Lip Reading

RAICHU

Demet EROL
161180030
Department of Engineering,
Computer Engineering, Gazi University
Ankara, Turkey
demetteroll@gmail.com

Müzeyyennur YILGIN
161180067
Department of Engineering,
Computer Engineering, Gazi University
Ankara, Turkey
muzeyyenyilgin@gmail.com

ABSTRACT

Lip reading is becoming increasingly important in today's world. Using the Miracl-VC1 dataset, words were determined from images in this study. The machine learning algorithms KNN, Decision Tree, SVM, Random Forest, and CNN, a deep learning model, were compared. The best results were obtained by using the Random Forest and CNN algorithms.

Keywords

Lip reading; Machine learning; Deep learning; CNN; KNN; SVM; Decision Tree; Random Forest.

1. INTRODUCTION

The World Health Organization estimates that there are over 460 million deaf people in the world. These people need sign language to communicate in society. However, because sign language is not widely used, communication is difficult. To address this issue, various solutions such as hearing aids and speech-to-text translation applications have been developed.Since external sounds have a negative impact on speech perception in these systems, adequate reliability cannot be attained in real life. Lip reading, on the other hand, uses visual data to eliminate noise-induced issues [1].

Lip reading is the identification of spoken words as well as the study of lip formation. Lip reading has become one of the artificial intelligence experiments as technology has advanced. [2] Areas of use include Aviation, military systems, defense, healthcare, sports refereeing, and the translation of historical documentaries and films [3].

Sindhura et al. suggested a lip reading paradigm related to AlexNet and Inception V3 Convolutional Neural Networks in an article published in 2018. The Miracl-VC1 dataset was used. In this analysis, model efficiency was tested both with and without the speaker. According to the findings of the trials, AlexNet had an accuracy of 86.6 percent and Inception V3 had an accuracy of 64.6 percent [4].

Lip reading models with DNN and LSTM were developed and compared in a 2019 article published by Kirange and Kulkarni. LSTM outperformed DNN in the dataset with Urdu numbers for ten speakers [5].

Many methods for attribute extraction for lip reading were attempted and compared in the article published by Morade and Patnaik in 2015. Support Vector Machine (SVM), Back Propagation Neural Network (BPNN), K-Nearest Neighborhood (KNN), Random Forest Method (RFM), and Naive Bayes (NB) training times and results were compared. The CUAVE and Tulips databases were used. As a consequence of the experiment, it was discovered that SVM worked best for the CUAVE database and needed less training time than other approaches. In Tulips results, BPNN performed the best [6].

Frank et al. suggested a solution to the speaker-independent lip reading dilemma in 2018. When the success rates of classifiers such as CNN, SVM, Random Forest, and LSTM were measured, it was determined that SVM produced the best performance [3].

2. TOOLS & TECHNOLOGIES

The project was created on Colab Notebook using the Python programming language. Cv2, Pandas, Numpy, Os, Dlib, Keras, and Sklearn libraries were included. Lips were observed in the image using the "shape predictor 68 face landmarks.dat" model for face detection. Pycharm and Qt were used for the created interface.

3. THE APPROACH

The words "begin" ,"choose," and "connection" were identified as targets in a dataset in which 5 female and 5 male speakers said each word 10 times [7]. The Dlib library was used to detect the faces in the images. The lip is represented by points between 49 and 68 in the “shape predictor 68 face landmarks.dat” model. Lip images were created using this model, resized, and stored in a separate folder.

With the pixels from the cut lip images, speaker ids, and targets, a dataframe was created. MinMaxScaler was used to normalize all features except the target. The data was split into two parts: 80 percent train and 20 percent test. Since they are excellent at classification, models were generated as the KNN, Decision Tree, SVM, Random Forest, CNN and compared.

3.1 KNN

N neighbors parameter was set between 3 and 10 then The accuracy was analyzed for both test and train data. As seen in Figure 3.1, the best result was obtained with four neighbors. According to the data from the confusion matrix, the words "choose" and "connection" were often confusing. Figure 3.2 depicts the classification report.

