The Indonesian Journal of Computer Science
www.ijcs.net
Volume 13, Issue 4, August 2024
https://doi.org/10.33022/ijcs.v13i4.4323

Comparison of Machine Learning Algorithms for Face Classification Using FaceNet Embeddings

Catoer Ryando[1], Riyanto Sigit[2], Setiawardhana[3], Bima Sena Bayu Dewantara[4]
catoerryando@gmail.com[1], riyanto@pens.ac.id[2], setia@pens.ac.id[3], bima@pens.ac.id[4]
[1]Department of Electrical Engineering, Electronic Engineering Polytechnic Institute of Surabaya
[2,3,4]Department of Informatics and Computer Engineering, Electronic Engineering Polytechnic Institute of Surabaya

Article Information

Received : 17 Jul 2024
Revised : 26 Jul 2024
Accepted : 30 Aug 2024

Keywords

Face recognition, FaceNet, Embedding, Classification methods

Abstract

In recent years, face recognition has grown significantly in importance and popularity. Google created FaceNet, a deep learning system, in 2015, and it performs very well in creating extremely precise and personalised numerical representations of faces, or embeddings. In order to swiftly and effectively identify people, this study evaluates FaceNet's effectiveness in producing face embeddings and applies it to a variety of classification techniques, including support vector machine (SVM), decision tree, random forest, and k nearest neighbours (KNN). A dataset with a wide range of positions, facial expressions, and lighting settings is used for the assessment. The findings of the experiment demonstrated that SVM with an radial basis function (RBF) kernel outperformed the other assessed classification techniques, achieving the maximum accuracy of 95%. These findings demonstrate the wide range of applications that face recognition technology may be used for, including identity management and security in different settings.

A. Introduction

Facial recognition has become a very important and rapidly growing topic in recent years. This is mainly due to the need for more reliable and accurate security systems, which is crucial in various sectors such as law enforcement, access control, and service personalization. In addition, advances in hardware and software technology have enabled the development of more sophisticated and efficient facial recognition systems. These include improvements in computer processing speed, the availability of large face databases, and advances in deep learning algorithms that enhance facial recognition capabilities.

FaceNet is one of the leading technologies in this field, developed by Google [1]. It uses deep learning techniques to generate a numerical representation (embedding) of a face [1] that effectively encapsulates key features. This process involves converting an image of a face into a series of numbers that reflect various aspects of the face, such as shape, feature position, and texture [2]. FaceNet is particularly effective in handling the wide variations in pose, lighting, and expression that often hamper other face recognition systems. This makes it particularly useful for applications that require a high degree of accuracy in face recognition across a wide range of conditions.

Although FaceNet has provided excellent results, there is still potential to improve the accuracy further. One approach that can be taken is through the use of advanced machine learning techniques to classify the resulting embedding. By applying algorithms such as SVM [3], decision tree [4], random forest, and KNN [5], we can more effectively distinguish between individuals even under very challenging conditions. These classifiers can help in reducing identification errors and improving system reliability. The performance evaluation of these classification algorithms can be seen from the results of metrics such as accuracy, average weighted precision, and average cross-validation, all of which provide insight into the effectiveness of the methods in real-world scenarios.

B. Literature Study

There are several very good face recognition algorithms that use deep learning. Facebook's DeepFace has a 97.35% accuracy rate with a 0.25% standard deviation [6]. A significantly greater accuracy of 97.45% with a 0.26% standard deviation is offered by DeepID [7]. Google unveiled FaceNet in 2015, with the greatest accuracy of 99.63%. The system was trained using a proprietary dataset made up of millions of photos from social media [1]. FaceScrub and CASIA datasets were used to train OpenFace, an extension of DeepFace and GoogleNet [8].

The accuracy of 99.83% is shown on the LFW dataset by combining deep learning with the FaceNet model for feature extraction and face classification using SVM [3]. This accuracy is greater than that of 99.63% when using FaceNet alone

[1]. Numerous classification techniques, including support vector machine (SVM) [9], k-nearest neighbours (KNN) [10], decision tree [11], and random forest [12], were extensively used in face recognition research and applications prior to the rise in popularity of deep learning.

The way SVM works is by identifying the hyperplane that best divides the data classes. KNN uses a feature space's closest neighbours to choose which data to classify. Based on a series of if-then rules extracted from data attributes, decision trees classify data. In order to increase accuracy and decrease overfitting, random forests, which are collections of several decision trees, mix predictions from different trees.

With many methods in face recognition, this research aims to use FaceNet as feature extraction from faces. FaceNet converts facial images into high dimensional vectors that describe the unique characteristics of each face, called embedding. This embedding can then be used to compare different faces and determine if they belong to the same person.

