Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-18671-5

Navigating the face recognition: unleashing the power of few-shot learning through metric-based insights

Sushant Jain[1] · Amit Pundir[2] · Sanjeev Singh[1] · Geetika Jain Saxena[2]

Received: 8 September 2023 / Revised: 2 February 2024 / Accepted: 17 February 2024

Abstract

Few shot classification is the task of classifying unseen classes having only a few samples of each of these unseen classes. A traditional approach of using the transfer learning on the unseen data tends to overfit the problem and thus causing challenges in generalization. We present face recognition using few shots by using metric based approach that consists of multiple stages of model training. In the first stage, the foundational model is trained using a general face recognition dataset like DigiFace-1m, which serves as a foundation. In the second stage, a metric learning loss, such as triplet loss, is applied to further refine and optimize the features learned by the model. This two-stage approach enhances the effectiveness of the model for real-time face recognition with limited data samples. The study takes a deeper dive and provides a detailed comparative analysis of different triplet mining strategies for face recognition based on triplet loss learning. These methodologies primarily utilize convolutional neural networks (CNNs) trained with triplet loss and compare different triplet selection techniques, including hard triplets, semi-hard triplets, and offline triplets. Standard datasets like DigiFace-1m, Labeled Faces in Wild (LFW), CelebA, and VGGFace2 are employed for experiments, but the selection of training classes differs between the two stages. This ensures that although the same dataset is used for training, the two networks are trained on different identities. The acid test comes in the form of real-time performance evaluation where support sets composed of a mere two samples each from a pool of 50 unseen classes take center stage. These support sets, along with query sets, are meticulously crafted from previously unexplored datasets. The results are nothing short of remarkable, with reported accuracy surpassing the 70% threshold a resounding triumph in the realm of few-shot learning, promising exciting possibilities for real-world applications.

Keywords Deep learning · Face recognition · Few shot learning · Meta learning

Sushant Jain, Amit Pundir, Sanjeev Singh and Geetika Jain Saxena contributed equally to this work.

✉ Geetika Jain Saxena
gsaxena@mac.du.ac.in

Extended author information available on the last page of the article

Published online: 01 March 2024

1 Introduction

Differentiating between objects with multiple features is one of the most complex tasks. Even the human eye is sometimes challenged to correctly differentiate between different objects solely by comparing certain features of the objects. Face recognition is a highly intricate task that poses significant challenges due to the structural and facial similarities between individuals. This complexity makes it difficult for computer programs to accurately evaluate and distinguish between faces.

Face recognition using AI models has been the subject of numerous studies and experiments. However, despite significant progress, face recognition remains a challenging task due to the structural similarity of facial features. Although facial representations differ across individuals, they often appear closer together in the latent space. Consequently, features or embeddings extracted from trained CNN models for face recognition may exhibit similarity. To overcome this issue and enable CNN models to effectively distinguish between different faces and generalize well, training on a large volume of data is essential. To address the limitation of requiring a high volume of data per identity, the concept of triplet loss-based modeling was introduced. While many studies and experiments have explored the use of the Triplet Loss Function for Face Recognition, there is a lack of comprehensive presentations on the experimental results of different combinations of triplet mining strategies.

Numerous research has been published and several methods and techniques have been proposed and published related to this task. Training a face recognition CNN model is resource-intensive and demands a large amount of data to achieve optimal results. Reliance on high volumes of data underscores the importance of having robust computational resources to meet computational demands. To effectively handle the high volume of data, it is crucial to incorporate a diverse range of feature-rich datasets. This includes masked images, occluded faces, low-resolution images, and more. By introducing such variety, CNN models can capture and learn the subtle differences present in the data, ultimately leading to a comprehensive and distinct representation in the latent space. Manual process of data annotation is an expensive job and can cause high technical debt. Thus, obtaining accurately labeled datasets for face recognition can be challenging due to the requirements of high volume and variety of data.

In order to circumvent the problem of data sufficiency for model training, meta-learning using metric-based learning methodology is used to perform face recognition using few shots. The metric-based loss called Triplet Loss is used on face verification and recognition tasks using a Siamese Network. We used offline and online triplet mining strategies along with the Triplet Loss function by selecting hard and semi-hard triplets. Another factor introduced while selecting hard and semi-hard triplets is selection of negative samples during triplet creation. We used random or best-fit negative sample strategy and performed model training using these combinations. Selection of a triplet must be such that the model learns image embeddings in such a manner that the distance between the anchor and positive image embeddings is closer than the anchor and negative image embedding [1]. Similarity between these embeddings is measured using the cosine similarity metric, while distance is computed using the Euclidean distance. These metrics provide valuable insights into the proximity and relationship between the embeddings, enabling effective comparison and analysis. This work also describes how selection of hyperparameters may influence the outcome.

Our research distinguishes itself by diverging from conventional methods that heavily rely on extensive datasets and deep networks for achieving broad generalization. Recognizing the limitations of this approach, we conducted experiments in scenarios marked by limited data availability. Our study intentionally employs substantially fewer samples per class yet manages to achieve noteworthy levels of generalization. Also, we introduce a novel element to our methodology - the integration of a few-shot learning strategy. This approach not only addresses the challenges associated with limited data scenarios but also significantly reduces the dependency on high computational resources. Consequently, our research stands out for its efficiency in model training with higher number of classes and a significantly lower number of samples, offering a practical and resource-conscious alternative that contributes to the advancement of the field.

A pre-trained CNN is used as a transfer learning model to further train the Triplet Loss model using the Siamese network and gradients are updated. The pre-trained CNN is modified, and custom convolutional and dense layers are added to perform the model training. We ensured that datasets were preprocessed, images were cropped, faces extracted and aligned before the training. As part of preprocessing activity, we used an image size of 112x112 and the alpha channel from the synthetic dataset (DigiFace-1m) was removed to remove transparency and reduce the number of image channels to 3.

Post-model training, results were evaluated on the standard real and synthetic datasets such as LFW, CelebA, VGGFace2 and DigiFace-1m. Results are evaluated using ‘Model Testing’ and ‘Real time’ evaluation. The evaluation is performed using few shot samples from support set and query set. These results are presented in Section 4.

