Look-Alike Face Recognition Using Deep Learning

Abdul Basit Mughal[1], Syed Mujtaba Haider[2], Rafi Ullah Khan[3]
[1]Department of Computer Science, Bahria University, Islamabad, Pakistan, [2]Department of Computational Mathematics, University of Pavia, Pavia, Italy, [3]Department of Electrical & Computer Engineering, Pak Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur, Pakistan.

Email: sardarbasitmughal344@gmail.com, syedmujtaba.haider01@universitadipavia.it, rafi.ullah@paf-iast.edu.pk

Corresponding Author: Abdul Basit Mughal (sardarbasitmughal344@gmail.com)

Received: 30-12-2023; **Accepted**: 02-03-2024; **Published**: 10-05-2024

Abstract: The identification of similarities and differentiation between look-alike and non-look-alike features in facial images is a burgeoning research area, holding significant promise for applications in imposter identification. This task is complicated by variations in poses, illumination, and gestures within images of the same individual. Convolutional Neural Networks have emerged as a potent tool for face recognition, demonstrating notable efficiency. This research presents a novel methodology that combines image processing and CNN techniques to predict look-alike facial images with a high degree of accuracy. Achieving an impressive 95% accuracy on a dataset comprising 5000 images, our approach surpasses current cutting-edge methods in the field. In the realm of look-alike face recognition, the IMDB-wiki dataset serves as the foundational dataset. For face recognition, we employed various VGG models, such as VGGFace, VGG16, and VGG19. These models possess the capability to extract diverse facial features. To determine the similarity between two faces, we utilized methods such as cosine similarity, KNN (K-Nearest Neighbors) Euclidean distance, and Manhattan distance. This research contributes to the advancement of effective techniques in the evolving landscape of look-alike face recognition.

Keywords: Face detection, facial recognition, look alike face detection, face similarity detection.

1. INTRODUCTION

In recent times, deep learning has been getting more and more common. An important area of application within this field is face recognition. In the context of face recognition, video frames or images are fed to computers as inputs and their objective is to determine the identities of individuals by comparing their facial features with those that have been previously labeled in a database. The rapid evolution of face recognition techniques can be attributed to the emergence of powerful image recognition methods that are rooted in deep learning, as well as the growing availability of facial data. These advancements have led to the utilization of face recognition techniques in various domains, such as identity verification for mobile banking, among others. Our Face Detection is a challenging task due to its strategy of finding all conceivable Faces in all conceivable areas with distinctive sizes of given pictures. In numerous proposed computer visions, it is considered vital to begin with a step. Nowadays Face detection is used in many fields including Surveillance, Forensic Analysis, and many more. As we know Face identification and face similarity are two different tasks so we will try to distinguish Face identification and face similarity. Face identification is a method to identify that there is a human face present in a given image or video While face similarity is the measure of resemblance between two facial images. As Shown in Figure 1. Some of the time individuals are confounded by the likeness of two people and accept that.

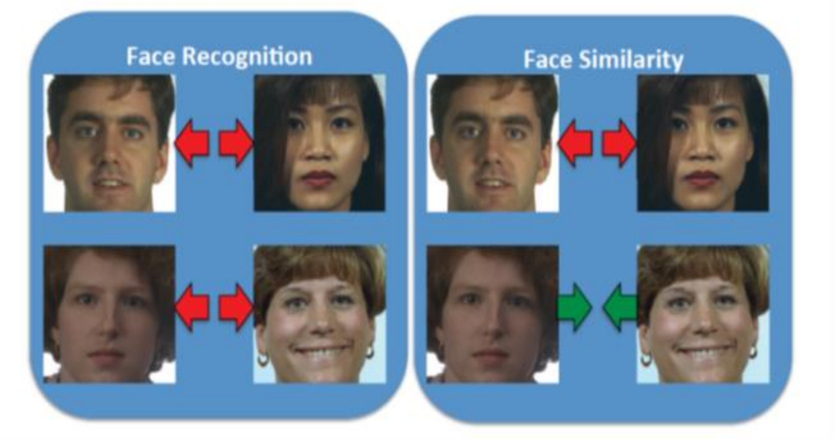

Figure 1: Difference between face identification and face similarity [7]

The individual looks comparable to somebody living in their neighborhood. There may be a few highlights of that individual which cause similitude. In some cases, it could be expensive to blend up to mix up somebody with another individual like somebody did something extraordinary but rather than giving him compensate they gave it to others. Similarly, in the event that somebody did awful rather than him, the police caught others. So face identification algorithms are trained with classification Neural Networks to extract features and select the most effective features for resemblance prediction.