There will be other techniques employed in the classification step, including SVM, decision trees, random forests, and KNN. A number of combinations of these techniques will be put to the test and their effectiveness examined. For instance, to classify facial identity, SVM may be fed the FaceNet embedding as input. It is anticipated that combining these techniques would result in the combination that offers the optimum efficiency and accuracy for face recognition. Measuring accuracy, precision, recall, and F1-score for every tested combination of techniques is part of the performance analysis process. To make sure the suggested techniques can be used in practical settings, this study will also assess the processing speed and computing resource needs.

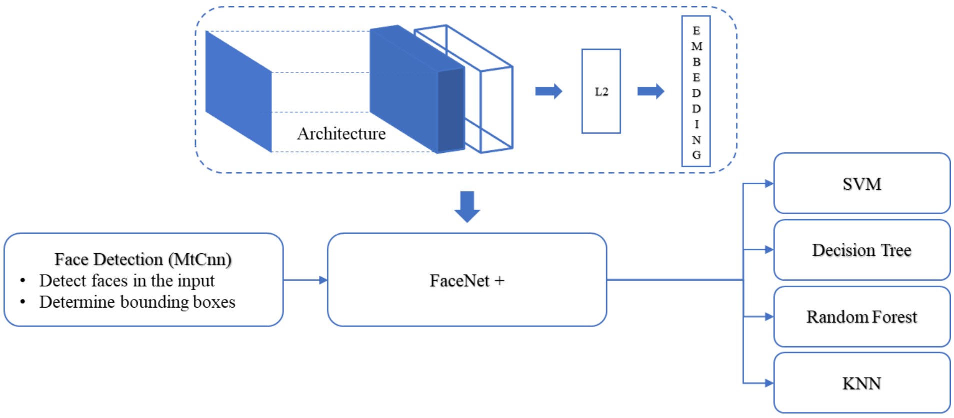

Figure 1. System diagram

C. Method

The process of the face recognition system, which consists of three primary parts: face detection, feature extraction, and classification, in Figure 1. Using MTCNN (Multi-task Cascaded Convolutional Networks), the procedure begins with face detection. This method finds faces in the input picture and calculates the bounding boxes around each face. FaceNet, a deep learning network that creates a vector representation or embedding of faces, is used for feature extraction after face identification. To guarantee that the embedding has a constant length a crucial factor for accuracy in later classification steps this method is followed by normalisation using L2. Using a variety of machine learning techniques, including support vector machine (SVM), decision tree, random forest, and k-nearest neighbors (KNN), the final step is to classify the embedding generated by FaceNet. Based on the embedding produced by FaceNet, these algorithms are used to recognize or identify faces; each method takes a different approach and produces a different outcome for the classification process.

1. MTCNN

This research uses a dataset from Kaggle that contains images of celebrity faces [13]. The dataset comes with 17 labels and provides 100 photos per label, making a total of 700 photos. These photos are used to train FaceNet, and will then be classified using different algorithms and parameters. Before the training process, face detection is performed first. In this dataset, the photo content is not limited to just a person's face, but also includes context or other information about the subject or the environment where the photo was taken, such as the background and objects around the subject.

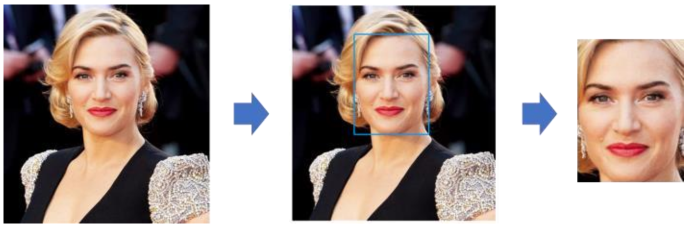

Figure 2. Face detection using MTCNN

The pre-processing of the dataset is done by removing other parts besides the face from the photo using Multi-Task Cascaded Convolutional Network (MTCNN) face detection [14], as illustrated in Figure 2. After the face is cropped using MTCNN, the processed dataset becomes cleaner, focused on the face, and has a uniform size of 160x160 pixels. This processed dataset is then used for training using FaceNet, which produces face embeddings. These embeddings are numerical representations of various facial features. These embeddings will then be classified using several machine learning techniques, including support vector machine (SVM), decision tree, random forest, and k-nearest neighbors (KNN).

2. FaceNet

FaceNet is a deep learning system designed for identity verification and facial recognition [1] with a high degree of accuracy. Figure 3 shows that FaceNet utilizes a deep neural network with the Inception ResNet architecture to process facial images as input. This architecture allows FaceNet to extract important features from faces more effectively, overcome the bottlenecks often encountered in deep neural networks, and build more complex models. The extracted features are then normalized using L2 normalization to produce the face embedding, a compact numerical vector representation [15].

