


Facial Paralysis Recognition Using Face Mesh Based Learning

Zeerak Mohammad Baig Name¹ ^a

¹*Academy of Computer Science and Software Engineering, University of Johannesburg, Corner Kingsway and University Road, Johannesburg, South Africa
zmbaig98@gmail.com*

Keywords: Facial Paralysis, Machine Learning, Support Vector Machine, XGBoost, K Nearest Neighbour, CNN, MobileNetV2, Face Mesh

Abstract: Facial paralysis is a medical disorder caused by a compressed or enlarged seventh cranial nerve. The facial muscles become weak or paralyzed because of the compression. Many medical experts believe that viral infection is the most common cause of facial paralysis; nevertheless, the origin of nerve injury is unknown. Facial paralysis hampers a patient's ability to blink, swallow, or communicate. This article proposes a deep learning-based approach and traditional machine learning-based approaches for facial paralysis recognition in facial images; such systems can aid in developing standardized medical evaluation tools. The proposed method has three components; the first detects faces or faces in each image, and the second extract a face-mesh from the given image using Google's media pipe. The face mesh descriptors are then transformed into the proposed face mesh image that is then fed into the final component, comprised of a convolutional neural network used to perform overall predictions. The study uses YouTube facial paralysis datasets (Youtube and Stroke face) and control datasets (CK+ and TUFTS face) to train and test the model for unhealthy patients. The best approach achieved an accuracy of 98.93% with a MobilenetV2 backbone using the YouTube facial paralysis dataset and Stroke face dataset for palsy images

1 INTRODUCTION

Facial paralysis or facial palsy is a condition whereby one cannot move the facial muscles of the face on one or both sides. This medical condition can result from nerve damage due to diseases such as brain tumours or Stroke and trauma (Parra-Dominguez et al., 2021). Suppose the early detection of facial palsy and treatment is delayed. In that case, it can result in many complications, which include damage to the seventh cranial nerve and excessive dryness in the eye, which may lead to eye infections, ulcers and even loss of vision. Furthermore, one may develop synkinesis, a condition in which a movement of one face part causes an unintentional movement of another face part (Tiemstra and Khatkhate, 2007).


Facial paralysis is a well-known medical condition that needs to be detected and treated early. Developing methods that can assist doctors in detecting facial palsy earlier can add a fair amount of value to the detection and treatment. These methods can also serve as the basis for forming standardised tools for medical assessments, treatment, and monitoring.

Our contributions presented in the study includes face mesh-based learning for facial paralysis recognition. The study also looks at variations of face mesh transformation to measure their impact on accuracy in the deep learning model. The study will present a detailed comparative study for both a traditional baseline approach and the proposed deep learning method.

The following sections will discuss the problem background and the methods used to tackle such problems. The article will then discuss the methods used for facial paralysis recognition in a facial image using the two approaches and their results, followed by an ablation study and conclusion

2 PROBLEM BACKGROUND

A delay in detecting and treating facial paralysis might result in a slew of complications. As previously said, detecting facial palsy is critical in determining the degree of face muscle and nerve dysfunction. It is critical to treat facial nerve palsy as soon as possible. This is because, as time passes after the onset

^a  <https://orcid.org/0000-0000-0000-0000>

of symptoms, nerve damage worsens and the rate of healing slows, raising the risk of complications. The nerve is affected for 3 to 7 days when the face nerve gets irritated. It is critical to remove the inflammation that has occurred in the nerve and prevent the progression of paralysis for the first week following the commencement of the disease (Hato et al., 2003).

The study of facial indicators has sparked a flurry of studies on automated facial nerve function evaluation based on biomedical visual capture of the face, particularly in the field of computer vision: traditional photos and video capture the face, as well as infrared (thermal imaging) and depth images (Hassaballah and Hosny, 2019). A feature extraction technique is carried out by a few image-based algorithms, which entails detecting the face region in the image and then extracting crucial points based on a specified model. It's also worth noting that several publicly available shape predictors use haar cascades to extract face features and bespoke feature extractors that provide decent results. The extraction of key points is critical since it is utilized to compute distances and angles between landmarks later (Boyko et al., 2018).

