Facial recognition using Siamese Networks with score level fusion

Bharath Reddy Mattapally, Sree Vardhan Ginjupalli, Lokesh Kumar Reddy Guduru
{ginjupallis@usf.edu, bharathreddy19@usf.edu, lguduru@usf.edu}

23-11-2022

Abstract

The task of making a positive identification of a face in a photo or video, against an existing database of faces is known as facial recognition. It starts with detection, which distinguishes human faces from other objects in the image, and then moves on to identification of those detected faces. While early methods used mathematical models for recognition, Face recognition has advanced dramatically as a result of the advancement of deep convolutional neural networks (CNNs). CNNs employ multiple convolution layers to learn data representations with varying degrees of feature extraction. In this paper, we employ a Siamese network style facial recognition paradigm with a score level fusion. The dataset for this task has been manually constructed. Inception ResNet V2 model was used for extracting 128 facial feature embeddings for each image in the training dataset. These embeddings were then used for recognizing the persons in the test set. A preliminary accuracy of 84% was achieved on the validation dataset with this method. We further tried to improve the accuracy by performing a score level fusion. Furthermore, our work focused on three preliminary research questions that directed this work. The impact of Gender, inversion (Horizontal & vertical flipping) and Grey scaling the test images, on predictions were studied which will be discussed further.

1 Introduction

Physical traits or biological measurements that can be used to identify people are known as biometrics. Researchers claim that the shape of an ear, the way a person sits and walks, distinct body odors, veins in one's hands, and even facial contortions are all unique identifiers. These characteristics help to define biometrics. While biometrics have other applications, they are most commonly used in security, and they can be classified into three types:

- Biological biometrics
- Morphological biometrics
- Behavioral biometrics

Facial recognition falls under the umbrella of biometric security. Voice recognition, fingerprint recognition, and retina or iris recognition are other examples of biometric systems. The technology is primarily used for security and law enforcement, but there is growing interest in other applications. It is one of the most advanced forms of biometric authentication, capable of identifying and verifying a person using facial features from a database image or video. Increased investments in facial recognition technology have been made in recent years. In 2022, venture funding for facial recognition start-ups is expected to skyrocket. With advancements in this technology, new use cases and business models in advertising, healthcare, security, proctoring, airports, and other fields have emerged.

Face recognition employs AI algorithms and machine learning to detect human faces in the background. The algorithm typically begins by looking for human eyes, then brows, nose, mouth, nostrils, and iris. Following the capture of all facial features, additional validations using large datasets containing both positive and negative images confirm that it is a human face. Some of the most common facial recognition techniques are feature-based, appearance-based, knowledge-based, and template matching. Each of these methods has benefits and drawbacks. The ability to recognize faces using feature-based methods depends on features like the eyes or nose. The results of this method may vary depending on noise and light. Furthermore,

appearance-based methods match the characteristics of face images using statistical analysis and machine learning. A face is recognized using predefined rules in a knowledge-based approach. Given the effort required to define well-defined rules, this could be difficult. Template-matching methods, on the other hand, compare images to previously stored face patterns or features and correlate the results to detect a face. This method, however, does not account for variations in scale, pose, and shape.

One of the most well-known algorithms for face detection is the Viola-Jones Face Detection technique, also known as Haar cascades. Haar cascades were popular long before deep learning and are one of the most widely used techniques for detecting faces. To begin, the classifier is trained using a set of positive images (images of faces) and a set of negative images (images with faces). The features are then extracted from the images. Figure 1 depicts some of the features extracted from images with faces. We search

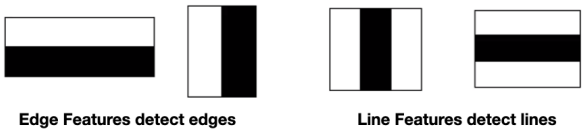

Figure 1: Edges in a Haar Cascade.

for the existence of the different features that are often present on human faces, like the eyebrow, where the area above it is lighter than the area below it, in order to identify faces from an image. An image is considered to possess a face when it combines all of these elements. Fortunately, OpenCV can perform face identification out of the box using a pre-trained Haar cascade, along with additional Haar cascades for detecting other objects.

Face recognition is a lot more difficult process than face detection, and it is one that academics are very interested in because they are constantly trying to increase the recognition's accuracy. Support vector machines can be used for facial recognition (SVM). A class of supervised learning techniques that can be applied for classifications are known as SVMs. SVM identifies the line that partitions classes by the greatest distance possible. The fundamental goal of SVM is to efficiently draw a boundary between two or more classes. We can utilize the line that is drawn to demarcate the classes later on to forecast forthcoming data. For instance, we can now use the dividing line as a classifier to determine if an unknown species is a dog or a cat, given its snout length and ear geometry.