Figure 3. FaceNet [1]

https://doi.org/10.33022/ijcs.v13i4.4323

FaceNet produces an embedding as shown in Figure 4, which is a lowdimensional numerical vector of 128 vector elements representing the face of each learning. This embedding is a compressed representation of important facial features, such as eye distance, nose shape, and facial contours. The main advantage of FaceNet embedding is its compactness. Compared to the original image, the embedding is much more compact, allowing for more efficient data storage and comparison [9]. In addition, embedding allows FaceNet to effectively compare facial similarities by calculating the distance between vectors. This allows FaceNet to identify the same person under various lighting conditions and shooting angles.

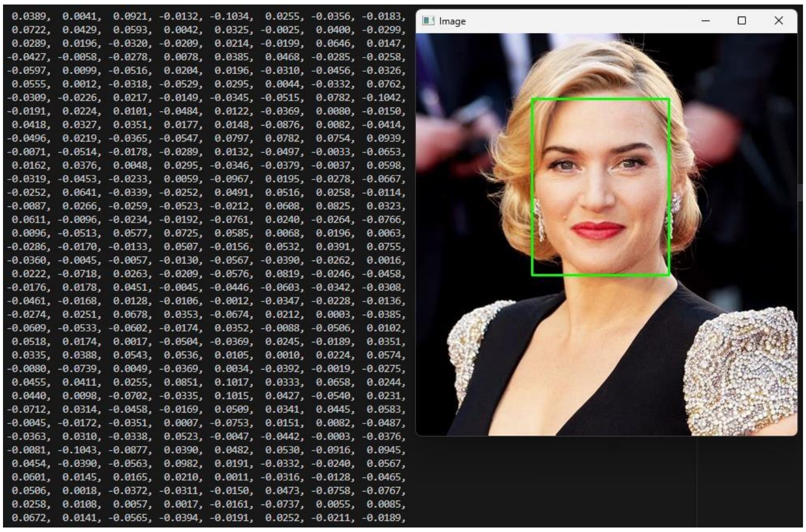

Figure 4. Embedding

3. SVM

One machine learning approach used for regression and classification is called support vector machine (SVM) [16]. As shown in Figure 5, SVM operates by determining the best hyperplane to divide data into distinct groups [17]. This hyperplane functions as a line or surface in a higher dimension to divide the data into two classes.

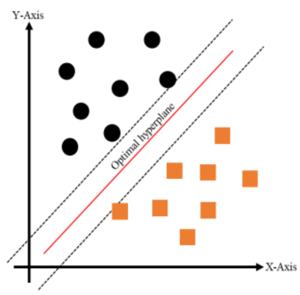

Figure 5. Support vector machine (SVM)

A kernel in support vector machines (SVM) is a function that raises the dimension of the data such that it may be separated in this higher space even while the original space is not linearly separable [17]. Sigmoid, Polynomial, Linear, and Radial Basis Function (RBF) are a few frequently utilised kernel types. When data can be divided linearly, linear kernels are used; in contrast, polynomial kernels utilise polynomials to translate data into higher dimensions. The RBF kernel is very helpful for non-linearly separable data as it maps the data into higher dimensions using a Gaussian function. Conversely, sigmoid kernels are often used in artificial neural networks and make use of sigmoid functions. A regularisation parameter in SVM, the C value regulates the trade-off between maximising margin and limiting classification error. In contrast, the RBF kernel's Gamma function establishes the relative weight of each individual data point. When there are more dimensions than samples, SVM remains efficient and performs very well in highdimensional domains. Because the approach only employs a subset of training points in the decision function known as support vectors, it is also memory efficient.

4. Decision Tree

A tree structure, as seen in Figure 6, is used by the decision tree machine learning technique for regression and classification to generate judgements based on input characteristics. A test on an attribute is represented by each internal node in the tree structure, which is made up of branches and nodes. A class prediction or regression result is represented by each leaf node [11].

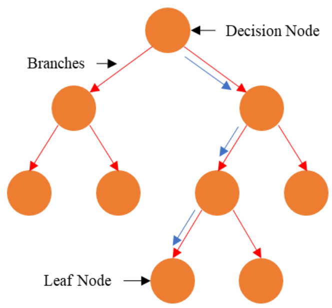

Figure 6. Decision tree

The first step in creating a decision tree is determining which qualities, according to measures like variance reduction, entropy (information gain), and Gini impurity, are the best for splitting the data [18]. The primary benefits of decision trees are their capacity to handle many attribute types (both category and numerical), their ease of understanding, and their ability to manage data with missing values. One crucial variable is max depth [19], which is the tree's greatest depth measured from its roots to its leaves. Max depth regulates the model's complexity to avoid overfitting in the case of a big model and underfitting in the case of a small model.