2.1 Existing Works and Solutions

Before doing face analysis, some works employ facial landmarks detection (i.e., facial keypoint extraction). Other studies treat facial paralysis as a binary classification issue (Guarin et al., 2018) (Wang et al., 2016) (Jiang et al., 2020). A similar technique is used by Dominguez (Parra-Dominguez et al., 2021). The method uses a shape predictor to extract various facial landmarks initially. The distances between different facial landmarks are then used to compute facial measures, and finally, a multilayer perceptron-based classifier is used for classification. Another method by Kim et al. offered a smartphone-based autonomous diagnostic system with three components: a facial landmark detector, a feature extractor based on facial regions, and a classifier (Kim et al., 2015). Huang et al. proposed utilizing a standard camera to identify facial palsy using deep learning. They framed facial palsy detection as an object detection issue, with the target objects being the deformation areas caused by facial palsy or simply the palsy regions on a patient's face. Face detection, facial landmark detection, and local palsy area identification are the three components of their suggested method, which is a hierarchical network. In their private database, the authors reported a prediction accuracy of 93% (Jison Hsu et al., 2018).

Another study presented a two-stage technique for classifying facial paralysis: first, distinguishing

healthy from unhealthy participants and classifying facial palsy among unhealthy people. It measured symmetry using four facial expressions: at rest, lifting the eyebrows, screwing up the nose, and smiling. The system used rule-based and machine-learning techniques to create a categorization model (hybrid classifier). In their private database, the authors reported a sensitivity of 98.12% in discriminating between healthy and unhealthy people (Barbosa et al., 2019). Based on the information above, we can see how much attention facial paralysis detection has received in the scientific community. As a result, alternative machine learning algorithms should be explored to detect facial paralysis in a picture with more accuracy.

3 EXPERIMENT SETUP

The goal of this study is to identify facial paralysis using a variety of face photographs. The first approach with traditional machine learning techniques examines the symmetry of the face to detect whether a particular image of a face is affected by facial paralysis or not. The second approach uses a face mesh and a convolutional neural network for facial paralysis recognition in a given facial image.

3.1 Datasets

For this project, we used four different publicly available datasets, two containing images of healthy patients, whereas the other two datasets consisted of pictures of unhealthy patients.

YouTube facial paralysis database (YFP) gathers facial images of subjects suffering from facial paralysis. The dataset contains 32 videos of 21 patients, with a few cases having several recordings. These videos are converted into a 6FPS picture sequence since the shortest facial palsy session lasts a second (Jison Hsu et al., 2018). The facial droop and facial paralysis image dataset was also used, which contained 1024 images of unhealthy patients.

Tufts Face Database, which is the most complete, large-scale face dataset available, includes seven image modalities: visible, near-infrared, thermal, computerized sketch, LYTRO, recorded video, and 3D images are used to gather images of subjects who are considered healthy (Panetta et al., 2018). The tufts database contains approximately 100000 images of 112 participants. To enhance robustness against expression variation, the CK+ facial expression database was also used during training of our model.

It's worth noting that while all four of the databases aim to make information easier to find for the creation of therapeutic applications, they're not identical in terms of image quality, lighting, or posing circumstances, nor are the activities done by the participants. In other words, while neither database is directly equivalent to the other for our categorization challenge, they were both useful in the design process.

Data set was divided into training and testing sets where the training set had a total of 3958 images, with half being unhealthy subjects. The test set had a total of 864 images which were also divided equally among healthy and unhealthy patients. YouTube facial paralysis data set and Stroke face data set were used to train the model for unhealthy patients. For the training set of unhealthy subjects, the study used a total of 1979 images, out of which 1547 images belong to the YouTube facial paralysis database, and the rest of the pictures belong to the Stroke-face data set. The test set for unhealthy patients comprised only stroke face data set images. The training set for healthy patients used a combination of the Tufts face data set and CK+ data set. The training set for healthy images comprised 981 images from the CK+ data set, while the rest were taken from Tufts face data set. The testing set for the healthy patients contains 432 images from Tufts face data set.

3.2 Evaluation Metrics

The study will report several metrics to measure the accuracy of the classifiers. Precision and Recall are helpful metrics of prediction success when the classes are severely unbalanced. Precision measures result from relevancy in information retrieval, whereas Recall measures how many relevant results are returned (Davis and Goadrich, 2006). The precision-recall curve depicts the tradeoff between precision and recall for various thresholds. With high accuracy suggesting a low false-positive rate and high Recall indicating a low false-negative rate, a significant area under the curve means good Recall and precision. High scores imply that the classifier delivers accurate results and that most positive outcomes are positive.