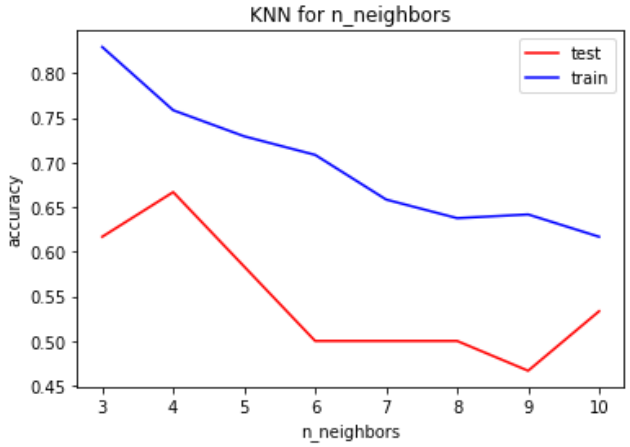

Figure 3.1 Train and test accuracy for KNN

	precision	recall	f1-score	support
0	0.588	0.526	0.556	19
1	0.500	0.455	0.476	22
2	0.435	0.526	0.476	19

Figure 3.2 Classification report for KNN

3.2 Decision Tree

The accuracy, while the max depth parameter was in the 1-14 range, was evaluated for test and train results. As shown in Figure 3.3, After the max depth was crossed 7, the test accuracy remained constant after declining, while the train accuracy reached 1.0, indicating overfitting. Figure 3.4 depicts the classification report.

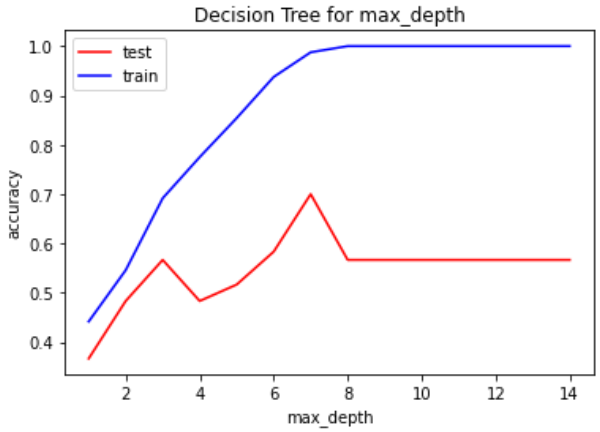

Figure 3.3. Train and test accuracy for Decision Tree

	precision	recall	f1-score	support
0	0.625	0.526	0.571	19
1	0.600	0.545	0.571	22
2	0.500	0.632	0.558	19

Figure 3.4 Classification report for Decision Tree

3.3 SVM

The SVM algorithm was tested on both train and test data. Although the terms "connection" and "choose" cause the most uncertainty, as seen in Figure 3.5, train and test accuracy are very similar. Figure 3.6 depicts the classification report.

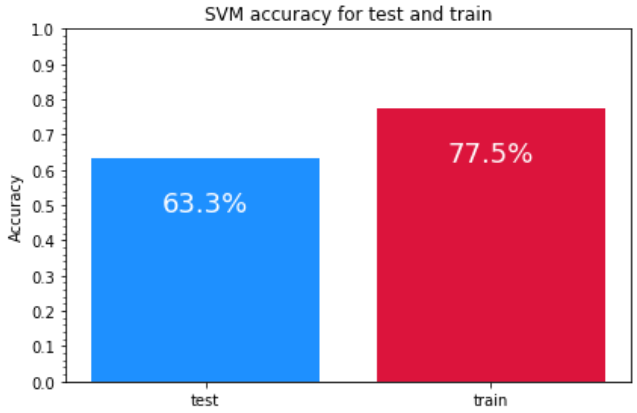

Figure 3.5 Train and test accuracy for SVM

	precision	recall	f1-score	support
0	0.750	0.789	0.769	19
1	0.632	0.545	0.585	22
2	0.524	0.579	0.550	19

Figure 3.6 Classification report for SVM

3.4 Random Forest

While the max_depth parameter has a range of 1-14, accuracy has been analyzed for both test and train results. While the max_depth was 10, the test accuracy was 0.92, as seen in Figure 3.7. Figure 3.8 depicts the classification report.