5. Random Forest

An ensemble-based machine learning technique called random forest is used to regression and classification problems. As seen in Figure 7, this approach integrates many decision trees to create a more reliable and stable model

[20]. In order to minimise model variance and avoid overfitting, a random portion of the training data and a random subset of the features are used to build each tree in the forest. For classification tasks, the majority vote is used to determine the final prediction result, and for regression tasks, the average prediction is calculated from all decision trees in the forest [12].

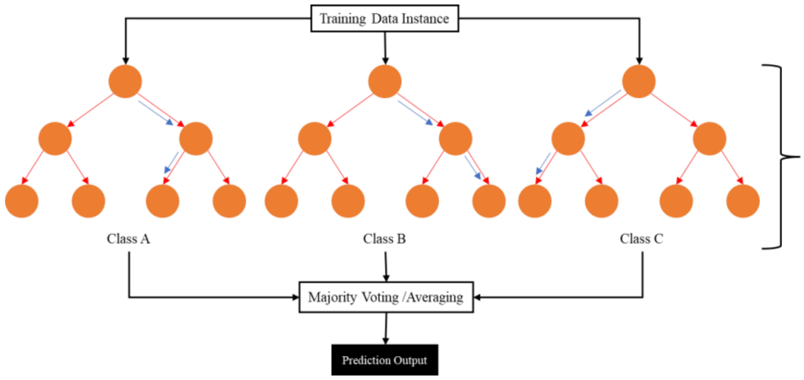

Figure 7. Random forest

The "n-estimators" parameter, which specifies how many decision trees must be constructed for the model, is one of the key elements of random forest. Since additional trees lower the total model variance, the model predictions will be more reliable and accurate the more trees are employed. The amount of time and resources needed to train the model must be traded off, however. Thus, it's critical to choose n_estimators appropriately in order to balance computational efficiency and model performance.

6. KNN

One machine learning technique for classification and regression problems is k-nearest neighbours (KNN). It is an example of an instance-based learning method that is non-parametric, meaning that KNN does not create explicit models during training or make any significant assumptions about the data's distribution. Rather, KNN generates fresh predictions by comparing them to preexisting data [21]. KNN uses the majority of the classes of its closest neighbours to classify a new data point in classification tasks [10]. In the meanwhile, as shown in Figure 8, the prediction value for the regression task is determined by averaging the values of the closest neighbours.

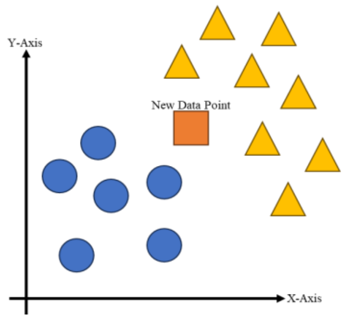

Figure 8. K-Nearest Neighbours (KNN)

The number of closest neighbours to be employed for predictions is determined by the KNN's primary parameter, n_neighborsn [22]. More neighbours are taken into account in the prediction process the higher the kkk number, which might smooth out the model but lessen its sensitivity to specific inputs. On the other hand, if kkk is too tiny, the model may be very susceptible to data noise.

D. Result and Discussion

FaceNet uses triplet loss to produce high-quality embeddings that ensure the distance between embeddings of similar faces is smaller than that of different faces. In this research, SVM, decision tree, random forest, and K-NN use these embeddings as input features to build classification models that are able to identify or classify faces into specific identities. To improve the classification accuracy, many parameters can be changed in each method, for example: kernel in SVM, tree depth in decision tree, number of trees in random forest, and number of neighbors in K-NN. To guarantee strong generalization in face recognition, model assessment is performed.

To provide a complete view of the model's capacity to identify face embeddings from FaceNet, the model's performance is assessed using accuracy, Average Cross Validation Score Comparison, and Weighted Average Precision Comparison in this findings and experiments section. While the Average Cross Validation Score gauges the model's resilience to changes in the data, Accuracy gives information on the total proportion of accurate predictions. By taking into account the distribution of classes in the dataset, the Weighted Average Precision comparison is crucial for assessing the precision of predictions. When these measures are combined, they provide profound insights that help choose the optimal model for face embedding categorization while maximizing performance and flexibility to a range of data scenarios.

In Figure 9, the Mean Cross-Validation Score Comparison graph gives a clear picture of the stability and generalization of the models across the datasets. SVM with RBF kernel (`C = 10, gamma = 'scale'`) and polynomial kernel (`C = 1, degree = 3`) show the highest scores, demonstrating their ability to handle non-linear and complex data. Random forest also showed consistent and excellent performance, especially with a larger number of trees and higher tree depth. K-NN with `n_neighbors = 3` and distance weighting showed good performance, although not as good as some SVM and random forest configurations. Decision tree has lower performance than other models, especially when the tree depth is limited.