To the best of our understanding, the studies that use Siamese network using triplet loss do not provide a comprehensive analysis on different triplet mining strategies under few shot settings. These studies [1–6] either utilize triplet loss directly or provide a task specific version [7] of triplet loss. Contributions of the study are:

1. In-depth examination of triplet loss employing various triplet mining strategies for parametric few-shot learning.
2. Model training and evaluation strategy designed within the constraints of a limited dataset, particularly in the realm of few-shot learning.
3. The study conducts experiments on samples from classes for 1-shot, 2-shots and 5-shots that are not seen during training and validation process. This adds a distinctive dimension to the study.
4. Design of two-stage training approach that improves the effectiveness of the model and provides insights into handling limited data scenarios in face recognition. Provides insights into the factors influencing performance of CNN models for face recognition.
5. Experimental results obtained from these carefully designed methodologies will contribute to advancing the field and addressing the challenges associated with face recognition tasks.

The organization of the paper is as follows. In Section 1, we introduce the study of triplet loss-based face recognition and present the motivation behind our study. Section 2 provides a comprehensive review of literature and existing work related to usage of triplet loss and face recognition. Section 3 covers methodology adopted to accomplish different tasks related to the study. Section 4 discusses and summarizes performance evaluation results of the experiments conducted related to the study. Finally, Section 5 provides conclusions and future work.

2 Literature survey

Early stages of face recognition can be traced back to various research texts [8, 9]. At that time, much of the focus was on manually designing and crafting features for face recognition.

However, in recent years, there has been a shift towards making machines more intelligent [10]. Researchers are now aiming to offload the responsibility of solving the aforementioned complex task to automated systems.

Over time, these initial research efforts evolved, leading to advancements in face detection and facial feature extraction techniques. Appearance-based methods such as fisherfaces [11], as well as feature-based approaches, were proposed to handle larger datasets consisting of facial images. Multiple approaches based on Support Vector Machines (SVM) [12–14], Principal Component Analysis (PCA) [11], and Hidden Markov Model (HMM) [15, 16] were also introduced to tackle face recognition tasks. Machine learning based approaches have been used by using the subspace discriminant ensemble-based approach [17]. A hybrid approach to recognize faces is also used by using Viola Jones, PCA and applying PCA on detected features. Viola Jones is still being used to detect a face and PCA is used along with it to detect different parts of the face such as face, left eye, right, nose and mouth [18]. Features are extracted from the detected parts of these faces and the face is recognized by applying PCA. These techniques served as fundamental building blocks for subsequent research conducted in controlled environments and with limited datasets.

Advancement of technologies in the current era is enabling identity authentication and authorization using face verification. This has enabled a face recognition system to be a generalized source of authentication. To achieve this, it is important that data is normalized, segmented and good quality features [19] are generated. Structural and facial similarity between different faces adds to the complexity of the face recognition task and research is being done to extract texture features [20] from eyes, nose, mouth and face. Face recognition generalizes well in control situations such as similarity matching only using the frontal view. However, there are scenarios where there is a pose instead of a frontal view. In that case the frontal view is calculated from the pose-view [21] angle before performing the face similarity. Current age of deep learning has enabled Face Recognition [22] tasks to be progressed at a level of fair maturity. This has been made possible by high computing systems, availability of datasets and evolution of new techniques, technology and algorithms. One such algorithmic technique uses triplet loss [1, 23] and different triplet mining [24] strategies to find the similarity [25] between faces. This is a novel technique that enables CNN to produce face embeddings that represent similar embeddings closer in the latent space and the different face embeddings have a larger distance between them. The technique is primarily used in Siamese [26, 27] network-based modelling. This approach has been used to conduct similar experiments to perform face recognition [28] where the face images are occluded with a mask [29–33]. The triplet-based model using a Siamese network has been used in unsupervised learning to generate more accurate pseudo labels [34] for person re-identification tasks [35]. It has been observed that for partial matching and to counter occlusion, an evolved version of the triplet loss function [29] can be used to further improve the performance [36, 37] of the model on the standard datasets. The triplet loss-based approach has been extended to relatively new concepts of Few-shot learning [30] whereby only a limited set of datasets is required to achieve significantly good performance on the given face recognition task. This significantly reduces the requirement of having a high volume of training datasets. The use of triplet loss in our experiments has proven to be effective in reducing the requirements for a large number of samples [38] per class. This approach mimics the behaviour of few-shot learning methods, as highlighted in a recent study by Holkar et al. [26]

The conventional methodologies in the field often rely on extensive datasets and employ deep neural networks to achieve broad generalization. However, this conventional approach faces a notable limitation due to its dependency on large datasets. It is this limitation that serves as a primary motivation for our study, prompting us to explore and conduct experiments in scenarios characterized by limited data availability. In contrast to the common practice of employing numerous samples per class, our study specifically investigates the efficacy of utilizing significantly fewer samples per class. The objective is to demonstrate that even with restricted data, it is possible to achieve substantial generalization.

In addition to addressing the challenges associated with limited data scenarios, our study also leverages a few-shot learning strategy. This innovative approach aims to minimize the reliance on extensive computational resources traditionally required for model training. By adopting a few-shot learning strategy, we effectively reduce the computational burden, leading to a notable reduction in the overall model training time. This not only contributes to resource efficiency but also underscores the practicality and applicability of the proposed methodology in scenarios where computational resources are constrained.

3 Methodology

The system pipeline consists of multiple stages. Each stage corresponds to a specific task. In stage-1, the base network is trained and in stage-2 triplet loss network is trained by using the base network from stage-1. The stage-1 task is intended to develop a model that is used for features extraction. To achieve this, the widely adopted dataset DigiFace-1m is chosen. The data is augmented offline, and the model is trained thereafter. VGG16 is used as a base network for transfer learning. As shown in the computation graph in Fig. 2, the weights corresponding to block-5 of VGG16 and the fully connected layers are updated during the training process. Further details related to base network selection, dataset selection criterion and model training are presented in the subsequent sections respectively.

3.1 Base network training methodology (Stage -1)

3.1.1 Base network selection criterion

The pre-trained VGG16 model is selected based on the following considerations:

1. **Simple architecture** - The architecture of VGG16 is straightforward and easy to understand, consisting of stacked convolutional layers and dense layers.
2. **Pre-trained** - VGG16 has been pre-trained on the large-scale image dataset ‘ImageNet’. This enables the network to learn generic features from large-scale datasets.
3. **Performance on medium-sized dataset** - VGG16 performs well on medium dataset. It does not perform well like other recent models like Resnet50, Inception, etc. However, our objective of training on a medium sized dataset is fulfilled by VGG16.
4. **Shorter** training time - Because of shorter training time than other standard networks, VGG16 is the ideal selection for the experiments.