4 METHODS

The structure of the study consists of two approaches for a detailed analysis of facial paralysis recognition. Both methods include facial detection, landmark extraction, feature extraction, and classification. The first approach uses the traditional machine learning

approach using various facial distance measures between landmarks, as depicted in Figure 1, to make classifications.

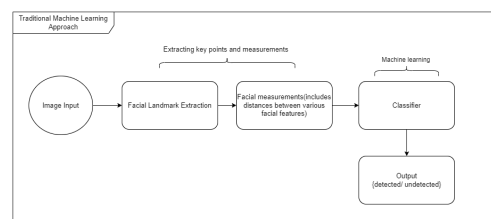


Figure 1: Traditional machine learning approach for facial paralysis recognition.



Figure 2: Deep learning approach for facial paralysis recognition.

The second approach is a deep learning approach to facial paralysis recognition. It uses mediapipe to generate a face mesh from a given facial image. The face mesh is generated using a model which focuses on semantically significant facial areas, predicting landmarks around the mouth, eyes, and irises more correctly at the cost of higher computational power. The input for this particular model is a 256 by 256 picture. Either the face detector or tracking from a previous frame provides this image. The model divides into numerous sub-models after obtaining a 64 64 feature map. All 478 face mesh landmarks are predicted by one submodel, which also produces crop boundaries for each region of interest. The remaining submodels use the matching 2424 feature maps created by the attention mechanism to forecast regional landmarks (Grishchenko et al., 2020). The generated mesh is then placed on a blank background and fed to a mobilenetV2 architecture for classification, as depicted in Figure 2.

4.1 Traditional Machine Learning Approach with Facial Distance measures

This approach uses traditional machine learning techniques where data pre-processing is done manually

before classification. This project implements four variations of the same method involving different kinds of classifiers. The feature extraction and facial measure component remain the same, whereas different classifiers predict whether the patient is healthy or not. These classifiers include a support vector machine, XGboost classifier, K Nearest Neighbor and an ensemble learning approach using the random forest classifier.

4.1.1 Facial Landmarks Extraction

The input image is initially converted to grayscale; after that, it is scaled down to 70% of its original size. The input image is also normalized before facial landmark extraction.

The facial landmark method begins by locating the face in a picture. The face detector is a method of detecting a human face in an image and delivering data in the form of bounding boxes or rectangle box values (Khan et al., 2019). We determine minor facial traits like brows, lips, and so on after detecting the face's position in a photograph. Facial landmark detection informs us of all the necessary elements of a human face.

Once the face has been detected in an image, the system uses the dlib's facial landmark detector to estimate the position of 68 coordinates (x, y) that map the facial points on a person's face. It's a landmark facial detector using pre-trained models (Wu et al., 2017). The extracted data is then stored for further processing.

4.1.2 Facial Distance Measures

Once the key points have been extracted from an image, we compute various distances between these key points. This approach evaluates the image intending to detect the symmetry level between the two sides of the face. Information from the brows, eyes, nose, and mouth is extracted in the suggested measurements. Twenty-one various distances were calculated using the facial key points. The multiple distances presented in Figure 3 allow us to compute the asymmetry level of a human face to categorize them into healthy and unhealthy subjects. The figure below shows the different facial distances and descriptions.

The proposed facial measures are used to compute the asymmetry level between the face's left and right sides of the face. The work of (Parra-Dominguez et al., 2021) inspires facial measurements from A to I. The classifier will use the percentage differences between the various facial measures to determine if a subject is healthy or unhealthy.