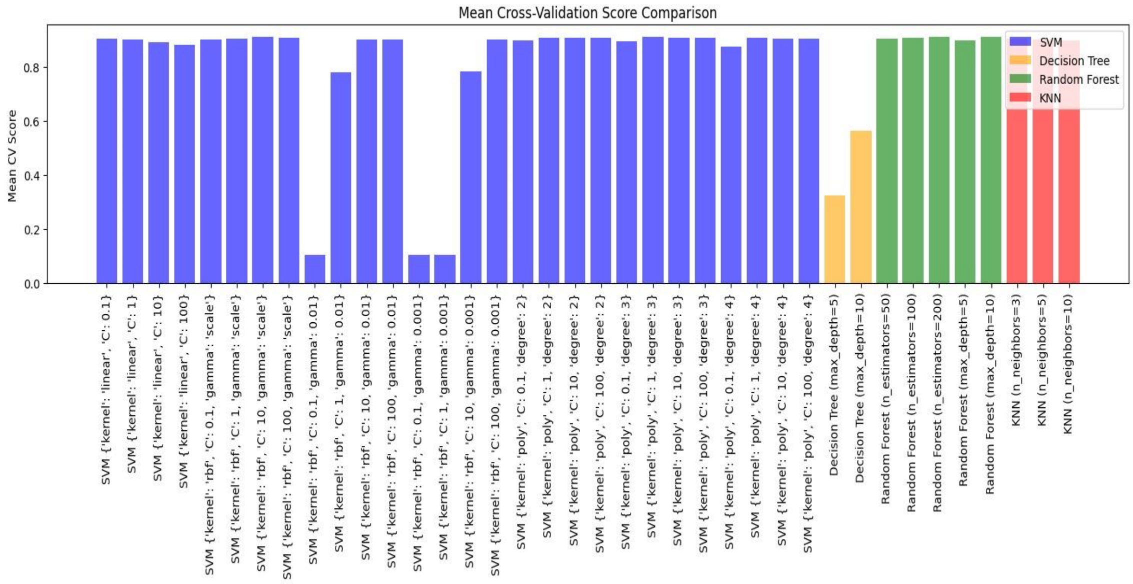

Figure 9. Mean cross validation

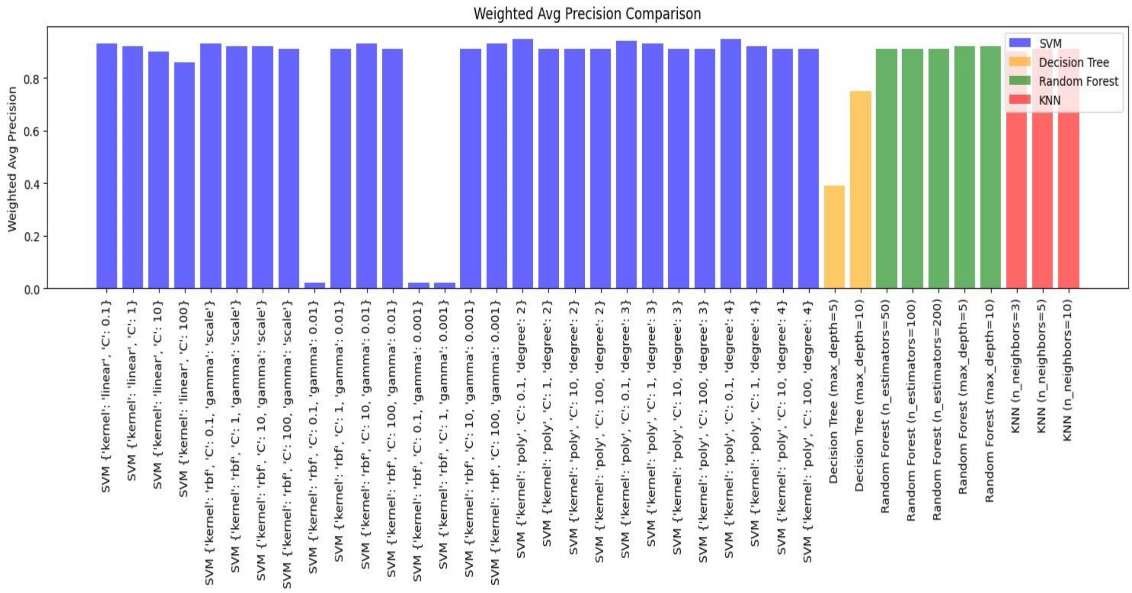

Figure10. Weighted avg

The Weighted Average Precision Comparison graph in Figure 10 shows that SVM with RBF kernel (`C = 10, gamma = 'scale'`) and polynomial kernel (`C = 1, degree = 2`) achieve the highest precision, highlighting their ability to handle nonlinear and complex data. Random Forest also showed excellent and consistent precision, especially with a larger number of trees and tree depth. K-NN with `n_neighbors = 3` and distance weighting provided good precision, although not as good as some SVM and random forest configurations. In contrast, decision tree has lower precision, especially when the tree depth is limited. This suggests that more complex models such as SVM and random forest tend to provide higher precision on non-linear and complex data compared to simpler models such as decision tree and K-NN.