System pipeline and methodology adopted for training and validating the base network is displayed in Fig. 1

3.1.2 Dataset selection

For selecting the datasets for model training and testing, evaluation is done on state-of-art datasets. Criterion for dataset selection is mentioned in Table 1. The criterion is:

1. **Active** - The dataset must be active for current research.

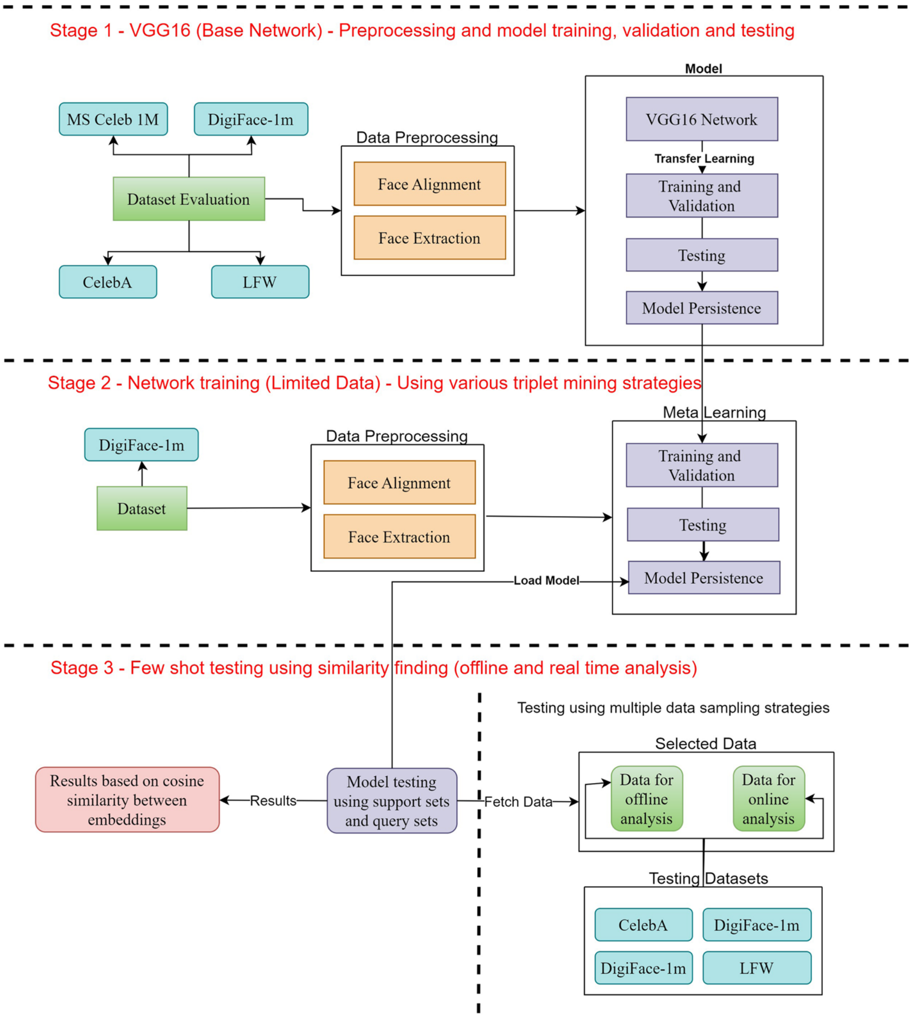
Stage 1 - VGG16 (Base Network) - Preprocessing and model training, validation and testing
MS Celeb 1M
DigiFace-1m
Dataset Evaluation
CelebA
LFW
Data Preprocessing
Face Alignment
Face Extraction
Model
VGG16 Network
Transfer Learning
Training and Validation
Testing
Model Persistence
Stage 2 - Network training (Limited Data) - Using various triplet mining strategies
DigiFace-1m
Dataset
Data Preprocessing
Face Alignment
Face Extraction
Meta Learning
Training and Validation
Testing
Model Persistence
Load Model
Stage 3 - Few shot testing using similarity finding (offline and real time analysis)
Testing using multiple data sampling strategies
Results based on cosine similarity between embeddings
Results
Model testing using support sets and query sets
Fetch Data
Selected Data
Data for offline analysis
Data for online analysis
Testing Datasets
CelebA
DigiFace-1m
DigiFace-1m
LFW

Fig. 1 Overview of system pipeline and methodology for training and testing

2. **Class Sufficiency** - The dataset must have enough classes having enough samples on which training can be done.
3. **Sample sufficiency per class** - One common method to increase the size of the dataset is by augmenting the data. Embeddings of an augmented image closely resemble those of the original image. When working with limited data, it is preferable to use a dataset that naturally incorporates variations in the samples. This allows the network to generate more accurate results without relying heavily on data augmentation. Based on these considerations, we opted for a dataset that contains a minimum of 50 samples per class (without augmentation) for training the base network.
4. **Balanced** - To avoid biases and fair distribution, we ensured that each class must be represented equally.

Table 1 Dataset selection criterion

Dataset	Criterion			
	Active	Class sufficiency	Samples sufficiency per class	Balanced
DigiFace-1m	Yes	Yes	Yes	Yes
MSCeleb-1m	No	Yes	Yes	Yes
CelebA	Yes	Yes	No	Yes
LFW	Yes	Yes	Yes	No

Based on the criterion mentioned in Table 1, DigiFace-1m dataset is selected for model training and testing. Details of the DigiFace-1m dataset are presented in Table 2

3.1.3 Dataset preprocessing - base network

Because of the inherent complexity of the facial features, the model is trained on the frontal view of the face. However, some images are pose-variant and do not present a frontal view. As part of the data preprocessing, the dataset is iterated and face alignment, extraction and resizing are performed on every sample. Sample(s) on which automatic alignment or extraction could not be performed is discarded from model training and testing. Face alignment is performed using the OpenCV library. MTCNN is used to perform face extraction and the image is resized to 112x112 pixel with a reduction of alpha channel.