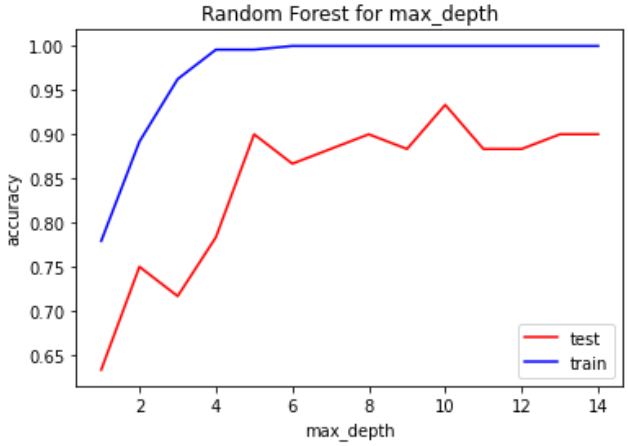

Figure 3.7. Train and test accuracy for Random Forest

	precision	recall	f1-score	support
0	1.000	0.895	0.944	19
1	0.909	0.909	0.909	22
2	0.810	0.895	0.850	19

Figure 3.8 Classification report for Random Forest

3.5 CNN

CNN is a deep learning algorithm consisting of different trainable layers. These layers are Convolution Layer, Non-linearity Layer, Pooling Layer and single or multiple fully connected layers. CNN architecture is shown in Figure 3.9. [8]

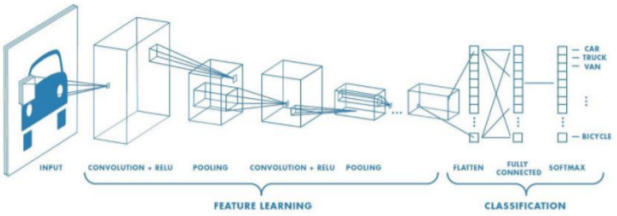

Figure 3.9 CNN architecture [9]

The Convolutional layer consists of many learnable filters. The feature map is created by hovering the filters over the entire image. By changing the filter coefficients, important regions on the image are determined. Output size may be reduced due to filter sizes. Padding can be applied to prevent this. Padding means adding a frame of zeros [8,10].

The network has a linear structure due to some mathematical operations in the convolution layer. The Non-linearity Layer is used to put the deep network into a non-linear structure. It aims to get rid of negative values so that the model can learn better. Generally, Rectified Linear Units (ReLU), sigmoid, and tanh functions are used. However, ReLU is more desirable because it is more successful [8,10].

The pooling layer is used to reduce the width and height of the data. The fact that this layer causes data reduction prevents overfitting. It also reduces the computational load for the next layer. It is traversed on the image by the pool size specified and returns a single value at a time, depending on its type. Max pooling and Average Pooling are widely used [8,11]

After these layers are repeated as many times as desired, the fully connected layer is found. The image in matrix form is turned into a flat vector and the learning process begins. It is done by taking all the neurons in the preceding layer and linking them to every neuron in the present layer to produce global semantic information. This step is completed by updating the weight and bias values. If the amount of data is small or the network is very large, the Dropout layer can be used. It prevents the network from memorizing by removing some nodes. After this stage, output is produced with various classifiers. Softmax is preferred because it is generally successful [8,11].

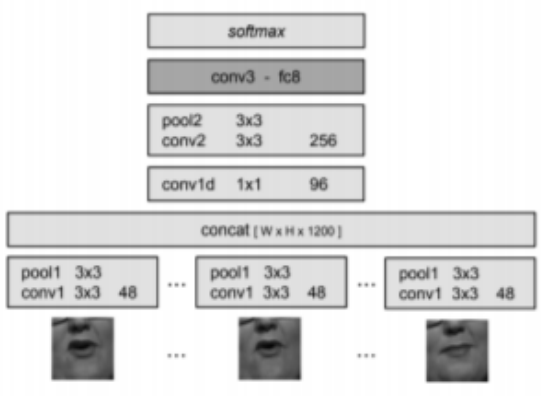

Figure 3.10 CNN systems [12]