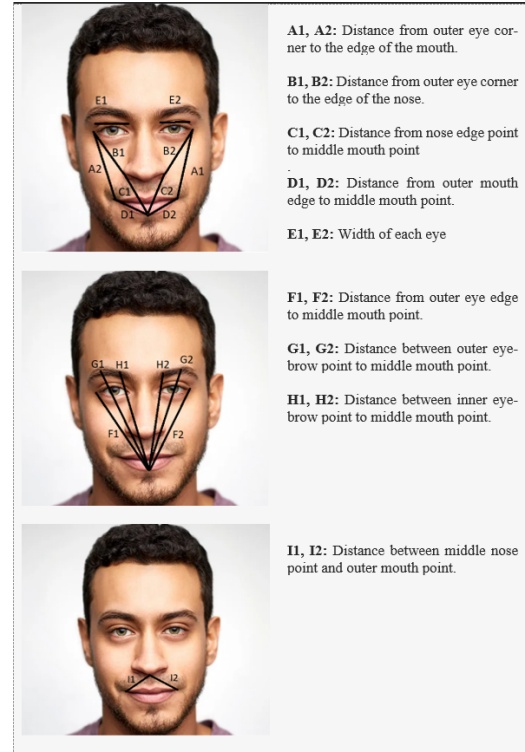


Figure 3: Distance measures between various facial landmarks.

Measure	Description
D1	Percentage difference between B1 and B2
D2	Percentage difference between C1 and C2
D3	Percentage difference between A1 and A2
D4	Percentage difference between D1 and D2
D5	Percentage difference between E1 and E2
D6	Percentage difference between I1 and I2
D7	Percentage difference between F1 and F2
D8	Percentage difference between H1 and H2

Table 1: Percentage distance measures between various facial landmarks

4.1.3 Classifiers

As mentioned previously, our first approach uses four types of classifiers to predict whether the subject falls under the healthy or unhealthy patient category. The first classifier is a support vector machine. The support vector machine algorithm aims to identify a hyperplane in an N-dimensional space that distinguishes between data points (Mgheed, 2021). A Linear kernel was used in this presentation along with the control error of 10.

The proposed method also uses an Xgboost learning algorithm. XGboost, also known as extreme gradient boosting, implements gradient boosted decision trees. Decision trees are constructed sequentially in

this approach. In XGBoost, weights are very significant. All independent variables are given weights and subsequently fed into the decision tree, which predicts outcomes (Chen and Guestrin, 2016). This classifier was used with 10000 estimators where the max depth of the tree is six, along with a gbTree booster.

Two other machine learning classifiers are also used, which include K nearest neighbour and an ensemble learning approach using a random forest classifier. The k-nearest neighbours method, often known as KNN or k-NN, is a non-parametric, supervised learning classifier that makes classifications or predictions about grouping individual data points based on closeness. It may be used for both regression and classification issues. However, it is most commonly employed as a classification technique, based on the idea that comparable points can be discovered close together (Sun and Huang, 2010).

Random forest classifier employs averaging to increase predicted accuracy and control over-fitting by fitting several decision tree classifiers on various sub-samples of the dataset. If `bootstrap=True` (default), the sub-sample size is regulated by the `max samples` argument; otherwise, the whole dataset is utilized to create each tree (Garreta and Moncecchi, 2013). The random forest classifier was set to have 10000 estimators, and the rest of the parameters were kept to default, including `max-depth` set to `None` and `minimum leaf samples` set to 1.

4.2 Face-mesh Based Learning Using MobileNetV2 Architecture

Traditional machine learning techniques have inherent limitations when identifying features and information in picture data. Due to their multi-level architecture, CNNs, in particular, assist in getting around these restrictions. This approach has a facial landmark extraction component. It then uses Google's media pipe, a cutting-edge tool that calculates 468 3D face markers in real-time, even on mobile devices, to produce a facial mesh.

Before feeding our Convolutional neural network with train and test samples, image samples must be pre-processed. The images are firstly resized to 224 by 224. Some photos can be in grayscale one channel. Therefore, we convert them to a three-channel using `dstack` from NumPy. The process then reads the image in RGB format and applies pixel normalization. Google's mediapipe is then used to extract a facial mesh from the normalized image. Once the facial mesh is generated, it is placed on a black background, concluding the image pre-processing stage. Once the image pre-processing has been completed,

our convolutional neural network is ready to accept the input data. Before feeding data to the CNN, the training data goes through a data augmentation stage, which increases the diversity of a dataset without the need to collect more data.