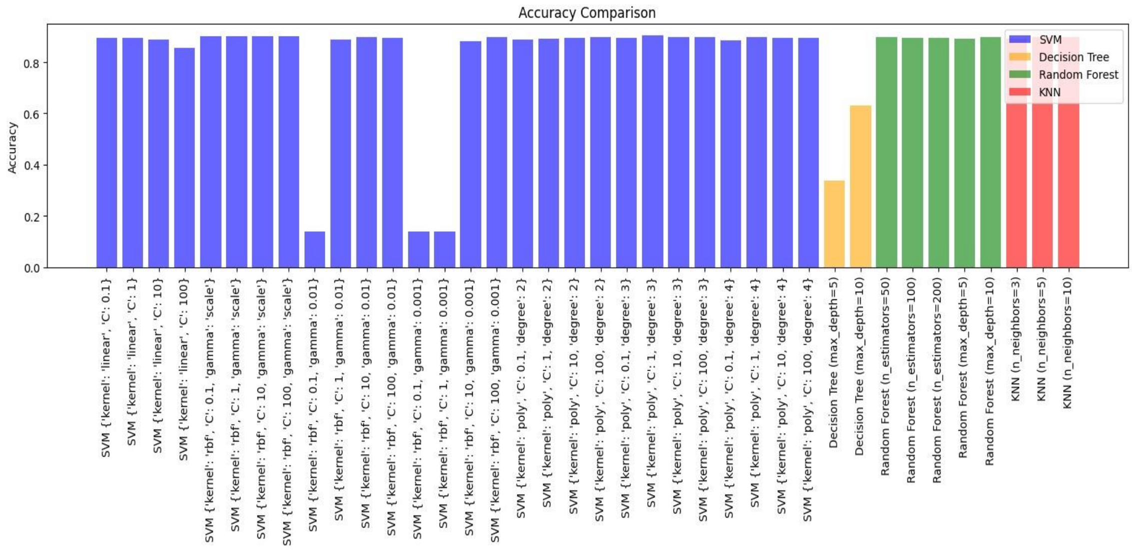

Figure 11. Accuracy

Table 1. Accuracy of training

Parameters	Accuracy	Parameters	Accuracy
SVM {'kernel': 'linear', 'C': 0.1}	0,895	SVM {'kernel': 'poly', 'C': 100, 'degree': 2}	0,8975
SVM {'kernel': 'linear', 'C': 1}	0,895	SVM {'kernel': 'poly', 'C': 0.1, 'degree': 3}	0,895
SVM {'kernel': 'linear', 'C': 10}	0,8875	SVM {'kernel': 'poly', 'C': 1, 'degree': 3}	0,905
SVM {'kernel': 'linear', 'C': 100}	0,855	SVM {'kernel': 'poly', 'C': 10, 'degree': 3}	0,8975
SVM {'kernel': 'rbf', 'C': 0.1, 'gamma': 'scale'}	0,9	SVM {'kernel': 'poly', 'C': 100, 'degree': 3}	0,8975
SVM {'kernel': 'rbf', 'C': 1, 'gamma': 'scale'}	0,9	SVM {'kernel': 'poly', 'C': 0.1, 'degree': 4}	0,885
SVM {'kernel': 'rbf', 'C': 10, 'gamma': 'scale'}	0,9025	SVM {'kernel': 'poly', 'C': 1, 'degree': 4}	0,8975
SVM {'kernel': 'rbf', 'C': 100, 'gamma': 'scale'}	0,9	SVM {'kernel': 'poly', 'C': 10, 'degree': 4}	0,895
SVM {'kernel': 'rbf', 'C': 0.1, 'gamma': 0.01}	0,14	SVM {'kernel': 'poly', 'C': 100, 'degree': 4}	0,895
SVM {'kernel': 'rbf', 'C': 1, 'gamma': 0.01}	0,8875	Decision tree (max_depth=5)	0,3375
SVM {'kernel': 'rbf', 'C': 10, 'gamma': 0.01}	0,8975	Decision tree (max_depth=10)	0,63
SVM {'kernel': 'rbf', 'C': 100, 'gamma': 0.01}	0,895	Random forest (n_estimators=50)	0,8975