As shown in Figure 2, there are three pictures of the genuine person and three pictures of their look-Alike. Face Recognition algorithm may fail in this case as these algorithms work on features or texture base models to recognize a face from given images so it may recognize a look-alike person as Genuine and genuine as look-Alikes.

Figure 2 (i): Genuine (ii) look-alike of Genuine [8].

2. LITERATURE REVIEW

Recently, many face detection models have been created using deep learning. Zhang et al. [1] introduced the Cascade Multi-task Convolutional Networks as a solution to this problem. The framework comprises three distinct subnetworks, namely the Proposal Network (P-Net), the Refine Network (RNet), and the Output Network (O-Net). Initially P-Net presented a collection of positive candidates and subsequently forwards them to R-Net for further enhancement. The resultant outcome is then passed to O-Net in order to produce the final results. The authors claim an average precision rate of 0.607 when evaluated on the Wider Face dataset [2].

Yolo5Face a very famous face detector works at a very high speed introduced by Qi et al. [3]. YOLO5Face builds upon the YOLOv5 model [4] with some minor adjustments, including the incorporation of a facial landmark regression with five key points and the utilization of Wing loss. The authors achieved notable outcomes in the Wider Face simplicity, medium, and hard sets. Their average mean precision scores were respectively 0.936 (easy), 0.915(medium) and 805 (hard). Face recognition is an important component of face identification and verification over time, several deep learning models have been created to tackle this challenge. Among them, Schroff et al. [5] proposed FaceNet, a deep convolutional network whose underlying architecture is based on Google Net. Face verification and face identification are based on the feature vectors produced by FaceNet, which are used in conjunction with ML techniques such as KNN, SVM etc. Liu et al. [6] applied inheritable Fisher vectors to examine kinship relations based on facial features. The cosine distance is the measure of presence of relation between two images in this research. Smaller the distance between vectors points towards the presence of a relation between two images and larger the distance between vectors points towards the absence of a relation between two images.

Wang et al. [7] introduced approach to learn similarity metrics straight from images by utilizing DL techniques. Also, a systematic triplet sampling algorithm was proposed to learn the model by assigning a stochastic gradient descent the outcomes demonstrated that the suggested algorithm outperforms other models relying on hand-crafted features.Rosenfeld et al. [8] conducted experiments on Totally-Looks-Like (TLL) Dataset which contains six thousand sixteen images in pairs from the wild. They tried to reproduce the pairing via a feature extraction using DCCN (Deep Convolutional Neural Networks). The results showed that algorithm did not perform well in terms of reproducing similar pairs as they were selected by humans.

Sadovnik et al. [9] proposed the new task to recognize face similarities between a pair of faces. They propose a new dataset to find facial similarity and proposed the Lookalike networks which classify similar faces directed similar face classification and at the same task it can perform well for face recognition.Lamba et al.[10] employs three crease commitments in their research: firstly, they analyze how well humans can recognize look-alike appearances in given pictures. Furthermore, they compare human recognition results with ten existing face recognition algorithms, and at long last, proposed an algorithm to improve the face verification accuracy. This proposed calculation yields altogether moved forward execution on the look-alike database.

The goal of Davis et al. [11] work was to employ computer vision techniques with the purpose to find ones, a celebrity look-alike. They used Convolutional Neural Network with 16 Convolutional layers with relu activation functions on the IMDB-WIKI dataset. A scoring method was used to evaluate, a lower score indicates a closer distance and is thus a better score.

Rustam et al. [12] used Fuzzy Kernel C-Means. The system uses the Radial Basis Function Kernel. Chi-Square is used to reduce the number of features that are 2 extracted from images. Their proposed classification method for recognizing faces that look alike has made better accuracy than other ways from previous studies.

Faradina and others [13] used SVM the way they sorted things out in this study. They added two types of kernels, function kernel and polynomial kernel to it afterward. For recognizing who look-alike's faces are, SVM has correctly given high accuracy levels. Khodabakhsh et al. [14] used Convolutional Deep Neural Networks are used for face detection which prove to be helpful in capturing every frame’s facial landmarks. The normalized landmarks in each video are turned into vectors. This methodology has improved detection of individuals who were not recognized by the system before, made possible through a simple calculation between behaviors vectors derived from matching vide. In addition, since the network deals only with landmarks it does not depend on any physiological similarity between pupils. Further, landmarks exhibit lower sensitivity to perturbations and quality issues than other features for instance optical motion fields. This characteristic guarantees their extraction with certainty.