The proposed study uses Keras's sequential model. MobileNetV2 architecture forms the first layer of our model, which is a lightweight and memory-efficient architecture, followed by a two-dimensional Global Average Pooling layer. Global average pooling is intended to take the role of fully linked layers in conventional CNNs. The goal is to produce one feature map in the final `mlpconv` layer for each category that corresponds to the classification problem. We average each feature map, and the resultant vector is sent straight into the softmax layer rather than constructing fully linked layers on top of the feature maps. We then add a dropout layer with a 20% dropout rate. Drop out is a regularization strategy to stop overfitting during the training of a neural network model. A specific number of neurons in the network are ignored or dropped out randomly using the technique. Finally, we add a fully connected layer with a softmax activation function for binary classification.

Slight variations in the colour of the mesh generated mild variations in the performance and accuracy of the method. The proposed system used a white background along with a black face mesh, a black background with a white face mesh and a black background with a different coloured face mesh. All the variations in feature extraction are depicted in the Figure below.



Figure 4: Variations of face meshes generated.

The reason for generating various feature templates was to develop a variety of results for a comprehensive comparison. The results section will elaborate on the scores achieved using each feature template depicted above.

Classifier	Precision	Recall	F1-Score	Accuracy
SVM	81.5%	74.5%	75.5%	78.09%
XGBoost	94%	93.5%	93.5%	93.81%
KNN	87.5%	78.5%	80.5%	83.16%
RFC	94.5%	95%	94.5%	94.68%

Table 2: Classification report for traditional machine learning classifiers

5 Results

5.1 Traditional Machine Learning Approach with Facial Distance Measures

In our first approach for facial paralysis recognition, we used traditional machine learning classifiers which included:

1. Support Vector machine.
2. XGBoost Learning Algorithm
3. K Nearest Neighbours
4. Random Forest Classifier

Precision and recall measures and the F1 score were calculated for each of the classifiers using scikit learn’s classification report. Scikit learn’s classification report generates precision, recall and precision scores for a classifier. Finally, the overall accuracy score was calculated for each of the classifiers. The function used to calculate the accuracy score computes subset accuracy, meaning that the set of labels predicted for a sample should match the corresponding set of ground truth labels. Table 2 summarizes the classification scores for all the various classifiers used in our initial method.

The support vector machine achieved an overall accuracy of 78.09%, with an average recall of 74.5%. This shows us that the classifier predicted the relevant cases correctly 74.5% of the time. Precision scores depict that classes were correctly labelled with 81.5% accuracy, whereas healthy patients were labelled with 75% accuracy. The overall accuracy of the classifier is 78.09%, indicating that 78.09% of the predicted labels matched precisely with the ground truth values.

The report shows that the XGBoost classifier performed better than the support vector machine, with a precision and accuracy of approximately 94%. The classifier used two thousand estimators, and the rest of the parameters were kept to default. The XGBoost classifier had a 20% increase in accuracy score compared to the support vector machine.

K nearest neighbour also outperformed the support vector machine with an accuracy score of 83% with a 5% increase in overall classification accuracy. It (KNN) achieved an average precision score

Method	Precision	Recall	F1-Score	Accuracy
Huang et al [7]	93%	88%	-	-
Barbosa et al [8]	-	98.12%	-	-
Kim et al [15]	92.3%	90%	-	88.9%
Gemma et al[2]	99.24%	-	-	97.22%
SVM	81.5%	74.5%	75.5%	78.09%
XGBoost	94%	93.5%	93.5%	93.81%
KNN	87.5%	78.5%	80.5%	83.16%
RFC	94.5%	95%	94.5%	94.68%
MobileNetV2	99%	99%	99%	98.93%

Table 3: Comparison with previous studies

of 87.5%. However, this classifier did not perform as well as the XGboost classifier.

A random forest classifier based on an ensemble learning technique outperformed all the classifiers in our approach with an accuracy score of 94.68%, as shown by the table below. The classifier used ten thousand estimators to predict while having the rest of the hyperparameters as default.

5.2 Face-mesh Based Learning Using MobilenetV2 Architecture

The second approach used a convolutional neural network for classification purposes, specifically a MobileNetV2 architecture. MobileNetV2’s architecture starts with a fully convolutional layer with 32 filters and is followed by 19 remaining bottleneck layers. Because ReLU6 is reliable when utilized with low-precision computing, we choose it as the non-linearity (Sandler et al., 2018). We add a global average pooling layer after the mobilenet architecture, which converts the features into a single vector per image. A drop-out layer follows the global average pooling layer to avoid overfitting. Finally, the model has a fully connected layer with a softmax activation function for classification.