However, due to SVMs limited representational capabilities, a CNN seems to be deemed fit for applications such as image recognition and is frequently employed in facial recognition. CNNs, which are distinguished by their hierarchical architecture for assembling pixels into different but similar representations, have greatly enhanced state-of-the-art performance and facilitated effective real-world applications. Several filters are used in a convolution operation in a typical machine vision application to detect the numerous components of the image. The early layer seeks to concentrate on general features, whereas the later layers seek to find extremely particular features. A CNN learns its values via training on a specific training set, which yields the data for the different filters in every convolutional layer. We have a distinct set of filter values at the conclusion of training that are utilized to identify particular features in the data. We can forecast what is in a subsequent image by applying this collection of filter values to it and using that projection to determine whatever is in the new image.

2 Literature Survey

In [3], they demonstrate the FaceNet system, which effectively learns a projection from face images to a condensed Euclidean space whose distances directly correlate to a metric of face similarity. FaceNet embeddings serve as feature vectors, making it simple to execute facial recognition after this space has been created. In contrast to other deep learning techniques, their methodology directly optimizes the embedding itself using a deep convolutional network, rather than passing it via a bottleneck layer beforehand. They employed online triplet mining to create triplets of substantially aligned matching and non-matching face patches for training. Greater representational efficiency is achieved as a result of this method.A harmonic triplet loss and the notion of harmonic embeddings were also proposed, which characterized several face embeddings (generated by various networks) that are compatible with one another and allowed for comparative evaluation. They used their strategy to achieve cutting-edge facial recognition performance with just 128 bytes per face. Their algorithm obtained a new record accuracy of 99.63% on the commonly used Labeled Faces in the Wild (LFW) dataset. It received 95.12% on YouTube Faces DB. On both of these datasets, their technique reduced the error rate by 30% when compared to the

best results that had previously been reported.

In [7], they presented the notion that deep CNNs' traditional softmax loss typically lacked discriminating. They researched a number of loss functions, including angular softmax loss, big margin softmax loss, and center loss, to solve this issue. However, the common goal of all of these enhanced losses was to increase interclass variation and decrease intra-class variance. Thus, they realized the same concept from a novel perspective and suggested a new loss function, called large margin cosine loss (LMCL). In order to minimize radial fluctuations, they deliberately rebuilt the softmax loss as a cosine loss by L2 normalizing both the feature and weight vectors. From there, a cosine margin term was added to further optimize the decision margin in the angle space. By virtue of normalizing and cosine judgment margin maximization, minimal intra-class variance and maximal inter-class variance was attained. CosFace is the name given to the final model developed using LMCL. The effectiveness of their suggested strategy was proven by extensive experimental assessments on the most well-known public-domain face recognition datasets, including MegaFace Challenge, Youtube Faces (YTF), and Labeled Face in the Wild (LFW), where they attained state-of-the-art performance.

In [1],they suggest that it's still difficult for them to design a system to choose the optimal threshold for actual use. To increase the recognition accuracy, they create an adaptive threshold mechanism technique. They suggest a registration and recognition-based two-operation system. A feature vector (or embedding) is extracted from an input facial image during the registration process using a deep neural network. The face may belong to a user who is already registered with the system or to a brand-new user who has never been registered before. Each time a face is registered, a threshold is assigned to it, and that threshold is then applied to all the other registered faces accordingly. Given a query image, they extract its feature embedding and calculate the similarity scores between it and other embeddings that have been previously recorded.

In [2], they claim that in numerous real-world face recognition settings, the training set is shallow, with only two face photos being supplied for each ID, and lacking intra-class variety compared to deep face data. As a result, the feature dimension may collapse, which might easily cause atrophy and over-fitting in the learnt network in the collapsed dimension. They made an effort to solve the issue by inventing a cutting-edge training technique called Semi-Siamese Training (SST). Using effective optimization on shallow training data, a pair of Semi-Siamese networks that make up the forward propagation structure compute the training loss using an updating gallery queue. Numerous tests revealed that the suggested strategy enhanced learning for shallow face and traditional deep face data.

There is a sizable body of facial recognition and verification research. We will only briefly touch on the most pertinent current work because a review of it would go beyond the scope of this paper.