3.1.4 Model training, validation and testing

VGG16 is used for face recognition tasks by transfer learning and only training its last convolution layer and subsequent dense layers. Input to the model is an RGB image having width and height as 112 pixel and number of channels as 3. DigiFace-1m dataset has an alpha channel that was removed as part of the preprocessing. The model computational graph is presented in Fig. 2. The optimized model training configuration and parameters are presented in Table 3

Our objective was to train the model minimally on Face Recognition tasks and achieve sufficient accuracy of around 80%, so that model could minimally detect the face and be used for extracting the embeddings by the downstream network.

3.2 Triplet loss network training methodology (stage -2)

Triplet loss network uses the feature extraction network trained in stage-1 of network pipeline. The softmax layer of the feature extraction network is removed, and the features are extracted using the last dense layer of 128 dimensions. This model is fine-tuned by attaching a convolutional layer and the couple of fully connected layers that are normalized and the triplet loss

Table 2 Selected dataset for base network (Stage -1)

Dataset	Type	Total #Classes	Selected #Classes	Selected classes name	Selected #Samples per class
DigiFace-1m	Synthetic	10k	200	0 – 199	72

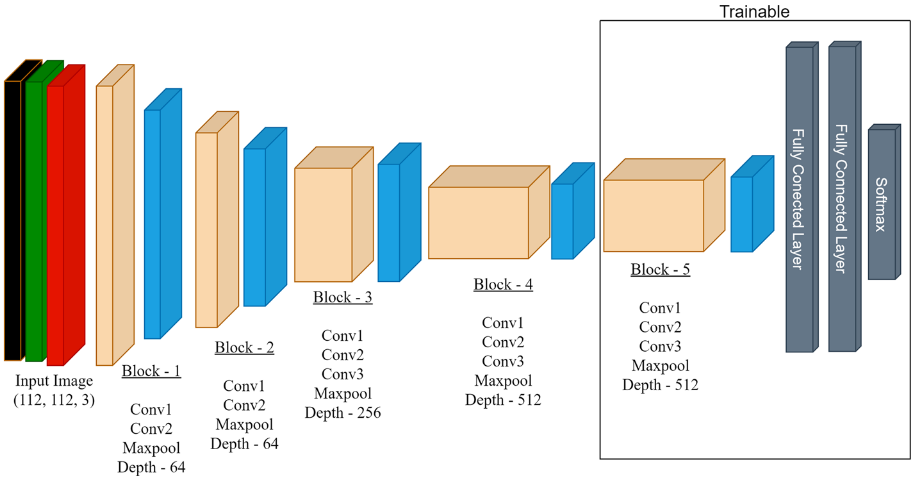

Fig. 2 VGG16 computation graph - face recognition task

function is applied thereafter. As part of the training process, the size of mini batch plays a critical role in determining the optimal triplet for that particular batch. Hence, it is suggested to have maximum representations of different classes in the mini batch. During the training process, the triplets are mined, and training is performed. The loss is updated based on the distance between the anchor, positive and negative samples mined for each class in the mini batch. The methodology adopted for training the triplet loss network is displayed in Fig. 1

3.2.1 Training and validation dataset selection

All the experiments have been conducted on standard datasets namely DigiFace-1m, CelebA and LFW. DigiFace-1m is a large dataset with 10k identities and 72 image samples per class. LFW and CelebA datasets have multiple identities but a limited number of samples per class. The selection of the dataset for training the model is based on the criterion mentioned in Table 1. The base model is trained on the DigiFace-1m dataset, specifically on 200 classes as indicated in Table 2. For training the triplet loss network, we also utilize the DigiFace-1m dataset, but with different classes and samples. By selecting different classes and samples, we ensure that the training data for the triplet loss network is distinct from the data used to train the base network. Further details about the dataset are mentioned in Table 4. The "Selected Classes" column in the table indicates that the class names range from 2000 to 2049.

3.2.2 Dataset preprocessing - triplet loss network

This section outlines the preprocessing techniques applied to the selected datasets, including image resizing, face extraction, face alignment, and normalization. Precursor for face

Table 3 Optimized model training parameters for base network

Base model	Learning rate	Optimizer	Batch size	Total #Classes	Selected #Samples per class
VGG16	0.001	SGD	32	200	72

Table 4 Selected dataset - triplet loss network (stage 2)

Dataset	Type	#Classes in dataset	Selected #Classes	Selected classes	Selected #Samples per class
DigiFace-1m	Synthetic	10k	50	2000 to 2049	63

recognition is face detection task. To enable a CNN model to perform well on face detection tasks, face extraction is done using MTCNN and face alignment using OpenCV as part of the preprocessing. The images in DigiFace-1m dataset are of size 112x112 and 4 channels. After preprocessing, we resized the images to 112x112 and 3 channels and removed the alpha channel.

3.2.3 Hyperparameters selection

Choosing appropriate hyperparameters significantly influences the training results of CNN models for face recognition. This section discusses the process of selecting hyperparameters, such as learning rate, batch size, and margin values, and the considerations involved in their determination. Rationale behind the chosen hyperparameters is provided, ensuring a robust training process. For achieving good results from the model training and faster convergence, we emphasize the need of selecting a batch size that must have a sufficient representation of samples from each class for a decent triplet mining strategy. We chose to have a batch size of 1024 so as to have sufficient representation of each class in a mini batch.

3.2.4 Triplets mining strategies and network training

This section presents an in-depth exploration of various triplet mining strategies employed in face recognition tasks. Different techniques, such as Hard Triplets, Semi-Hard Triplets, and offline triplets, are examined for advantages and challenges. We provide insights into selecting suitable triplets to train the CNN models effectively. Triplets are formed by selecting an anchor (A) and a positive (P) sample from the same class and a negative (N) sample from any other class. There are primarily two ways of selecting a triplet, i.e., Hard triplets and semi hard triplets [1]. Triplets are selected such that the network learns that the Euclidean distance between the anchor and positive pair is less than the anchor and negative pair by a margin. This process of network training [1] is presented in Fig. 4

Schematic of our network design is represented in Fig. 3. Different sizes of shapes in the mini batch specify unequal distribution of samples per class. Deep learning architecture block specifies that transfer learning is performed on VGG16, and some custom layers have been added as well. These layers include a 1x1 filter-based convolution layer and a couple of fully connected layers. The output is L2 normalized to provide the normalized 128-dim embeddings.