In the project, the dataset was split into 70 percent train, 24 percent validation, and 6 percent test before CNN was used to create a model. The video data consists of the keras library's 5D tensor (samples, frames, height, width, color depth). The model was created in the manner depicted in Figure 3.11.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv3d (Conv3D)              (None, 18, 98, 98, 64)    1792
_________________________________________________________________
max_pooling3d (MaxPooling3D) (None, 9, 49, 49, 64)     0
_________________________________________________________________
conv3d_1 (Conv3D)            (None, 7, 47, 47, 128)    221312
_________________________________________________________________
max_pooling3d_1 (MaxPooling3 (None, 3, 23, 23, 128)    0
_________________________________________________________________
conv3d_2 (Conv3D)            (None, 2, 22, 22, 256)    262400
_________________________________________________________________
max_pooling3d_2 (MaxPooling3 (None, 1, 11, 11, 256)    0
_________________________________________________________________
flatten (Flatten)            (None, 30976)             0
_________________________________________________________________
dense (Dense)                (None, 4096)              126881792
_________________________________________________________________
dropout (Dropout)            (None, 4096)              0
_________________________________________________________________
dense_1 (Dense)              (None, 2048)              8390656
_________________________________________________________________
dropout_1 (Dropout)          (None, 2048)              0
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 6147
=================================================================
Total params: 135,764,099
Trainable params: 135,764,099
Non-trainable params: 0
```

Figure 3.11. Model summary

Adam was used as the optimizer in the model, and categorical_crossentropy was used to calculate the loss. Epoch 20 and batch size 21 was assigned. After 20 epochs, the train's loss value was 0.08 and the validation's loss value was 0.27. Figure 3.10 depicts the epoch-related accuracy values for train and validation.

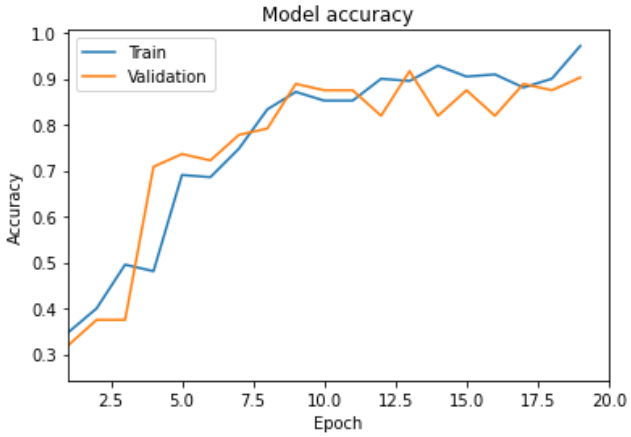

Figure 3.12 Train and validation accuracy for CNN

The created model was saved with the save function of the Keras library. The interface seen in Figure 3.13 was designed with Pycharm and Qt Designer. Lips were detected with "shape predictor 68 face landmarks.dat" on video images taken from the user using OpenCV. After pressing the start button, the video recording was started and the user was expected to say one of the words "begin", "choose" and "connection". After pressing the stop button, one of every 5 frames was taken from the video recording and preprocessing steps such as resize and scale were performed. After the test data created was given to the model, the prediction made was transferred to the user as text with QLabel.

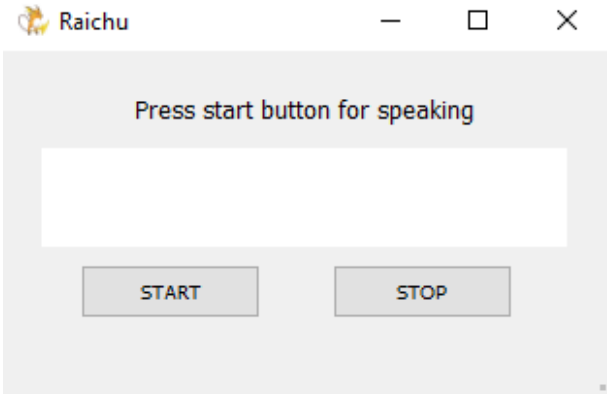

Figure 3.13 GUI

3.6 Results