SVM {'kernel': 'rbf', 'C': 0.1, 'gamma': 0.001}	0,14	Random forest (n_estimators=100)	0,895
SVM {'kernel': 'rbf', 'C': 1, 'gamma': 0.001}	0,14	Random forest (n_estimators=200)	0,895
SVM {'kernel': 'rbf', 'C': 10, 'gamma': 0.001}	0,8825	Random forest (max_depth=5)	0,8925
SVM {'kernel': 'rbf', 'C': 100, 'gamma': 0.001}	0,8975	Random forest (max_depth=10)	0,8975
SVM {'kernel': 'poly', 'C': 0.1, 'degree': 2}	0,8875	KNN (n_neighbors=3)	0,8925
SVM {'kernel': 'poly', 'C': 1, 'degree': 2}	0,8925	KNN (n_neighbors=5)	0,8975
SVM {'kernel': 'poly', 'C': 10, 'degree': 2}	0,895	KNN (n_neighbors=10)	0,8975

In Table 1 and Figure 11, the embedding of FaceNet is used as an input feature for SVM, decision tree, random forest, and K-NN classification models. The results show that SVM with polynomial kernel at configuration `C = 1` and `degree = 3` achieves the highest accuracy of 0.905, while SVM with RBF kernel at configuration `C = 10` and `gamma = 'scale'` also performs very well with an accuracy of 0.9025. Random forest and K-NN showed consistent performance with an accuracy of around 0.8975, signaling their ability to capture data variations without overfitting. In contrast, decision tree showed lower performance with the highest accuracy of 0.63 at a tree depth of 10, possibly due to the tendency of overfitting the training data. Thus, SVM with polynomial kernel and RBF provide the best accuracy for face embedding classification from FaceNet, while random forest and K-NN are also reliable choices, but decision tree requires additional techniques to improve its accuracy.

E. Conclusion

In this work, we use the FaceNet embedding to investigate how well different machine classification methods recognise and recognise faces. Google created FaceNet. involves classifying the face embedding produced by FaceNet using a number of classification methods, including support vector machine (SVM), decision tree, random forest, and k-nearest neighbours (KNN).

The findings demonstrated the capacity of SVM to handle complicated and non-linear data, with the polynomial kernel (C=1, degree=3) achieving the maximum accuracy of 90.5% and the RBF kernel (C=10, gamma='scale') following closely behind with an accuracy of 90.25%. With an accuracy of around 89.75%, random forest demonstrated stable and strong performance, particularly at greater tree number and tree depth, demonstrating its dependability in capturing data variability without overfitting. With an accuracy of around 89.25%, KNN with n_neighbors=3 also fared well, but not as well as other SVM and random forest setups. However, in contrast to the other algorithms, decision tree performed worse, with the greatest accuracy of 63% at a tree depth of 10, maybe as a result of the training data's propensity to be overfit.

To provide a thorough picture of the model's capacity to categorise face embeddings, measures for accuracy, mean cross validation score, and weighted average precision were used in the model assessment process. The greatest results were obtained while categorising face embeddings from FaceNet using SVM with polynomial kernel and RBF. Random forest and KNN are also dependable options, however decision tree needs further methods to increase its accuracy. Overall, more complex models such as SVM and random forest are superior in handling non-linear and complex data than simpler models such as decision tree and k-NN. The study's findings provide crucial advice for selecting the best classification method for practical face recognition applications. This study demonstrates that a high degree of face recognition accuracy can be attained with the use of the embedding produced by FaceNet and an appropriate classification technique. This offers a potent and effective solution for security and service personalisation applications that need accurate face recognition.

F. References

[1] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 07-12-June, pp. 815–823, 2015, doi: 10.1109/CVPR.2015.7298682.

[2] C. Wu and Y. Zhang, “MTCNN and FACENET Based Access Control System for Face Detection and Recognition,” Autom. Control Comput. Sci., vol. 55, no. 1, pp. 102–112, 2021, doi: 10.3103/S0146411621010090.

[3] L. Q. Vu, P. T. Trieu, and H. S. Nguyen, “Implementation of FaceNet and support vector machine in a real-time web-based timekeeping application,” IAES Int. J. Artif. Intell., vol. 11, no. 1, pp. 388–396, 2022, doi: 10.11591/ijai.v11.i1.pp388-396.

[4] F. D. Adhinata and A. Junaidi, “Gender Classification on Video Using FaceNet A lgorithm and Supervised Machine Learning,” vol. 1, no. 1, 2022.