3.2.5 Triplet loss objective

The objective of the triplet loss function (f) represented in (1) is to establish that the embeddings of anchor and positive samples are represented closely in the latent space. And embeddings of anchor and negative (N) samples must be at distance greater than the anchor and positive sample distance. Thus, the objective is to minimize the distance between the

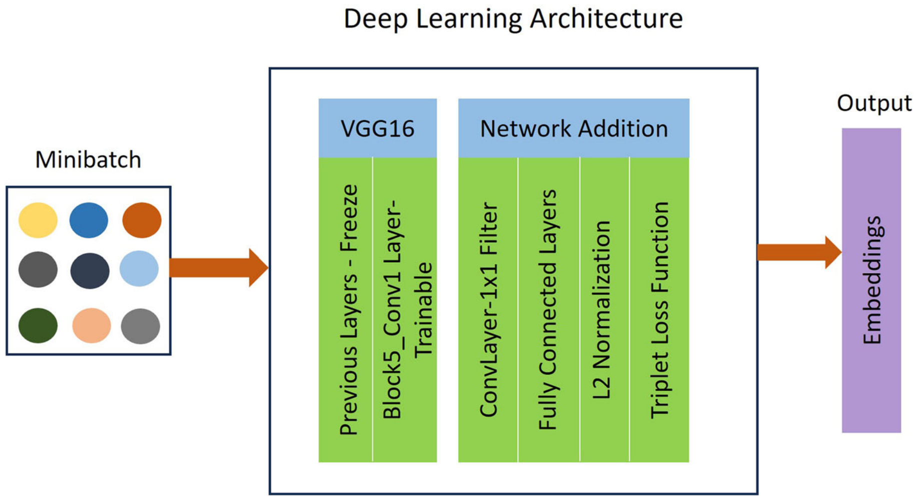

Fig. 3 Triplet loss - network training

anchor and the positive and maximize the distance between the anchor and negative. The triplet loss function is represented as (1). The function *f(x)*$_{\mathrm{i}}$ produces the embeddings of a sample 'x' that belongs to i^{th} class.

$$L_{\mathrm{A,P,N}} = \max(||f(A) - f(P)||_2 - ||f(A) - f(N)||_2 + \alpha, 0) \tag{1}$$

For a mini batch of size 'M', the loss per batch is represented as (2):

$$L_{\mathrm{A,P,N}} = \frac{\sum \max(||f(A) - f(P)||_2 - ||f(A) - f(N)||_2 + \alpha, 0)}{M} \tag{2}$$

3.2.6 Hard triplets

A negative image sample (N) is selected such that the Euclidean distance between anchor (A) and negative is less than the Euclidean distance between anchor and positive (P) samples embeddings. Equation (3) represents distance (d) of hard triplets.

$$d||A, N|| < d||A, P|| \tag{3}$$

3.2.7 Semi-hard triplets

A negative sample image is selected such that the Euclidean distance between anchor image and negative image embedding is less than the Euclidean distance between anchor and positive image embedding by a margin. Equation (4) represents the distance (d) in case of semi-hard triplets by a margin of α.

$$d||A, P|| < d||A, N|| < d||A, P|| + \alpha \tag{4}$$

3.2.8 Determining the number of triplets

With reference to (3) and (4), it is sufficient to select one best triplet per class per mini batch. This can be achieved by selecting the best anchor positive and anchor negative pairs. We created all possible anchor positive combinations [1] of the triplets for a mini batch. Thus, if a class has ‘n’ samples then there are $\binom{n}{2}$ possible triplets for that class. This ensures to have sufficient representations of samples per class in a mini batch.

3.2.9 Approaches adopted for negative sample selection

The selection of negative samples is a crucial factor that significantly impacts the convergence of triplet loss. In our study, we conducted experiments using two primary strategies for selecting the negative sample either by selecting a random negative sample or by selecting the best negative sample. The approach to select a random negative and best negative sample is presented below.

- **Selecting a random negative**

  1. There are ‘c’ classes and each class has ‘s’ number of samples.
  2. ‘S’ denotes the total number of samples from (c-1) classes.
  3. Then, a negative sample is selected at random from the set of ‘S’ for each anchor and positive pair.

  Algorithm 1 is used for selecting a random negative sample from a mini batch for generating a triplet.

$$\text{Let } S = \{s1, s2, s3, s4, s5...(c - 1 * s)\}$$

$$n \in S \text{ where ‘n’ represents random negative sample}$$

Algorithm 1 Find a random negative embedding.

```
Input: Euclidean distance between anchor (A) embedding and negative (N) embedding. Anchor Embedding (anchorembedding) and the class labels (classes)
Output: Random negative embedding as (negativeembedding)
initialize $negativeembedding \Leftarrow -1$
intialize $foundnegative \leftarrow false$
for $i = 1$ to $classes$ do
    $clazz \leftarrow randomize(classes)$
    for $i = 1$ to $clazz$ do
        $sample \leftarrow clazz$
        $andist \leftarrow euclidean\_distance(anchorembedding, sample)$
        if $andist < anDist$ then
            $negativeembedding \leftarrow sample$
            $foundnegative \leftarrow true$
            break
        end if
        if $foundnegative = true$ then
            break
        end if
    end for
end for
```

- **Selecting Best Negative**

  1. This approach requires determination of a negative sample by iterating through all the classes in a mini batch and finding the negative sample image that is closest to the anchor image as presented in (5).

$$argmin_n ||A - N|| \tag{5}$$

Algorithm 2 is used for selecting the best negative sample from a mini batch for generating a triplet.

Algorithm 2 Find $best\_negative\_embedding(anDist, anchorembedding, classes)$.

```
Input: Distance between anchor and negative embedding, anchor embedding, labels
Output: Best negative embedding
initialize $negativeembedding \Leftarrow -1$
initialize $minDist \Leftarrow -1$
for $i = 1$ to $classes$ do
    $clazz \leftarrow i$
    for $i = 1$ to len($clazz$) do
        $sample \leftarrow clazz$
        $andist \leftarrow euclidean\_distance(anchorembedding, sample)$
        if $minDist < andistandandist < anDist$ then
            $minDist \leftarrow anDist$ State $negativeembedding \leftarrow sample$
        end if
    end for
end for
```

3.2.10 Testing strategies

This section describes testing strategies used while evaluating the performance of the triplet loss model. The testing strategies are divided into two categories ‘model testing’ and ‘real time testing’. These categories are decided based on the number of samples per class, that are to be compared using cosine similarity for evaluating model performance. Data sampling and splitting strategies are presented in Table 5.