Su et al [15], intoduced celebrity face matching system comprises three main steps: facial detection, face alignment and also using the techniques of human perception. A major issue that surfaces in this research is identifying the level of correlation between two faces which have got no similar labels to help gauge different faces. A multitude approaches are utilized during face detection include MTCNN, LFFD, Retina Face, Center Face YOLO5 Face and DB Fac. Evaluation of the performances of these models was done on Wider Face dataset. The classes loss utilized for face detection are focal lose, bounding box lost and land mark regression Lose. For classification, focal loss is used; bounding box loss refines detected face bounding boxes through the use of a regression based on bounding box regression and landmark-regression losses serve as means to refine positions associated with facial landmarks which have been determined.When one discusses face alignment, which consequently aims at transforming detected faces into canonical forms; a camera captures various pictures taking account of different angles and distances with varying poses. At such occasions, faces are not parallel as regards to rotation; scale and translation which makes alignment the focal principal in order of aligning crucial facial features. To match faces, two methods are employed: the two methods are K Nearest Neighbor’s (KNN) and Mean Cosine Similarity.

3. METHODOLOGY

In Lookalike face recognition IMDB-wiki dataset performs as source data.VGG architectures, including models such as VGGface, VGG16, and VGG19, are employed for feature extraction. The various metrics including cosine similarity, Manhattan distance and KNN Euclidean distance are employed to determine the resemblance between the input image and those within the IMDb dataset. Upon receiving the image, the system encompasses a range of preprocessing procedures. This entails re-scaling the image to the needs of VGG models for its inputs which take the form of 224x224x3. Then the face recognition models are fed the preprocessed image. The models excel in crucial information extraction such as nose, eyes and other facial parts. The architecture of the deep neural network consisting of convolutional layers as exemplified by VGGFace is purposely build to self-learn the hierarchical representations of the features, which are assembled from the data. The lower layers are concentrated on extracting low level features such as edges, corners and textures. higher in the hierarchy, mid-level layers comprehend the more intricate patterns as well as attributes, for example, the subtleties of how facial elements are aligned to each other (e.g., eyes, nose, and mouth). The top layers extract the main high level features necessary for face recognition including, structural features, identity information and discriminant information. This procedure produces vectors which are further utilized in the calculation of the resemblance among various faces. For this purpose we use cosine similarity, KNN Euclidean distance and Manhattan distance metrics. In the IMDB-wiki dataset, some pictures don't have any faces and some pictures have multiple faces so we can discard them. We represent items present in the dataset in the form of vectors

