Mandy Glasbeek (2056467), Wiebke Koopman (2002952), Nikhil Shukla (2025129), Carlijn van Rooij (u1273571) & Kristel van Rooij (u1273632) - Group: 2 - Codalab account: group_2

Siamese network for re-identification of voices

For the re-identification task for voices we started with building a Convolutional Neural Network (CNN). After diving into the literature we decided to incorporate CNN into a Siamese Neural Network (SNN). This model is very suitable for tasks that compare two inputs with each other on similarity (Chicco, 2021), in this case voice recordings from various persons. Our SNN consists of three CNN layers with a ReLU activation, batch normalization and dropout rate. The dropout rate was used as a regularization measure in order to prevent overfitting and the batch normalization was used to stabilize the training process (Santurkar, Tsipras, Ilyas, & Madry, 2018). Subsequently, three linear layers with two ReLU activation functions were used. A contrastive loss is applied since this function is suitable to compare two inputs. The architecture of our model is presented in Figure 1. We used Jupyter Notebook in Anaconda for your reproducibility.

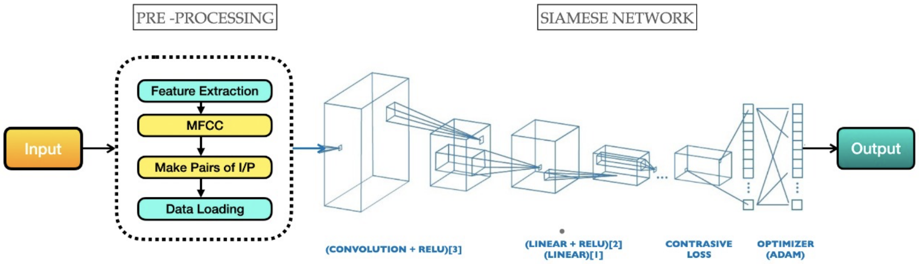

Figure 1. Diagram of the architecture

To optimize our SNN we tuned some hyperparameters. Specifically, the kernel size (3, 4), dropout rate (0.2, 0.4), number of epochs (50, 70, 100, 150), learning rate (0.01, 0.001, 0.00146, 0.0001) and batch size (32, 64, 128, 256) were tuned. The network was trained with the commonly used Adam optimizer. Below, Table 1 with the results of the validation set is shown. The best performing models (kernel size = 3, dropout rate = 0.2) with different values of number of epochs, learning rate and batch size, respectively, are included. Next to that Figure 2 is showing the training loss over epochs.

Model	Top 1 score	Top 3 score	Top 5 score	Top 10 score	Accuracy
(100, 0.0001, 32)	0.1552	0.3385	0.4708	0.6750	0.1453
(100, 0.0001, 64)	0.1958	0.3917	0.5135	0.6938	0.1506
(150, 0.0001, 64)	0.1906	0.4042	0.5156	0.6823	0.1531

Table 1. Results of the SNN

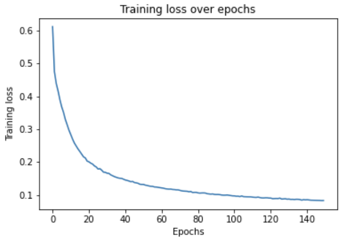

Figure 2. Training loss over epochs

Discussion and conclusion

- The accuracy of our architecture is 0.6167 on the test set. We think the performance can be improved with a few adjustments.
- First, we could extract different and more features from the voice recordings (for example zero-crossing rate).
- Second, the network architecture Recurrent Neural Network (RNN) or a CNN-RNN may be more suitable for this task.

References

References from code script

Facial similarity with siamese networks in Pytorch (2021). Retrieved from https://github.com/harveyslash/Facial-Similarity-with-Siamese-Networks-in-Pytorch/blob/master/Siamese-networks-medium.ipynb

A very simple siamese network in pytorch (2019). Retrieved from https://www.kaggle.com/jiangstein/a-very-simple-siamese-network-in-pytorch

Rosebrock A. (2020). Building image pairs for siamese networks with Python. Retrieved from https://www.pyimagesearch.com/2020/11/23/building-image-pairs-for-siamese-networks-with-python/

References from report

Chicco, D. (2021). Siamese neural networks: An overview. *Artificial Neural Networks*, 73-94.

Santurkar, S., Tsipras, D., Ilyas, A., & Madry, A. (2018). How does batch normalization help optimization?. Paper presented at the 32nd Conference on Neural Information Processing Systems. Retrieved from https://arxiv.org/pdf/1805.11604.pdf