3.2.11 Similarity score estimation

As shown in the network design in Fig. 4, the output of the network is the extracted features or embeddings. Cosine similarity is used as the distance measure for testing and evaluation

Table 5 Testing strategies - data sampling

Strategy	Data category	Data sample	Description
Model Testing	Training-Test Split	Training set (63 samples per class) Testing set (7 samples per class)	Training data is seen by model during training and testing data is not seen during the model training
Model Testing	Support-Query Split	Support set (63 samples per class) Query set (7 samples per class)	Support set and Query Set are not seen during the model training
Real Time	Support-Query Split	Support set (2 samples per class) Testing set (1 sample per class)	Support set and Query Set are not seen during the model training

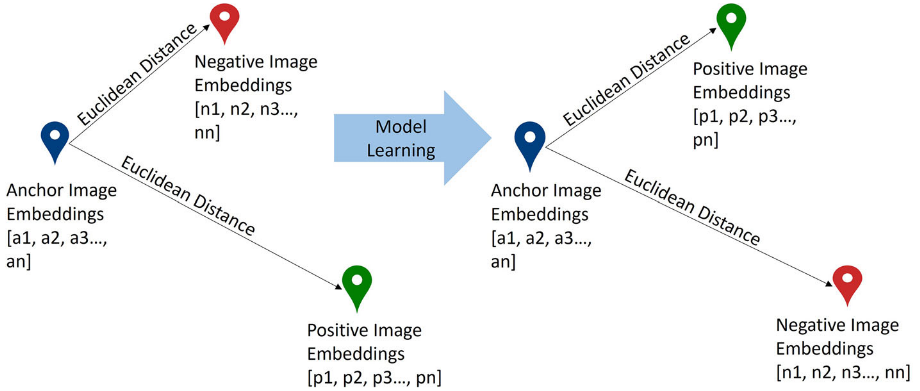
Euclidean Distance
Negative Image Embeddings [n1, n2, n3..., nn]
Anchor Image Embeddings [a1, a2, a3..., an]
Euclidean Distance
Positive Image Embeddings [p1, p2, p3..., pn]
Model Learning
Euclidean Distance
Positive Image Embeddings [p1, p2, p3..., pn]
Anchor Image Embeddings [a1, a2, a3..., an]
Euclidean Distance
Negative Image Embeddings [n1, n2, n3..., nn]

Fig. 4 Network transformation - triplet learning

of our experiments on face recognition datasets on completely unseen dataset.

$$\text{cosine similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \cdot \|\mathbf{B}\|} \tag{6}$$

In (6) A and B represents "training and testing" or "support and query" samples embeddings.

3.2.12 Hardware specifications

This section captures hardware resources used during the model training along with training times of different models. The hardware resources details are mentioned in Table 6.

4 Discussions and results

This section comprehensively lists and describes the performance evaluation results of the base network and the triplet loss network. Results of triplet loss-based CNN are governed by underlying pre-trained model that is used as a base network. It also represents the compilation of results for experiments that are conducted with different triplet mining strategies. Influence of hyperparameters on the results is described in Section 3.2.3. Triplet loss network performance evaluation results are also presented on real time strategy with limited samples on unseen data

Table 6 Hardware specifications

GPU specifications	CPU specifications
Tesla V100-PCIE	Intel (R) Xeon (R) Gold 5218 CPU @ 2.30 GHZ
Number of GPU Cards -2	CPU Cores - 64
Memory 32 GB Per card	Memory 1TB

4.1 Model training hours

The study specifies 'Base Network' that acts as the embedding network as one CNN model and the 'Triplet Loss' model with different triplet mining strategies as another CNN model. Both models trained on same hardware specified in Table 7 reflect the training time taken by 'Base Network'and 'Triplet Loss' CNN models. Higher training times for triplet loss networks are attributed to the non-vectorized implementation of the triplet selection algorithms.

4.2 Results

This section provides quantitative results related to base network and triplet loss network from Section 3

4.2.1 Base network

The model is trained on the DigiFace-1m dataset as specified in Table 2. The dataset percentage split used for training, validation and testing is 80-10-10%. The model is trained using transfer learning from VGG16 pre-trained model on the task of face recognition. As shown in Fig. 2, transfer learning was applied from the last convolution block of VGG16 network by unfreezing its last layer and adding custom fully connected layers. The input is classified using softmax. The model is run for 500 epochs. Training and validation graphs clearly suggest that convergence is achieved around the 90-100th epoch. 200 classes and 72 samples for each class are chosen for model training. A standard split of 80-10-10 is used for training, validation, and testing respectively. The performance evaluation results are mentioned in Table 8. The intermediate results are listed in the form of confusion matrix for the intermediate model that is used as a feature extractor. Since the model is trained on 200 classes and corresponding confusion matrix will be challenging to represent, we have presented the complete confusion matrix in Fig A1 of appendix. Here, we are presenting confusion matrix Fig. 5 for randomly selected 10 classes. Accuracy and Loss graphs are presented in Figs. 6 and 7 respectively.

4.2.2 Triplet loss network - quantitative results

The triplets' losses for different triplet mining strategies are shown in Fig. 8. It is clear that the selection of the best negative sample in both hard triplets and semi-hard triplets mining strategy converges faster than the randomly selected negative sample approach.

We report the quantitative evaluation results in Tables 11 and 10 for different triplet mining strategies presented in Section 3. The most optimized hyperparameters are presented in Table 9.