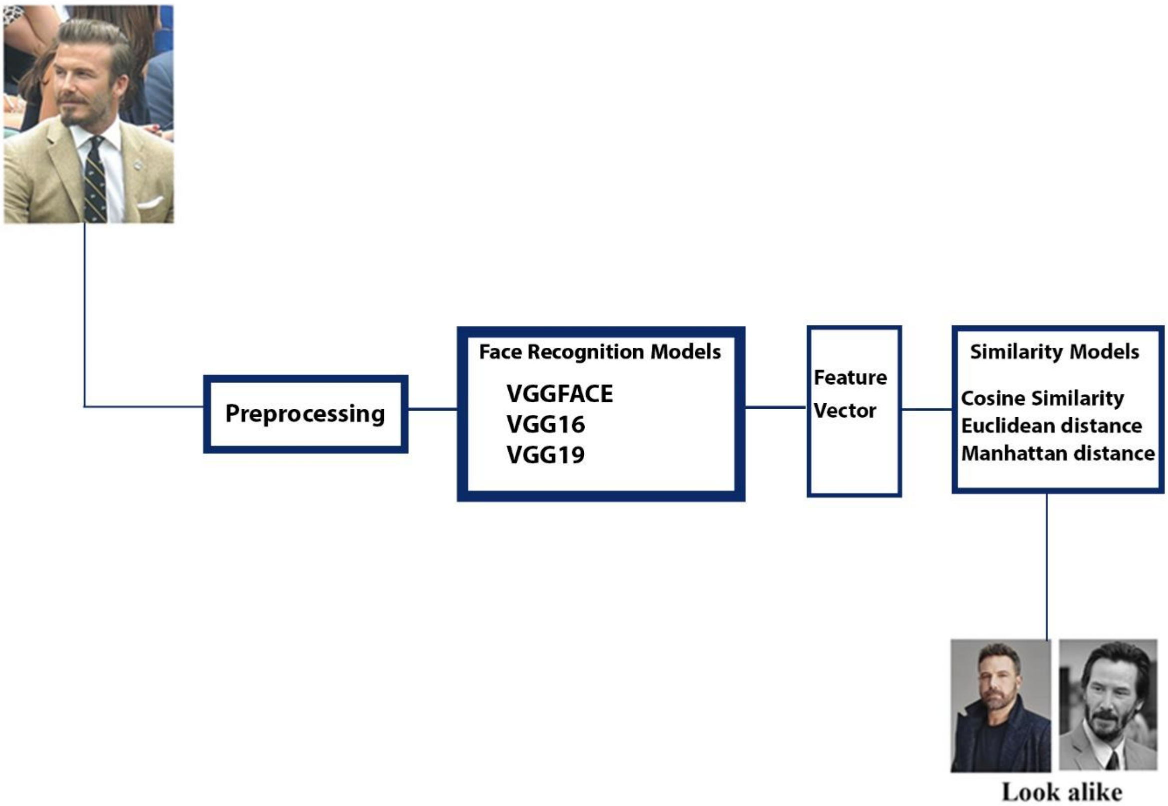

Figure 3 Workflow of Lookalike face detection.

3.1 Face Recognition Models

For look-alike face recognition, we utilized three renowned VGG face networks, namely VGGface, VGG16, and VGG19.

1. VGGFace

VGGFace is one of the best-known and widely applied model for face recognition developed at the University of Oxford by the Visual Geometry Group (VGG). The VGGFace model is to be for face verification. It calculates embeddings from a face image and it makes a comparison of them with the embeddings calculated from another face given. Another set of measures like cosine distance, Euclidean distance, etc supports this. A face-match decision is achieved by measuring the evaluated distance against the predefined threshold.

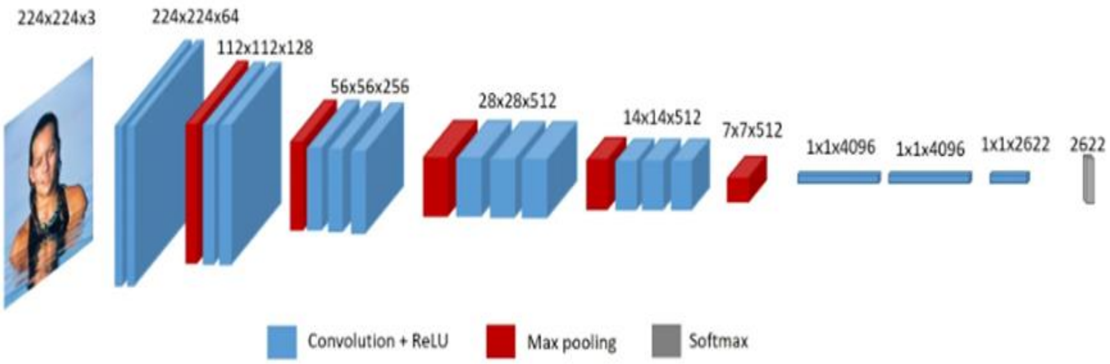

Figure 4 VGGFace Architecture

Figure 4 gives the information about the layers of VGGFace architecture VGGFace is comprised of 38 layers which comprise a series of convolutional layers, VGGFace is very easy to train and gets good results in challenging scenarios.VGGFace has been trained on an extensive dataset comprising 2.6 million images of faces.

2. VGG16

In 2014, VGG16 was developed at Oxford University by researchers Andrew Zisserman and Karen Simonyan. It was designed for two main purposes: includes the task of object localization, i.e. the detection of objects in images, and image classification. Notable is the VGG16 precision of 92% on the ImageNet dataset. Nevertheless, the kernel size filters being larger with 13 convolutional layers led to AlexNet overtaking VGG16 eventually.