[5] F. D. Adhinata, D. P. Rakhmadani, and D. Wijayanto, “Fatigue Detection on Face Image Using FaceNet Algorithm and K-Nearest Neighbor Classifier,” J. Inf. Syst. Eng. Bus. Intell., vol. 7, no. 1, p. 22, 2021, doi: 10.20473/jisebi.7.1.2230.

[6] M. Wang and W. Deng, “Deep Face Recognition : A Survey,” pp. 1–31.

[7] Y. Sun, X. Wang, and X. Tang, “Deep learning face representation from predicting 10,000 classes,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., no. July, pp. 1891–1898, 2014, doi: 10.1109/CVPR.2014.244.

[8] M. S. Brandon Amos, Bartosz Ludwiczuk, “Openface: uniwersalna biblioteka rozpoznawania twarzy z aplikacjami mobilnymi,” C. Sch. ..., 2016 r., vol. 16, no. October 2015, pp. 1–18, 2016, [Online]. Available: http://reportsarchive.adm.cs.cmu.edu/anon/anon/usr0/ftp/2016/CMU-CS-16-118.pdf

[9] A. A. F. 1 and M. A. 3 1 Yassin Kortli 1, 2,*, Maher Jridi 1, “Face Recognition Systems: A Survey,” Sensors, vol. 8, no. 1, 2020, doi: 10.1186/s13673-0180157-2.

[10] X. Guo, "A KNN Classifier for Face Recognition," 2021 IEEE 3rd Int. Conf. Commun. Inf. Syst. Comput. Eng. CISCE 2021, no. Cisce, pp. 292–297, 2021, doi: 10.1109/CISCE52179.2021.9445908.
[11] D. Maturana, D. Mery, and Á. Soto, "Face recognition with decision tree-based local binary patterns," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6495 LNCS, no. PART 4, pp. 618–629, 2011, doi: 10.1007/978-3-642-19282-1_49.
[12] B. Nassih, A. Amine, M. Ngadi, Y. Azdoud, D. Naji, and N. Hmina, "An efficient three-dimensional face recognition system based random forest and geodesic curves," Comput. Geom. Theory Appl., vol. 97, p. 101758, 2021, doi: 10.1016/j.comgeo.2021.101758.
[13] V. Thakur, "Celebrity Face Image Dataset," Kaggle, 2024. Available online: https://www.kaggle.com/datasets/vishesh1412/celebrity-face-image dataset.
[14] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," IEEE Signal Process. Lett., vol. 23, no. 10, pp. 1499–1503, 2016, doi: 10.1109/LSP.2016.2603342.
[15] D. Akingbesote, Y. Zhan, R. Maskeliūnas, and R. Damaševičius, "Improving Accuracy of Face Recognition in the Era of Mask-Wearing: An Evaluation of a Pareto-Optimized FaceNet Model with Data Preprocessing Techniques," Algorithms, vol. 16, no. 6, 2023, doi: 10.3390/a16060292.
[16] S. Muawanah, U. Muzayanah, M. G. R. Pandin, M. D. S. Alam, and J. P. N. Trisnaningtyas, "Stress and Coping Strategies of Madrasah's Teachers on Applying Distance Learning During COVID-19 Pandemic in Indonesia," Qubahan Acad. J., vol. 3, no. 4, pp. 206–218, 2023, doi: 10.48161/Issn.27098206.
[17] M. A. Chandra and S. S. Bedi, "Survey on SVM and their application in image classification," Int. J. Inf. Technol., vol. 13, no. 5, 2021, doi: 10.1007/s41870017-0080-1.
[18] H. F. Zhou, J. W. Zhang, Y. Q. Zhou, X. J. Guo, and Y. M. Ma, "A feature selection algorithm of decision tree based on feature weight," Expert Syst. Appl., vol. 164, no. August 2020, p. 113842, 2021, doi: 10.1016/j.eswa.2020.113842.
[19] S. Y. Yi Liu, "Journal of Mathematics - 2022 - Liu - Application of Decision Tree-Based Classification Algorithm on Content Marketing.pdf."
[20] E. Xi, "Image Classification and Recognition Based on Deep Learning and Random Forest Algorithm," Wirel. Commun. Mob. Comput., vol. 2022, 2022, doi: 10.1155/2022/2013181.
[21] S. Zhang and J. Li, "KNN Classification With One-Step Computation," IEEE Trans. Knowl. Data Eng., vol. 35, no. 3, pp. 2711–2723, 2023, doi: 10.1109/TKDE.2021.3119140.
[22] W. Liu, J. Wei, and Q. Meng, "Comparisions on KNN, SVM, BP and the CNN for Handwritten Digit Recognition," Proc. 2020 IEEE Int. Conf. Adv. Electr. Eng. Comput. Appl. AEECA 2020, pp. 587–590, 2020, doi: 10.1109/AEECA49918.2020.9213482.