All of the research by [4,6,8] use a multi-stage complicated system that combines the results of a deep CNN model with PCA for dimensionality reduction and then use an SVM for classification. In order to "warp" facial data into a conventional frontal view, Zhenyao et al. [8] use a deep network. They then learn a CNN that categorizes each face corresponding to a defined set. PCA on the output of the system combined with a group of SVMs is utilized for face verification.

Our work draws inspiration from these CNN based embedding extraction and similarity prediction to predict the faces.

3 Datasets

A two-phase procedure was used to collect the dataset. We encouraged students from the University of South Florida from a variety of ethnic backgrounds and creeds to post a picture every day for one to two weeks in the first phase. It was made sure that they changed while doing so and took photos under different lighting circumstances both during the day and at night, wearing various costumes. In order to replicate data from the actual world, they were also asked to include images of them wearing and without wearing their spectacles. A number of them even cut their facial hair, and they contributed numerous realistic photographs. A standard data folder structure that is appropriate for picture recognition tasks was assembled and used to store the information collected from 56 pupils.

The objective of the second phase was to remove any background elements that belonged to other persons or things and only accommodate the face features of one person in the photograph. The face in each photograph had to be found in order to accomplish this. The Haar Cascade classifier was used to determine the coordinates of the relevant face. Using

these coordinates, the base image was then cropped to contain only the pixels corresponding to the student's face. The photos were kept in the same directory structure. All of the photos had three channels and were RGB-based.

4 Methods

We experimented with different models, which would be discussed in the experiments section and finalized the best approach as our methodology. In our methodology, we used a Siamese network style embedding similarity estimation metric to predict the faces. The model used was Inception Resnet V2 as discussed in [5]. Figure2 depicts the architectural details of this network.

This network was written from scratch and the pre-trained weights trained on ImageNet dataset that were already published was used to be loaded in the model. This allowed the Tensorflow to pass inputs of different shapes to the model, which otherwise wouldn't have been possible if used the Tensorflow in-built Inception Resnet V2 model. The pre-trained model was trained as such that it extracts 128 meaningful feature embeddings of an image. The same principle was used to get it to predict the 128 feature embeddings corresponding to each image in the training set.

4.1 Train-Test Split

The dataset was divided into sets of 20 images/4 images per person as the train test split. This meant that a ratio of 0.9 and 0.1 was use to split the dataset. This was done using a random seed of 1, so that the results could be replicated every time.

4.2 Data Pre-processing

The data from the train set was pre-processed using the ImageDataAugmentor class from tensorflow. The data was initially standardized and then passed into the data augmentor. The augmentor ensured that Horizontal, vertical flips were performed at random and this augmented image was passed to the model to be trained. The idea behind such an augmentation was to inculcate generalization capabilities to the model such that it does learn to extract the facial characteristics in general and not just the representation of a single style.

4.3 Training

As it is known that Siamese styled networks use the same model wights for both training ad validation, the same approach was followed here. The idea is to get the Inception Resnet V2 model to predict the 128 feature embedding of every image corresponding to a single person in the train set. These embeddings were averaged upon to deliver the final feature representation of that corresponding person. The name of the person and their averaged embedding was stored in a dictionary. This dictionary is the encoding that our model will refer to while making a prediction for a face. This ends the training phase. Note, that unlike traditional deep learning methods, the model is not optimizing it's weights but rather storing the feature vectors that it has already learned.

4.4 Testing

In order to test the model's performance, the recognition pipeline was to be set up. For this, the images in the test set was sent to the model to have their feature embedding vector predicted. The predicted embedding was then used to calculate it's Euclidean distance from the rest of the encoded embeddings and the minimal distance was identified. This was a sign of similarity, suggesting that the embedding was closely related to the identified person. As such, the name of the person with which the embedding predicted in the test set had the least distance was predicted as the recognized face. Using Euclidean distance, 184 out of 225 images resulted in a correct prediction and the resultant accuracy was 81.77%.

4.5 Score-Level Fusion

Initially, the Euclidean distance was used to estimate the similarity metric between the predicted feature embedding and the encoded feature embeddings to predict the name of the person the face was corresponding to. Later on, we took a weighted average of the Euclidean distance and the Cosine distance to get the similarity metric. Weights of 0.3 and 0.7 were assigned to Euclidean distance, and cosine distance respectively. This kind of fusion further improved the model accuracy from 81.77% to 84%. As such, there were 189 accurate predictions in a test set consisting of 225 images.