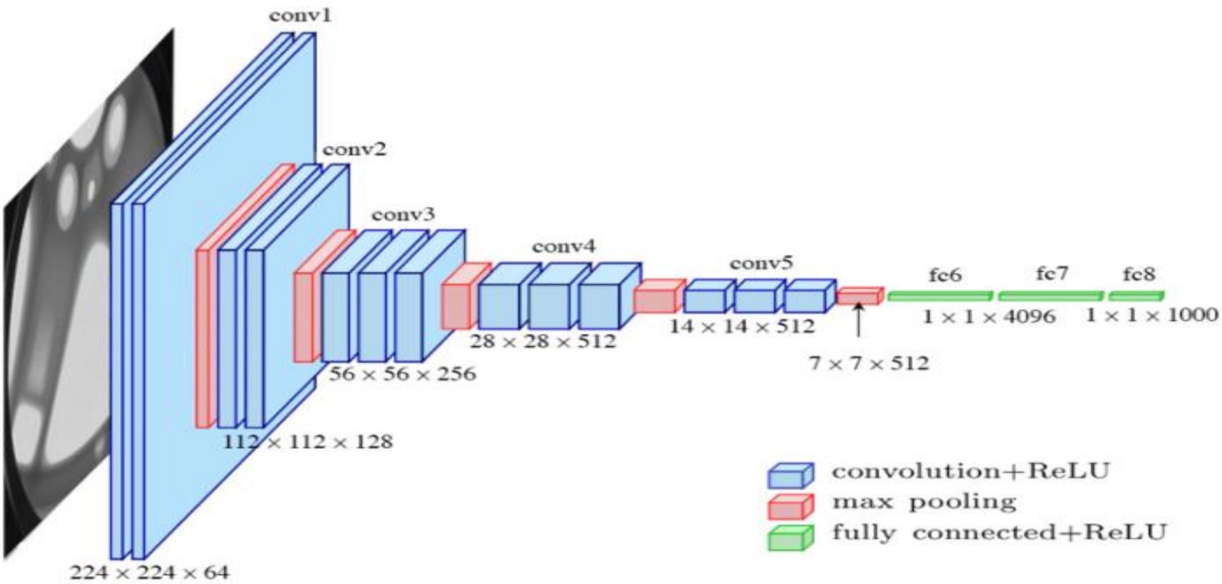

Figure 5 VGG16 Architecture

Fig 5 shows the architecture of VGG16 Network.VGG16 network consist of 16 layers with trainable weights, comprising thirteen convolutional layers and three fully connected layers.

3. VGG19

VGG19 comprises 19 layers in total, with 16 of them being convolutional layers, three fully-connected layers, five max pooling layers, and a softmax layer. The model is built for image classification into one of 1000 object classes. The VGG19 outperforms the VGG16 and the recognition of partial faces is considerably improved.

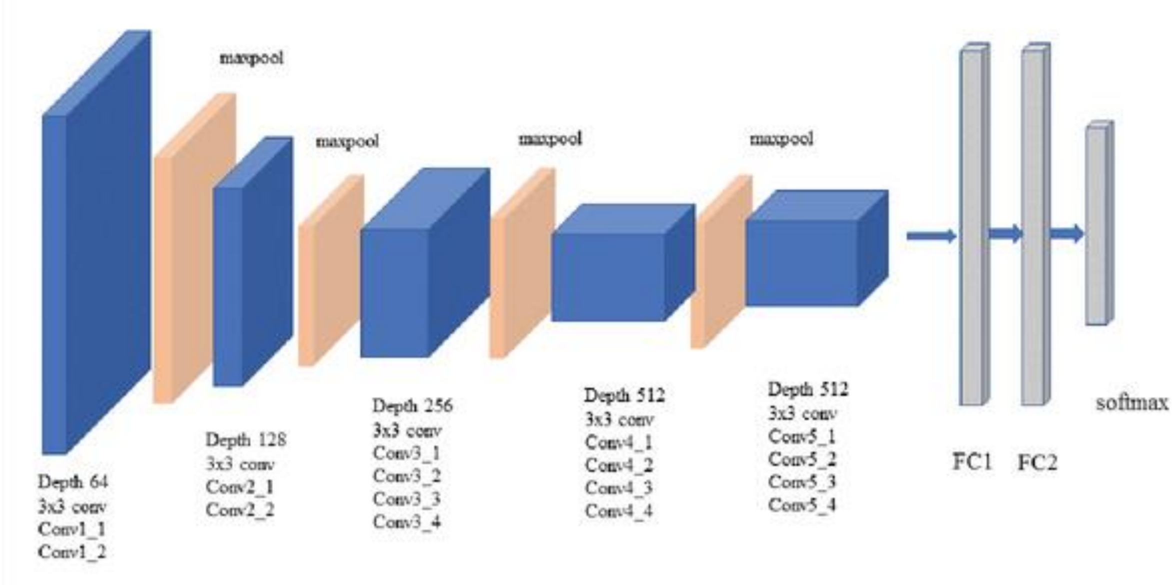

Figure 6 VGG19 Architecture

Figure 6 shows detailed information about the architecture of the VGG19 model, featuring a convolutional sequence, fully connected, and max-pooling layers.