Deep-learning-based approach outperformed the traditional machine learning approaches in this particular use case with an overall accuracy of 98.93%. Let’s compare our deep-learning approach by taking the best-performing conventional technique, a random forest classifier in our case. We can see a 4% increase in the overall accuracy of the classifier.

Comparing our results against Huang et al., we can see a 5.5 per cent increase in precision. Work by Gemma et al. was able to achieve higher average precision than our model, but in terms of accuracy, our approach had a 1.71% increase. It is important to note that results for both kim et al (Kim et al., 2015) and Barbosa et al (Barbosa et al., 2019) made use of a private database.

Feature Template	Precision	Recall	F1-Score	Accuracy
Template A	99%	99%	99%	98.93%
Template B	99%	99%	99%	98.93%
Template C	99%	99%	99%	98.63%

Table 4: Classification report for deep learning approach using mobilenetV2 architecture

6 Ablation Study

The study implemented an ablation experiment to assess the performance of the deep learning model. The experiment generated various feature templates, as shown in Figure 4, to analyze the variance in the performance of the model. A cross-data set analysis was performed to measure the impact of data imbalance in the face of variability. Finally, the experiment generated a t-SNE or t-Distributed Stochastic Neighbor Embedding report by converting the four-dimensional feature maps to 2-dimensional ones. The scatter plot for the 2-dimensional features helps us to determine which input data seems similar to the deep neural network.

6.1 Different Feature Templates

For a comparative study, we generated different colours for face-mesh at the feature extraction stage. Table 4 summarizes the model’s overall classification report with different feature templates as reported in Figure 4. The table above shows that the convolutional neural network performance in terms of accuracy was similar when given the first two types of feature templates. However, with a black background and a white face mesh, CNN’s performance decreased by 0.3%. The overall results show a massive improvement from the traditional techniques, with an accuracy of 98.93 %.

6.2 Cross Data set Validation

The performance of our model showed a great deal of variation when different combinations of data sets were used for training and testing purposes. Training and validation loss/accuracy curves were generated to analyze if the model was overfitting or not. Apart from the original combination of the data set, the experiment creates two different combinations of the data set that is already in use. The first combination used YouTube Facial paralysis data set, and TUFTs face data set for model training. In contrast, the Stroke face data set and Ck+ data set were used as testing sets for Unhealthy and healthy patients, respectively.

The second combination had a slight variation from the first combination. The training and testing

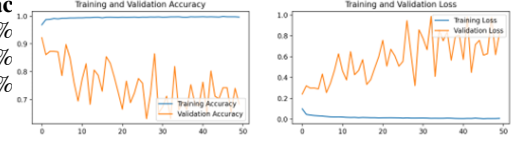


Figure 5: Training and validation learning curves for first combination of data sets.

set for unhealthy subjects remain the same, whereas, for healthy subjects, we swapped the CK+ and the TUFTS face data set for training and testing purposes. Results in Figure 6 below show that the model overfits faster than the first combination of data sets due to a steeper validation loss curve. It is important to note that during such experiments, the ratios between various data sets may vary due to the different sizes of the data sets. We do not claim that different data set combinations used in this experiment were entirely equal in ratio. However, it gives us a good indication of whether data imbalance impacts the face of variability.

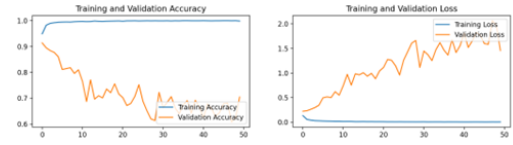


Figure 6: Training and validation learning curves for Second combination of data sets.

The third combination of data sets was similar to the data set mentioned in section 3.1. However, we reduced the number of CK+ data set images from 981 to 730 for the training set of healthy subjects. Figure 7 below shows that the validation loss and training loss decrease gradually, which indicates that the model is not overfitting. This supports the claim of Huang et al., where adding CK+ makes our model more robust against facial expression variation. The decline in healthy subject images from the CK+ data set resulted in an overall accuracy of 98.74% which has a 0.20% decrease from the original model where 981 images were used from the CK+ data set.