4.6 Experiments

We also experimented with different styles of pre-trained networks namely MobileNet, ResNet50 and InceptionResnetV2 from the Tensorflow library. The dataset was interpolated to the required shapes these models were trained on the ImageNet dataset. The face recognition problem was converted into a classi-

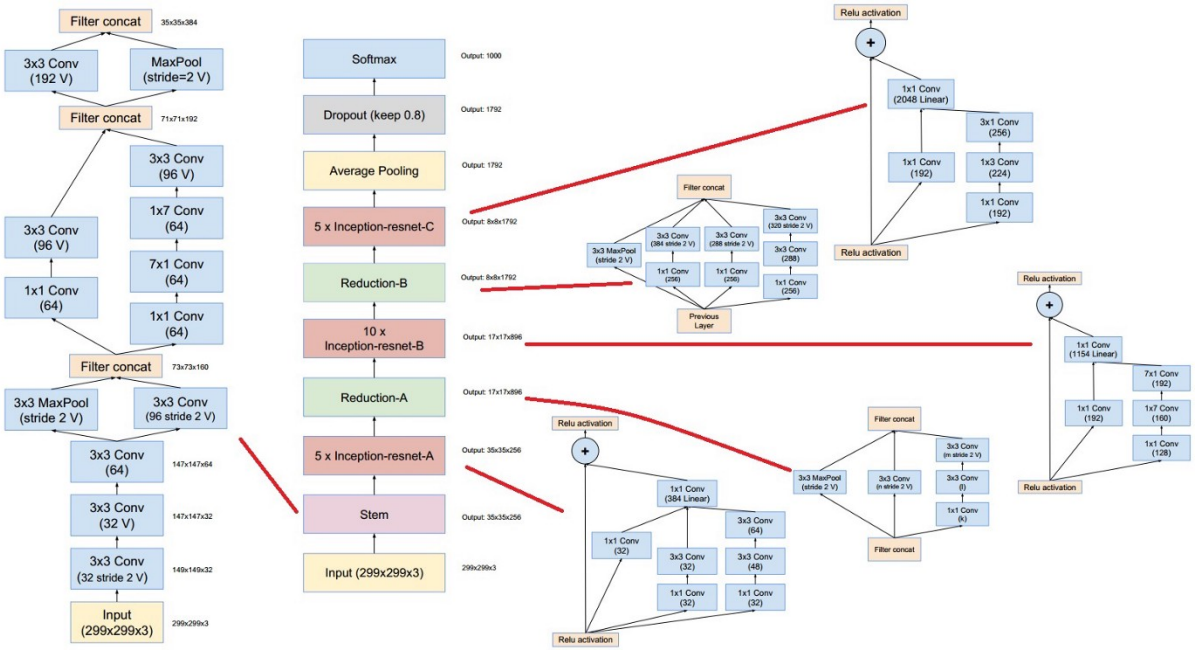

Figure 2: The architectural details of the Inception Resnet V2 Model.

fication problem, and the top layers from these models were removed and replaced with a MLP model with The last dense layer consisting of 56 nodes, as 56 was the number of classes to be predicted. Adam optimizer was used, with a learning rate of 0.01 and early stopping callback with waiting number 5 was scheduled. The training resulted in a model having the best accuracy 0f 72% after which it started to overfit. Hence, these models were deemed unfit for our task. On researching the reason behind it, we figured that these models require faces with low intra-class deviation and high inter-class deviation to perform well. But our dataset had no such resemblances. Hence, these methods were omitted. Training and validation curves are showcased for one such model in Figure3,4.

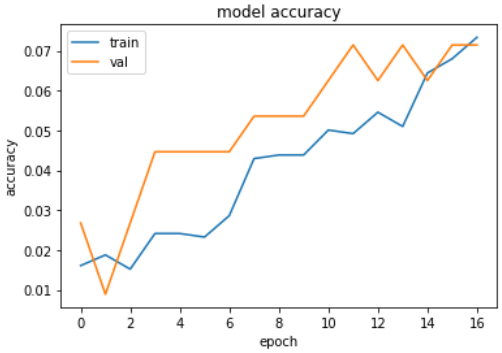

Figure 3: Training and Validation accuracy vs Epochs

5 Research Challenges

Throughout the facial recognition pipeline our idea was to focus and study three research questions.

- The effect of inversion (Horizontal and Vertical Flipping of Test Images) on the Prediction Accuracy.
- The effect of color inversion (RGB to GrayScale) on the prediction accuracy.
- The effect of genders on the prediction accuracy.

Each of these aspects would be discussed to approximately understand the black box learning methodology of our predictor.

