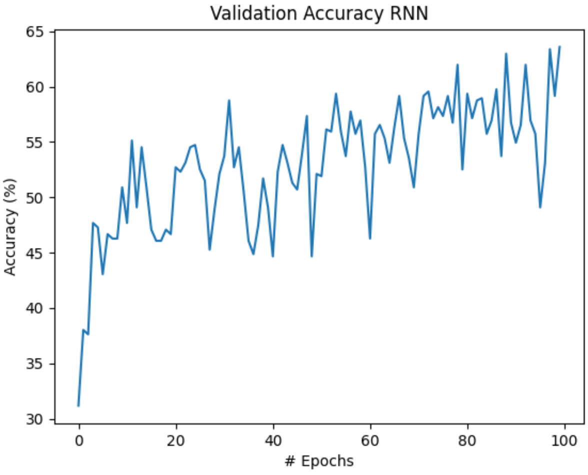

Figure 3: Accuracy of LSTM model on validation set for 100 epochs.

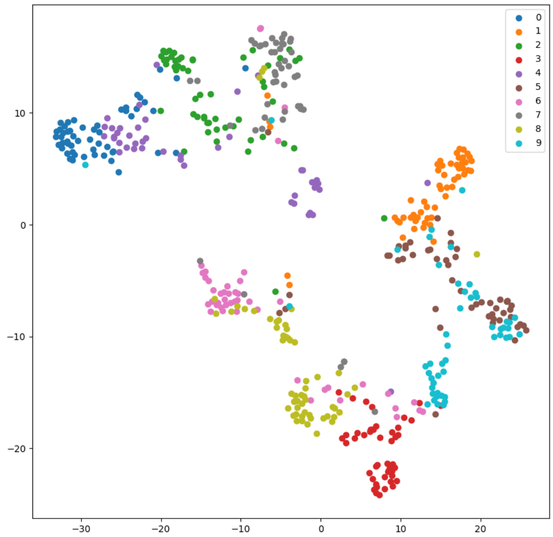

Figure 4: T-SNE evaluation of the last layer of LSTM model without non-linearity.

call scores for some class, indicating that it was able to classify digits with much higher accuracy. It is possible that the CNN model may not have been well-suited to this particular task, or that it required more training or tuning to achieve better performance.

Considering the T-SNE evaluation plots, RNN forms better clusters for all groups than CNN model.

4 Task 3: Data Augmentation

The Goal of task 3 is to use single speaker for training and it is then evaluated using various speakers.The further parts of this task include using data augmentation methods for extending the training set and the use of Contrastive loss based on different views of the training samples.

Single Speaker without data augmentation: The confusion matrix shows the number of predicted classes versus the actual classes for the test set. The accuracy for the test set is 0.4433, and the precision, recall, and F1-score are reported for each class. The macro and weighted average of the precision, recall, and F1-score are also reported. The model's performance is moderate, with some classes having high precision and recall, while others have low precision and recall. There is room for improvement in the model's performance.

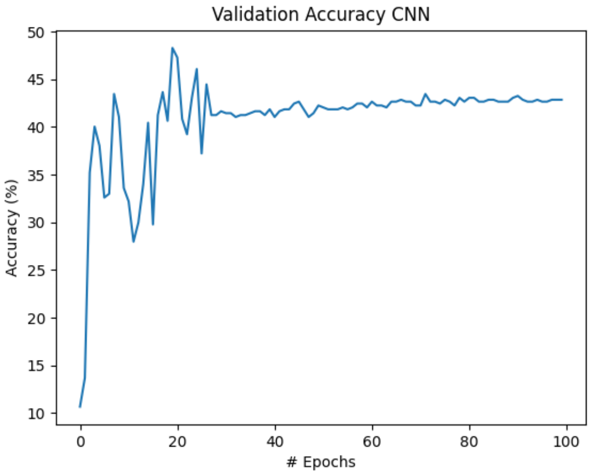

Figure 5: CNN Accuracy trained using single speaker without data augmentation.

We use the following two data augmentation techniques:

- *Pitch Shifting:* It is a technique by altering the pitch of audio samples to simulate different speakers or to create variations of the same speaker.

- *Frequency Masking:* This can be used in data augmentation for speech recognition by selectively masking or attenuating certain frequency ranges in audio samples to simulate different environments or background noise conditions.

Single Speaker with data augmentation:

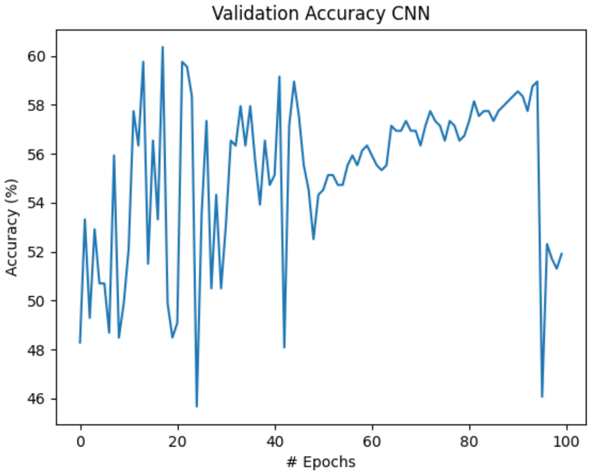

Figure 6: CNN Accuracy trained using single speaker with data augmentation.

The evaluation metrics suggest that the LSTM model with a single speaker did not perform very well on the test set, achieving an accuracy of 61.75 percent. The confusion matrix shows that the model had difficulty correctly classifying samples across all classes, with particularly poor performance on classes 2, 4, and 5. The precision, recall, and F1-score for each class were also generally low, with some classes having values as low as 0.14 for recall and 0.21 for F1-score. The macro-average of the metrics across all classes was also relatively low at 0.39.

Single Speaker with data augmentation and Contrastive Loss:
Looking at the confusion matrix, we can see that the model seems to be struggling with several classes, with low precision and recall values, such as classes 1, 2, 3, and 4. On the other hand, the model seems to perform relatively well on classes 0, 5, 7, and 8, with higher precision and recall values. It gave very bad accuracy but it can be improved by having a loss that bottles up the loss for all classes as well.

Conclusion

Overall, the LSTM-RNN model appears to be less accurate and less precise than the CNN model for this task. However, it is important to note that different models may perform better or worse on different tasks, and that the choice of model may depend on the specific requirements of the task at hand. WIth the Contrastive Loss it will be better if there was a combination of Cross enropy as there are multiple number of classes

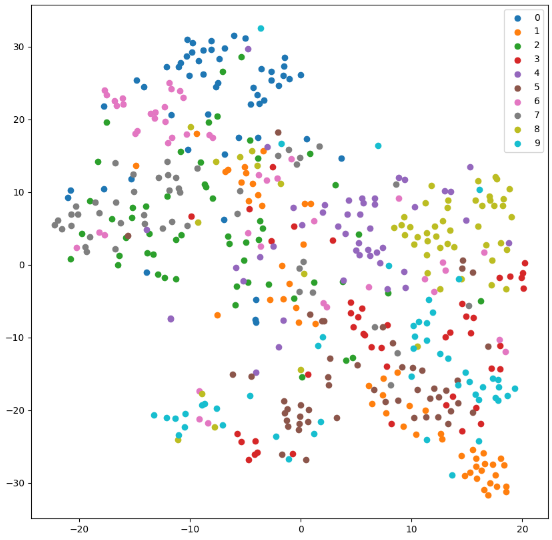

Figure 7: CNN T-SNE trained using single speaker with data augmentation and contrastive loss.

References

contrastive-loss-function-in-pytorch