Table 7 Model training hours

Model	Training time
Base network	1 hour, 40 minutes
Semi-hard triplets and random negative	35 hours
Semi-hard triplets and the best Negative	55 hours
Hard triplets and random Negative	33 hours
Hard triplets and the best negative	56 hours

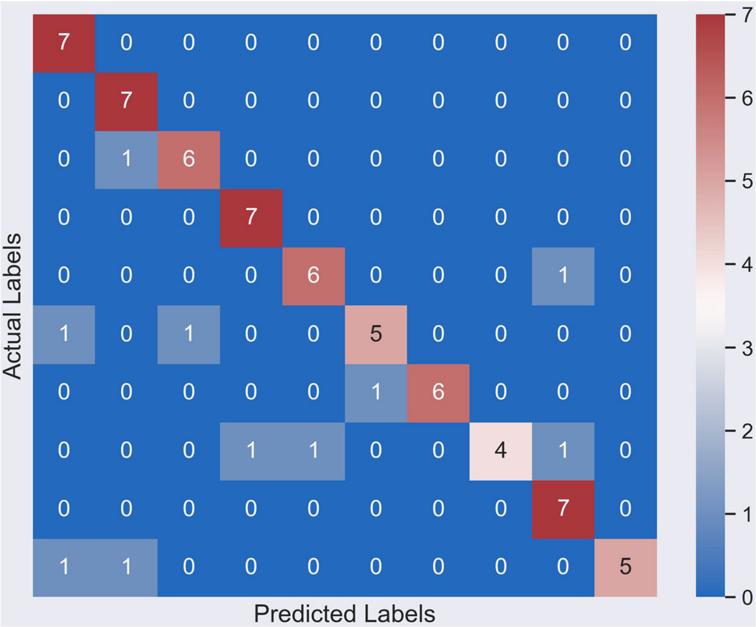

Fig. 5 Confusion matrix for base network for randomly selected 10 classes

As per experimental results mentioned in Table 11 using ‘model testing’ based performance evaluation strategy, it is evident that semi-hard triplet selection with the randomly selected negative sample yields best results on both seen and unseen data. Selection of a base network that provides optimal embeddings of an image plays a significant role in triplet loss base model. In case of ‘real time’ testing strategy, a new dataset VGGFace2 is utilized to evaluate performance of model using limited datasets. In this performance evaluation strategy, classes in query set and support set are same, but samples are different. Only 2 samples per

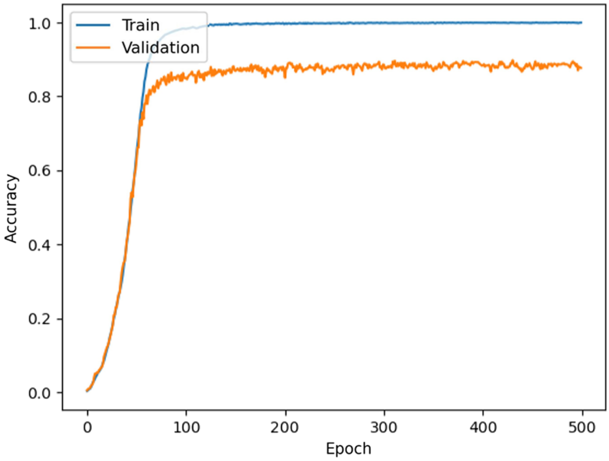

Fig. 6 Base network - training and validation accuracy

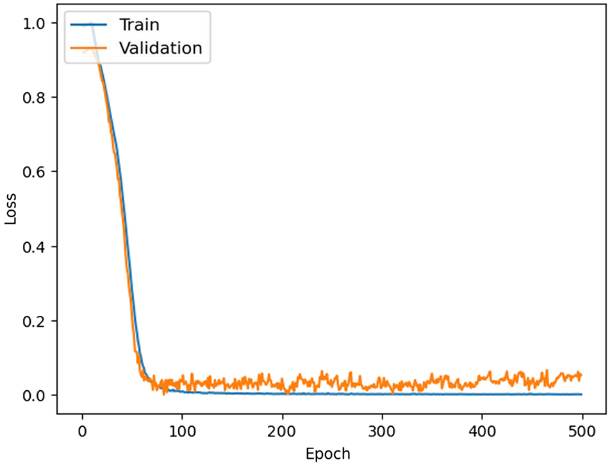

Fig. 7 Base network - training and validation loss

Table 8 Base network - performance evaluation results

Training loss	Training Acc	Validation loss	Validation Acc	Testing loss	Testing Acc
0.00364	99.67%	0.00323	89.74%	0.00431	88.76%

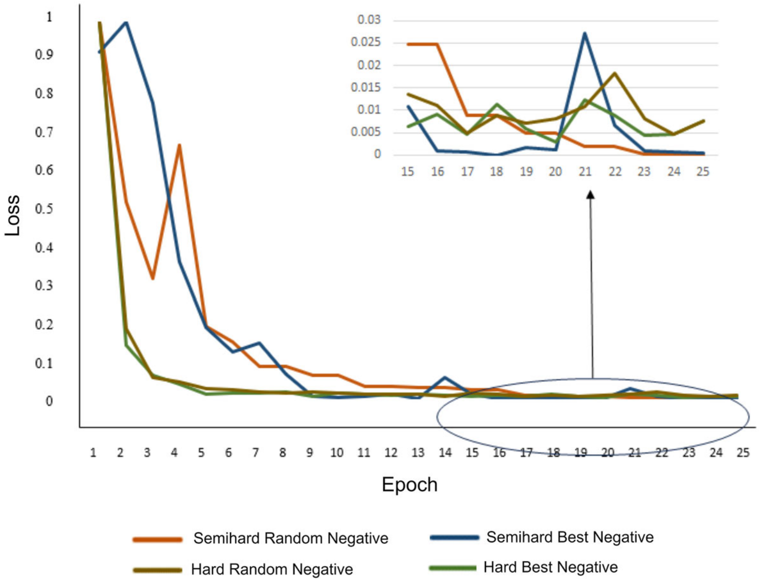

Fig. 8 Triplet losses for different triplet mining strategies

Table 9 Optimized hyperparameters for triplet loss network

Batch size	Optimizer	Learning rate	Margin
1024	SGD	0.001	0.2

class are selected in support set for different testing executions. Query set has only one image per class. Performance evaluation results are presented in Table 10. Similarity matches in ‘real time’ test strategy between samples of different classes are presented in Fig. 9. This suggests that triplet loss-based training is particularly useful in constrained environments where number of samples is limited or few. Selection of the ‘margin’ variable and the ‘batch size’ play a significant role while model training. An optimally chosen batch size must be such that significant samples per class are present in each mini batch. Performance results for experiments related to offline triplet mining are inconclusive and are mentioned as ‘inconclusive’ in Table 11.

4.2.3 Comparative analysis

This section describes the comparative analysis between different studies. Table 12 displays the comparisons between different techniques that have primarily used triplet loss and few shot learnings in their experiments. Our experiments are performed in multiple few shot configurations, provide detailed analysis of triplet mining techniques, and tested on unseen classes.