3.2 Similarity Models

For Lookalike face recognition after extracting useful features in the form of vectors from the images. The vectors are given as input to models to compute the likeness between the images. We used three different similarity models which are KNN Euclidean distance, Cosine Similarity, and Manhattan distance.

For every similarity model a threshold is defined and then the similarity is compared with it to classify the pair of images as similar or different. The cosine similarity defines the likeness between the inner product of two non-zero vectors. It can be simply calculated as per below.

$$Similarity = cos\ (\theta) = \frac{x\ .\ y}{||x||\ ||y||} = \frac{\sum_{i=1}^{n} x_i\, y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \qquad \text{Equation 1}$$

Where x and y represent the dot product of vector x and y.

Euclidean distance is considered as the distance among two vectors, represented as x and y, is expressed as:

$$Euc\ (x, y) = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2} \qquad \text{Equation 2}$$

Manhattan distance is characterized as the distance among two vectors. Computed along axes at right angles is given as:

$$Manhattan\ distance\ (D) = (|a_1 - a_2| + |b_1 - b_2|) \qquad \text{Equation 3}$$

4. RESULTS

The input and output image are compared to show their similarity as a percentage. Our model was trained on the IMDb-wiki dataset and for testing purposes, we also used the IMDb-wiki dataset. The 80% of the dataset is allocated for training, while the remaining 20% is reserved for testing. The IMDb-wiki dataset contains more than 500,000 images, however we used 5000 images for our model. The model demonstrated a 95% overall accuracy on the images. We employed three distinct models, namely VGGFace, VGG16, and VGG19, to extract useful features. Additionally, we applied three different similarity models: KNN Euclidean distance, Cosine Similarity, and Manhattan distance. Notably, the maximum accuracy was attained using the VGGFace model in conjunction with the cosine similarity function.

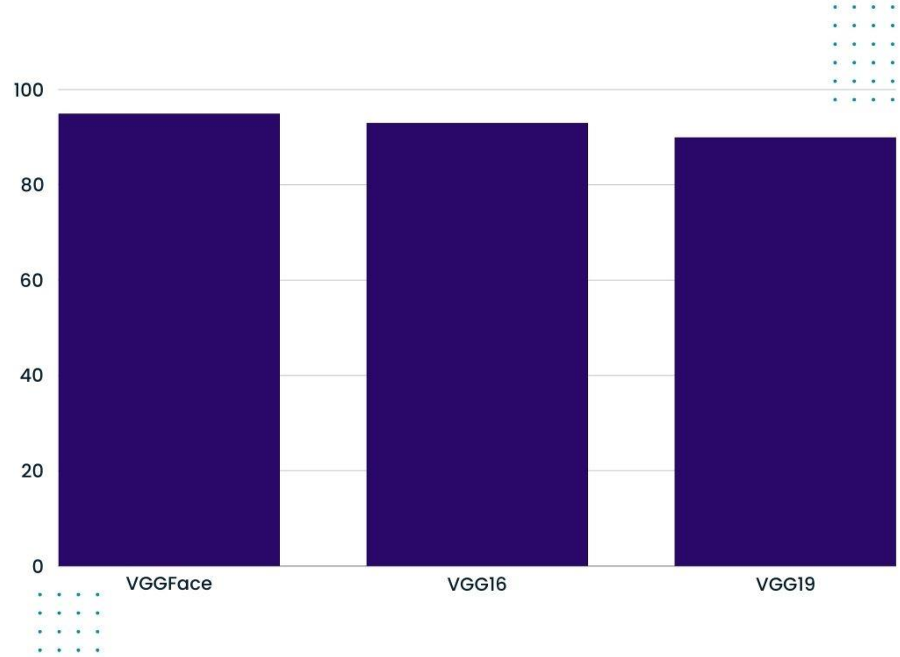

Figure 7 Accuracy of the Model Using Cosine Similarity

Figure 7 depicts the model accuracy using cosine similarity, where VGGFace exhibits the highest accuracy compared to other models such as VGG16 and VGG19.

We used the same photographs to compare fifty outcomes from the initial milestone implementation to fifty results from our enhanced implementation. We next divided each score by the highest score obtained throughout all 100 trials to obtain values between 0 and 1, and we multiplied this result by 100 to obtain a score ranging from 0 to 100. We contrasted each groups' mean and median distances. In the revised implementation, the median score increased from 25 in the initial implementation to 17 in our observation. A lower score is preferable because it denotes a shorter distance, with a distance of 0 denoting the same face.