Figure 3.14 depicts the best performance of the algorithms for test data. The Random Forest algorithm and CNN generated the best results. While the term "connection" is widely misunderstood, it was found at the most basic level in the Random Forest algorithm and CNN.

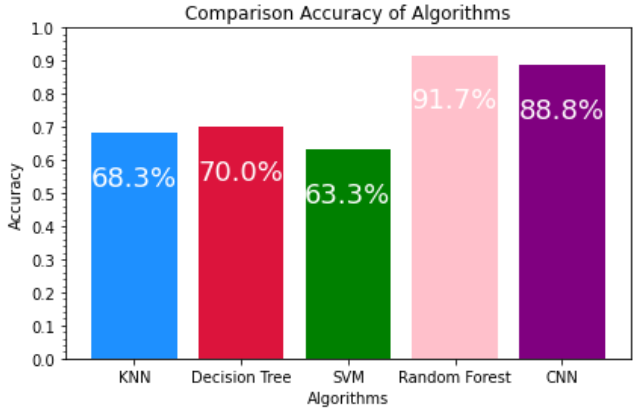

Figure 3.14 Comparison Results

In the created interface, the model was tested with speaker-independent data. The test results were transferred to the user as in Figure 3.15. The results showed successful predictions were made, although the words "connection" and "choose" were sometimes confused.

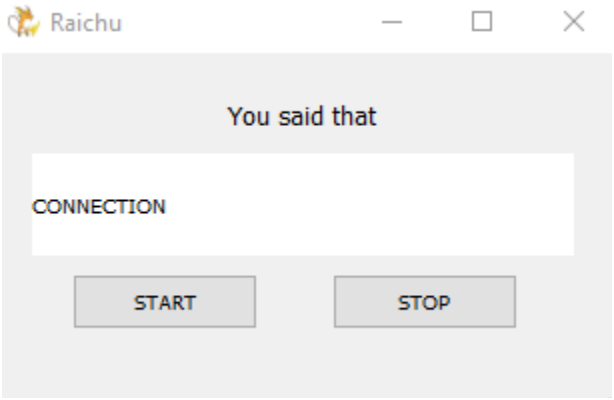

Figure 3.15 Prediction result

4. CONCLUSION

Lip reading has been a significant need in a variety of fields today. In this application created with the words "begin", "choose", and "connection" from the Miracl-VC1 dataset. CNN, KNN, SVM, Decision Tree, and Random Forest were used. Random Forest, with 91.7 percent accuracy, and CNN, with 88.8 percent accuracy, are the highest. In the next step was created a gui to validate the CNN model with data that is independent of the speaker.

5. REFERENCES

[1] Abrar, M.A., Islam, A. N. M. N., Hassan, M. M., Islam, M. T., Shahnaz C. and Fattah, SA., "Deep Lip Reading-A Deep Learning Based Lip-Reading Software for the Hearing Impaired", *2019 IEEE R10 Humanitarian Technology Conference*, 2019, 40-44.

[2] Khan, S., Azmi, H & Nair, A. and Mirza, H., "Implication and Utilization of various Lip Reading Techniques", *International Journal of Computer Applications*, 2017, 167, 25-27.

[3] Burton, J., Frank, D., Saleh, M., Navab, N. and Bear, H. L., "The speaker-independent lipreading play-off; a survey of lipreading machines", *2018 IEEE International Conference*

on Image Processing, Applications and Systems (IPAS), 2018, 125-130.

[4] Sindhura, P., Preethi, S. J. and Niranjana, K. B., "Convolutional Neural Networks for Predicting Words: A Lip-Reading System," 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques, 2018, 929- 933.

[5] Kulkarni, A. H. and Kirange, D., "Artificial Intelligence: A Survey on Lip-Reading Techniques," *2019 10th International Conference on Computing, Communication and Networking Technologies*, 2019, 1-5.

[6] Morade, S.S. and Patnaik, S., "Comparison of classifiers for lip reading with CUAVE and TULIPS database", Optik, 126(24), 2015, 5753-5761.

[7] https://sites.google.com/site/achrafbenhamadou/-datasets/miracl-vc1

[8] Fattah, P., Salihi, N. K., Rashid, T. A. and Shamsaldin, A. S. (2019), The Study of The Convolutional Neural Networks Applications, *UKH Journal of Science and Engineering*, 31-40.

[9] Das, A., Convolution Neural Network for Image Processing — Using Keras, https://towardsdatascience.com/convolution-neural-network-for-image-processing-usingkeras-dc3429056306, Last Access Date: 03.06.2021

[10] Doğan, Ö., CNN (Convolutional Neural Networks), https://teknoloji.org/cnnconvolutional-neural-networks-nedir/, Last Access Date: 02.06.2021

[11] Ülker, E., İnik, Ö., (2017), Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri, *Gaziosmanpasa Journal of Scientific Research*, 6, 1-20.

[12] Chung J.S., Zisserman A. (2017) Lip Reading in the Wild. In: Lai SH., Lepetit V., Nishino K., Sato Y. (eds) *Computer Vision* – ACCV 2016.