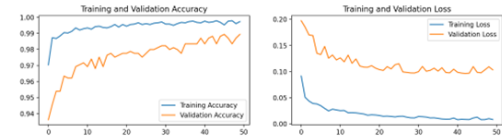


Figure 7: Training and validation learning curves for third combination of data sets.

6.3 TSNE report

The final part of our ablation study provides a TSNE report on the different feature vectors that were generated in our methodology i.e Section 4. By detecting observable clusters based on the similarity of data points with many attributes, t-SNE aims to uncover patterns in the multidimensional data by mapping it to a lower-dimensional space. By detecting observable clusters based on the similarity of data points with many attributes, t-SNE aims to uncover patterns in the multidimensional data by mapping it to a lower-dimensional space.

6.3.1 TSNE Report on Distance Measures Computed for Traditional Machine learning techniques

The TSNE report in Figure 8 provides a scatter plot of the two classes under observation. 0 represents healthy subjects, whereas 1 represents unhealthy subjects. The figure also shows small clusters of unhealthy classes forming within the cluster of healthy cases.

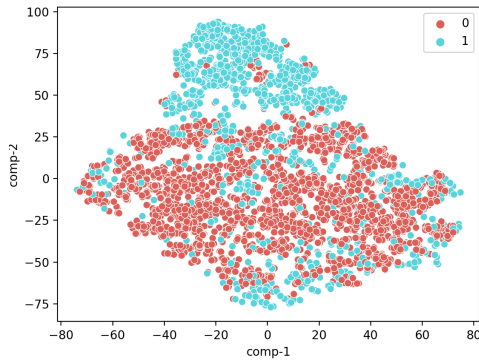


Figure 8: Scatter plot of t-SNE on distance measures calculated for traditional machine learning approach.

6.3.2 TSNE report for Different Feature Templates for Deep Learning Approach

There were significant differences in scatter plots when different feature templates were used for our deep learning approach. Figure 9 represents a TSNE scatter plot for a white face mesh. We see the formation of two different clusters within the scatter plot.

On the other hand, Figure 10 shows us a TSNE scatter plot for a coloured face mesh with a black background. We see a cluster of healthy subjects forming within the unhealthy subjects' cluster. This condition can occur due to some occlusions that may have malformed descriptors. Future study will look

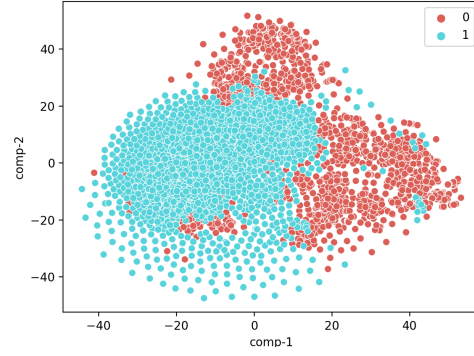


Figure 9: Scatter plot of t-SNE on feature template with a white face mesh over a black background.

into why such clusters formed and more robust quality checks will be employed at image pre-processing level so that occlusions with malformed descriptors are avoided.

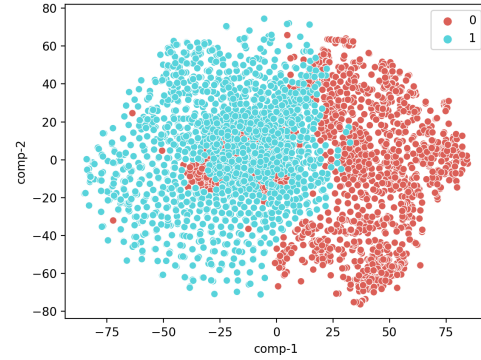


Figure 10: Scatter plot of t-SNE on feature template with a coloured face mesh over a black background.