Image Flipping Shape inversion didn't have a drastic impact on the accuracy of the model. The model still retained an accuracy of 81.77% when inverted images were passed. This is because the training images were augmented beforehand, where the images were flipped both horizontally and vertically and hence, the impact of inversion is negligible. Additionally, the images were also augmented for a rotation angle of 15 degrees, which further enhanced the generalization capability of the fused model. This

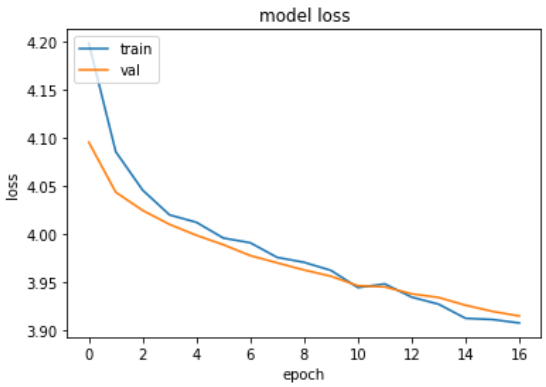

Figure 4: Training and Validation loss vs Epochs

kind of augmentation is seen to improve the performance of the model drastically.

Color Inversion Deep learning models have a constraint that they cannot infer when the channels of the image are different from the channels of the image used while training the model. As the model was trained on RGB images, it is inherently not possible to infer Gray scaled images with the model trained on the RGB images. As a workaround, the test images was first converted into gray scale and then stacked three time to replicate the shape of the RGB images. The important aspect to be noted is that, since the gray scaled image is repeatedly stacked in three different channels, it is infact different from the actual RGB images. This difference was used to anticipate the effect of color inversion. Similar to image flipping, color inversion didn't drastically impact the model performance. The drop in accuracy was 2%. This can be attributed to the fact that the facial embeddings extracted from the image is irrespective of the channels and is more generalized on the complex feature the pixels hold. Spatial information isn't a important detail in facial recognition tasks and many works in facial recognition, reduce the channels to 1, for faster performance. Our work supports the above mentioned logic in different papers.

Gender In order to examine this aspect, the data had to be manually created identifying the gender of all the images in the train set and test set. For simplicity, the genders were assumed to be Male and Female. The labelling in no way imposes the gender to anyone in the dataset. The data labelled with gender was used to derive the metrics of the model performance.

Interestingly, the model when made 36 incorrect predictions only 4 of them corresponded to female. While, 32 incorrect predictions were corresponding to the Male gender. And hence, such a bias were observed in the predictions. This upon further research was attributed to the facial hair and improper images in the test set. Since, women had no facial hair, their facial recognition wasn't a challenge. Figure5, 6 depicts the inconsistencies in the train and test set for a single class id. Such kind of inconsistencies showcases the generalized dataset we have collated and is in relevance to the real world data. Hence, an accuracy of 84% seems fair and decent.

Figure 5: An image in the Train set

6 Conclusion

Facial recognition still poses great challenges despite technological advancements in deep learning. Hence, this issue must be addressed properly. The systems aren't very efficient when performed on a custom dataset, which are very close to real world data. It is to be noted that, in real world, facial features such as facial hair is a variable component and the model must be tuned to these changes to perform better. Additionally, in real world scenarios, people can have different aesthetics that can alter the models recognition capabilities. In order to generalize on these aspects, the model has to learn from huge datasets, that incorporate these effects. While

Figure 6: Inconsistency in the Test set

our dataset was significantly small when compared to state of the art datasets, we can have it in our scope to enhance and automate the data collection pipeline, and then experiment with classical deep learning methods to come up with an accurate predictor. While our work used Siamese styled networks, fusion can be done with classical deep learning models by converting the recognition task as a classification task. Furthermore, vision transformers have been gaining traction in the computer vision space especially for recognition tasks. Since, transformer models effectively convert the image space into an embedding space on different patches, it is much likely to provide better results in this challenge. It would be interesting to compare the performance of vision transformers with our existing pipeline for facial recognition tasks.

References

[1] Hsin-Rung Chou, Jia-Hong Lee, Yi-Ming Chan, and Chu-Song Chen. Data-specific adaptive threshold for face recognition and authentication, 2018.

[2] Hang Du, Hailin Shi, Yuchi Liu, Jun Wang, Zhen Lei, Dan Zeng, and Tao Mei. Semi-siamese training for shallow face learning, 2020.

[3] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015.

[4] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust, 2014.

[5] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016.

[6] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[7] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition, 2018.

[8] Zhenyao Zhu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Recover canonical-view faces in the wild with deep neural networks, 2014.