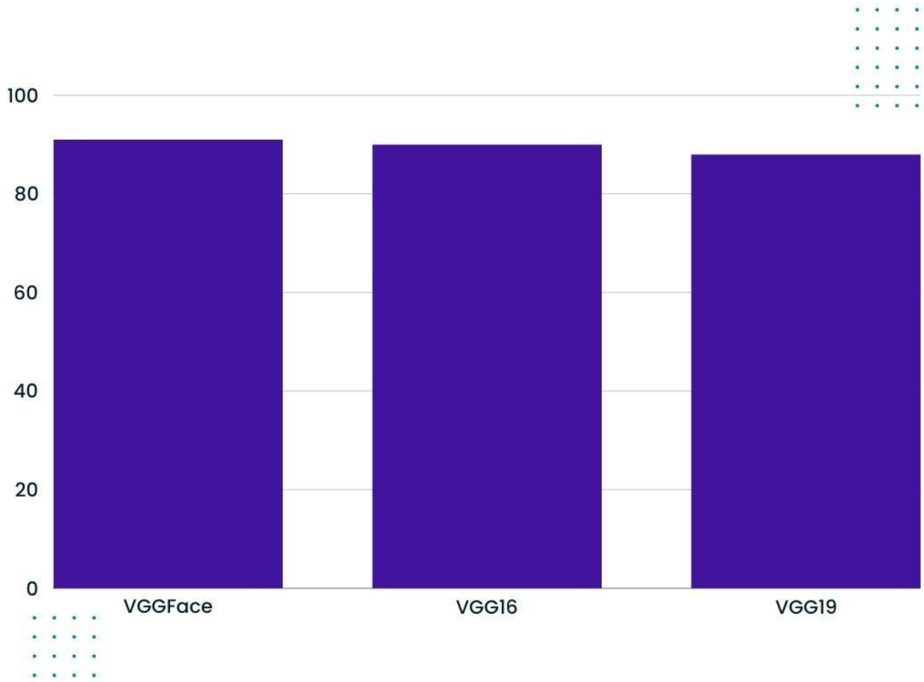

Figure 8 Accuracy of the Model Using Euclidean Distance

Figure 8 depicts the model accuracy using Euclidean Distance, where VGGFace exhibits the highest accuracy compared to other models such as VGG16 and VGG19.

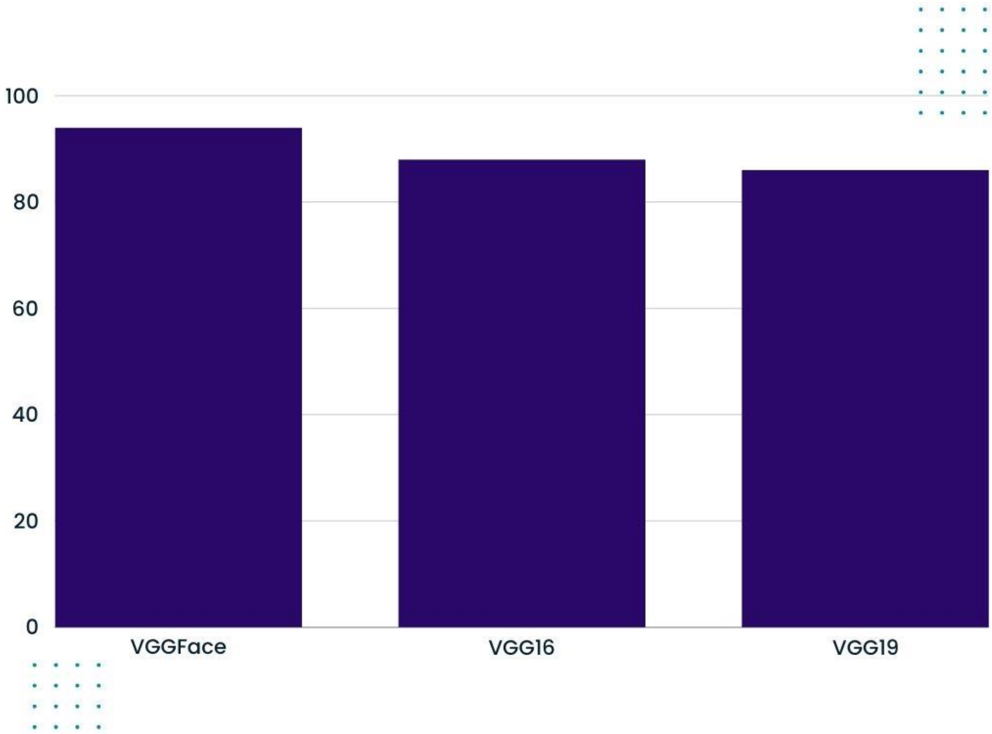

Figure 9 Accuracy of the Model Using Manhattan distance

Figure 9 illustrates the model accuracy using Manhattan distance. In this context, VGGFace demonstrates the highest accuracy among the models, with VGG16 following closely at 89 percent accuracy.