7 Conclusion

A method for detecting facial paralysis in a picture was presented, using two different approaches for a comparative study. The first approach extracted 26 facial measures computed using facial landmarks during the feature extraction phase and used various binary classifiers which provide a healthy or unhealthy label. Classifiers for the first approach included a support vector machine, XGBoost classifier, K Nearest Neighbor and a random forest classifier with a random forest classifier outperforming every other classifier with an accuracy score of 94.68%. On the other hand, the deep learning-based approach for image classification used MobileNetV2 as a base model for the overall structure and a different feature space resulting in a facial mesh. Looking at our results, we achieved an accuracy of approximately 98.93%,

which shows that the model outperformed all the previous studies mentioned in the article and our initial approach. Developing such incremental and improved methods results in higher reliability and accuracy in medical diagnostic systems. These methods can also serve as the basis for forming standardized tools for medical assessments, treatment, and monitoring.

REFERENCES

- Barbosa, J., Seo, W.-K., and Kang, J. (2019). parafacetest: an ensemble of regression tree-based facial features extraction for efficient facial paralysis classification. *BMC Medical Imaging*, 19(1):1–14.
- Boyko, N., Basystiuk, O., and Shakhovska, N. (2018). Performance evaluation and comparison of software for face recognition, based on dlib and opencv library. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, pages 478–482. IEEE.
- Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.
- Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240.
- Garreta, R. and Moncecchi, G. (2013). *Learning scikit-learn: machine learning in python*. Packt Publishing Ltd.
- Grishchenko, I., Ablavatski, A., Kartynnik, Y., Raveendran, K., and Grundmann, M. (2020). Attention mesh: High-fidelity face mesh prediction in real-time. *arXiv preprint arXiv:2006.10962*.
- Guarin, D. L., Dusseldorp, J., Hadlock, T. A., and Jowett, N. (2018). A machine learning approach for automated facial measurements in facial palsy. *JAMA facial plastic surgery*, 20(4):335–337.
- Hassaballah, M. and Hosny, K. M. (2019). Recent advances in computer vision. *Studies in computational intelligence*, 804:1–84.
- Hato, N., Matsumoto, S., Kisaki, H., Takahashi, H., Wakisaka, H., Honda, N., Gyo, K., Murakami, S., and Yanagihara, N. (2003). Efficacy of early treatment of bell’s palsy with oral acyclovir and prednisolone. *Otology & neurotology*, 24(6):948–951.
- Jiang, C., Wu, J., Zhong, W., Wei, M., Tong, J., Yu, H., and Wang, L. (2020). Automatic facial paralysis assessment via computational image analysis. *Journal of Healthcare Engineering*, 2020.
- Jison Hsu, G.-S., Huang, W.-F., and Kang, J.-H. (2018). Hierarchical network for facial palsy detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 580–586.
- Khan, M., Chakraborty, S., Astya, R., and Khepra, S. (2019). Face detection and recognition using opencv. In *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pages 116–119. IEEE.
- Kim, H. S., Kim, S. Y., Kim, Y. H., and Park, K. S. (2015). A smartphone-based automatic diagnosis system for facial nerve palsy. *Sensors*, 15(10):26756–26768.
- Mgheed, R. M. A. (2021). Scalable arabic text classification using machine learning model. In *2021 12th International Conference on Information and Communication Systems (ICICS)*, pages 483–485. IEEE.
- Panetta, K., Wan, Q., Agaian, S., Rajeev, S., Kamath, S., Rajendran, R., Rao, S. P., Kaszowska, A., Taylor, H. A., Samani, A., et al. (2018). A comprehensive database for benchmarking imaging systems. *IEEE transactions on pattern analysis and machine intelligence*, 42(3):509–520.
- Parra-Dominguez, G. S., Sanchez-Yanez, R. E., and Garcia-Capulin, C. H. (2021). Facial paralysis detection on images using key point analysis. *Applied Sciences*, 11(5):2435.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.
- Sun, S. and Huang, R. (2010). An adaptive k-nearest neighbor algorithm. In *2010 seventh international conference on fuzzy systems and knowledge discovery*, volume 1, pages 91–94. IEEE.
- Tiemstra, J. D. and Khatkhate, N. (2007). Bell’s palsy: diagnosis and management. *American family physician*, 76(7):997–1002.
- Wang, T., Zhang, S., Dong, J., Liu, L., and Yu, H. (2016). Automatic evaluation of the degree of facial nerve paralysis. *Multimedia Tools and Applications*, 75(19):11893–11908.
- Wu, Y., Hassner, T., Kim, K., Medioni, G., and Natarajan, P. (2017). Facial landmark detection with tweaked convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):3067–3074.