Table 1: Accuracy of face recognition Model.

Similarity Model	Accuracy	Face Recognition Model
KNN Euclidean distance	92%	VGGFace
Manhattan distance	93%	VGGFace
Cosine Similarity	95%	VGGFace

The table 1 presents the top-performing face recognition models based on KNN Euclidean distance, Cosine Similarity, and Manhattan distance. According to Table 1, it is evident that VGGFace achieves the highest accuracy when compared to the other models.

We experimented with evaluating the scoring mechanisms on images of the same individual from the same viewpoint and observed that it came up with a distance of 5, suggesting that it does have some inaccuracy but is a fairly accurate indicator of related photographs. Figure 6 illustrates how the algorithm gives equal weight to skin tone, facial features, and hair colour. To locate your celebrity clone, you don't have to completely match in every category, though.

Figure 10: People who resemble David Beckham.

David Beckham facial photographs are input into system, shown in Fig 10, the results of system are remarkably uniform, signifying that David Beckham bears the closest resemblance to Ben Affleck & Keanu Reeves. it implies that system for matching celebrities is capable of encoding David's countenances into a face cluster situated in close proximity to Keanu Reeves face clusters and Ben Affleck in the recognition dataset.

5. CONCLUSION

In conclusion, while our model has shown good performance, there is still room for improvement. To enhance the accuracy of our doppelganger generation system, future steps could include increasing the size of the testing set as we expand the dataset, which will provide more celebrities as reference points. Additionally, we could acclimatize our evaluation criteria to place greater emphasis on different facial features to further enhance the realism of the generated doppelgangers. With these enhancements, we can continue to refine our system and create even more realistic and accurate celebrity doppelgangers.

REFERENCES

[1]Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE signal processing letters, 23(10), 1499-1503.

[2] Yang, S., Luo, P., Loy, C. C., & Tang, X. (2016). Wider face: A face detection benchmark. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5525-5533).

[3] Qi, D., Tan, W., Yao, Q., & Liu, J. (2022, October). YOLO5Face: why reinventing a face detector. In European Conference on Computer Vision (pp. 228-244). Cham: Springer Nature Switzerland.

[4] Utralytics, "YOLOv5," [Online]. Available: https://github.com/ultralytics/yolov5

[5] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823)..

[6] Liu, Q., Puthenputhussery, A., & Liu, C. (2015, September). Inheritable fisher vector feature for kinship verification. In 2015 IEEE 7th international conference on biometrics theory, applications and systems (BTAS) (pp. 1-6). IEEE.

[7] Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., ... & Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1386-1393).

[8] Rosenfeld, A., Solbach, M. D., & Tsotsos, J. K. (2018). Totally looks like-how humans compare, compared to machines. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 1961-1964).

[9] Sadovnik, A., Gharbi, W., Vu, T., & Gallagher, A. (2018). Finding your lookalike: Measuring face similarity rather than face identity. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 2345-2353).

[10] Lamba, H., Sarkar, A., Vatsa, M., Singh, R., & Noore, A. (2011, October). Face recognition for look-alikes: A preliminary study. In 2011 International Joint Conference on Biometrics (IJCB) (pp. 1-6). IEEE.

[11Davis, C., & Jacquez, A. Finding Your Celebrity Look Alike.

[12] Rustam, Z., & Faradina, R. (2019, May). Application of Fuzzy Kernel C-Means in face recognition to identify look-alike faces. In Journal of Physics: Conference Series (Vol. 1218, No. 1, p. 012045). IOP Publishing.

[13] Rustam, Z., & Faradina, R. (2018, November). Face recognition to identify look-alike faces using support vector machine. In Journal of Physics: Conference Series (Vol. 1108, No. 1, p. 012071). IOP Publishing.

[14] Khodabakhsh, A., & Loiselle, H. (2020, September). Action-independent generalized behavioral identity descriptors for look-alike recognition in videos. In 2020 International Conference of the Biometrics Special Interest Group (BIOSIG) (pp. 1-6). IEEE.

[15] Su, F., & Sinnott, R. O. (2022, December). Celebrity Face Matching Using Deep Learning. In 2022 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE) (pp. 1-6). IEEE.