The SOTA models like FaceNet [1] and DeepFace [40] are designed to learn from various variations, such as changes in illumination and pose, to produce high-quality embeddings. Achieving this involves leveraging deep network architectures and training on extensive datasets comprising thousands of identities and millions of samples. It is crucial to note that our study does not seek to draw comparisons with these state-of-the-art models, which often employ proprietary datasets and intricate network architectures. Instead, our investigation is centered around a more constrained dataset and a less complex network architecture. In contrast to many existing studies that utilize a few-shot learning methodology, our approach differs in terms of both dataset size, mining and network depth. Unlike studies that often involve training and testing for classes either ≤20 or ≥1000 and high number of samples per class, our experiments cover 50 classes with very few numbers of samples. Moreover, our study introduces variations in the number of samples per class (one, two, or five), and these classes remain unseen during the training and validation phases. While many studies commonly employ Siamese Networks [21, 26] with contrastive loss and some studies use quadruped loss [39], our research delves into the intricacies of different triplet mining strategies. Our particular focus is on parametric few-shot learning, with an emphasis on testing for classes that were not part of the training or validation process. This distinct approach allows for a more comprehensive examination of the model’s generalization capabilities on unseen classes, given the limitations of our dataset.

Table 10 Performance evaluation results

Dataset	#Classes	Support Set-#Samples per class	Query Set-#Sample per class	Test Acc%	Response time
VGGFace2	50	2	1	70%	100 ms
VGGFace2	100	2	1	68%	110 ms
VGGFace2	200	2	1	62%	100 ms

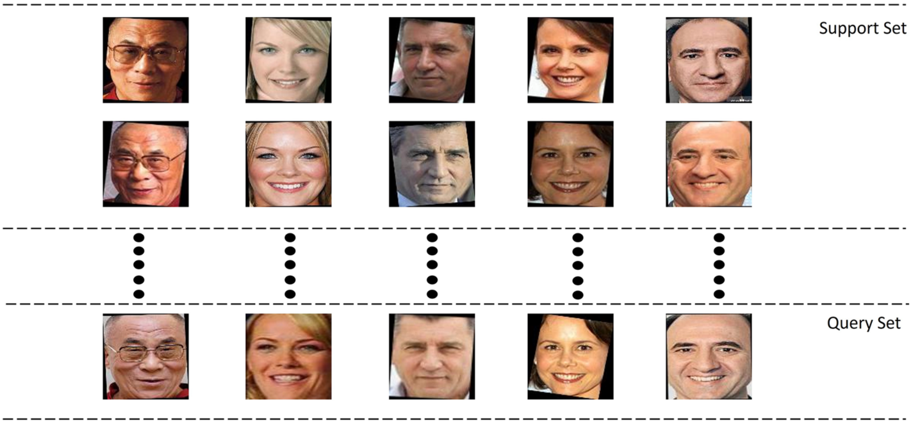

Fig. 9 Similarity - samples from support and query sets

5 Conclusions and future work

In this paper, we have presented a comprehensive analysis of methodologies for face recognition using few shots via the metric based learning. The performance is evaluated on unseen dataset and we observed an accuracy of over 70% in both real time and 'model testing' mode for each of the 50 unseen classes with a processing time of about 100 milliseconds. . The study compares performance of various triplet selection techniques and demonstrates effectiveness

Table 11 Triplet loss network -performance evaluation results (model testing)

Method	Results - evaluation test strategy			
	Trained dataset	Unseen dataset	Test accuracy-train-test split dataset	Test accuracy-support query split dataset
Semi-hard triplets and random negative	DigiFace-1m	CelebA	85.6%	69.23%
	DigiFace-1m	DigiFace-1m	85.6%	72.23%
	DigiFace-1m	LFW	85.6%	70.23%
Semi-hard Triplets and Best Negative	DigiFace-1m	CelebA	81.23%	67.37%
	DigiFace-1m	DigiFace-1m	81.23%	70%
	DigiFace-1m	LFW	85.6%	68.76%
Hard triplets and random negative	DigiFace-1m	CelebA	79.37%	66.57%
	DigiFace-1m	DigiFace-1m	79.37%	68.35%
	DigiFace-1m	LFW	79.37%	67.35%
Hard triplets and best negative	DigiFace-1m	CelebA	75.32%	62.37%
	DigiFace-1m	DigiFace-1m	75.32%	67.37%
	DigiFace-1m	LFW	75.32%	66.23%
Offline triplets	DigiFace-1m	CelebA	Inconclusive	Inconclusive
	DigiFace-1m	DigiFace-1m	Inconclusive	Inconclusive
	DigiFace-1m	LFW	Inconclusive	Inconclusive

Table 12 Comparative analysis of our study with SOTA models

Learning algorithms, type of network, loss function	Does the study use few shots?	Network Architecture used in the study	Does the study provide comprehensive comparison and results using different Triplet Loss Techniques	#Training classes used for model training	#Samples used for model training	Is the model tested on unseen classes?
Few shot learning using contrastive loss [26]	Yes	Resnet50	No	11	$\geq$ 1000s	No
Triplet loss and quadruplet loss [39]	No	Not Available	No	$\geq$ thousands	$\geq$ millions	No
Contrastive loss without few shots [21, 40]	No	Not Available	No	10575	494414	No
Facenet using triplet loss [40]	No	Modified Inception	Yes	$\geq$ thousands	$\geq$ millions	Yes
Triplet loss wihtout few shots [20]	No	Resnet50	No	85000	58 million	No
Our study	Yes	VGG16	Yes	50	3500	Yes

of triplet Loss in training a CNN for face recognition tasks. Our results show that two-stage training approach, incorporating a pre-trained VGG16 as base feature extraction network, yields promising results in limited dataset scenario using few shot learning. The study also highlights the impact of hyperparameters and data sampling on performance.

While this study provides valuable insights into face recognition with limited data using few shots, there are several directions for future research. Different few shots learning techniques can be used with cross domain datasets to further improve the training methodology and training times. Overall, this study lays the foundation for further advancements in face recognition with limited data, and future research can build upon these findings to address the challenges and explore new possibilities in this field.

Appendix A: Extended data

The Figure displays confusion matrix for intermediate results from Stage-1 training and testing. The confusion matrix for 200 classes is hard to interpret and hence provided in this section.

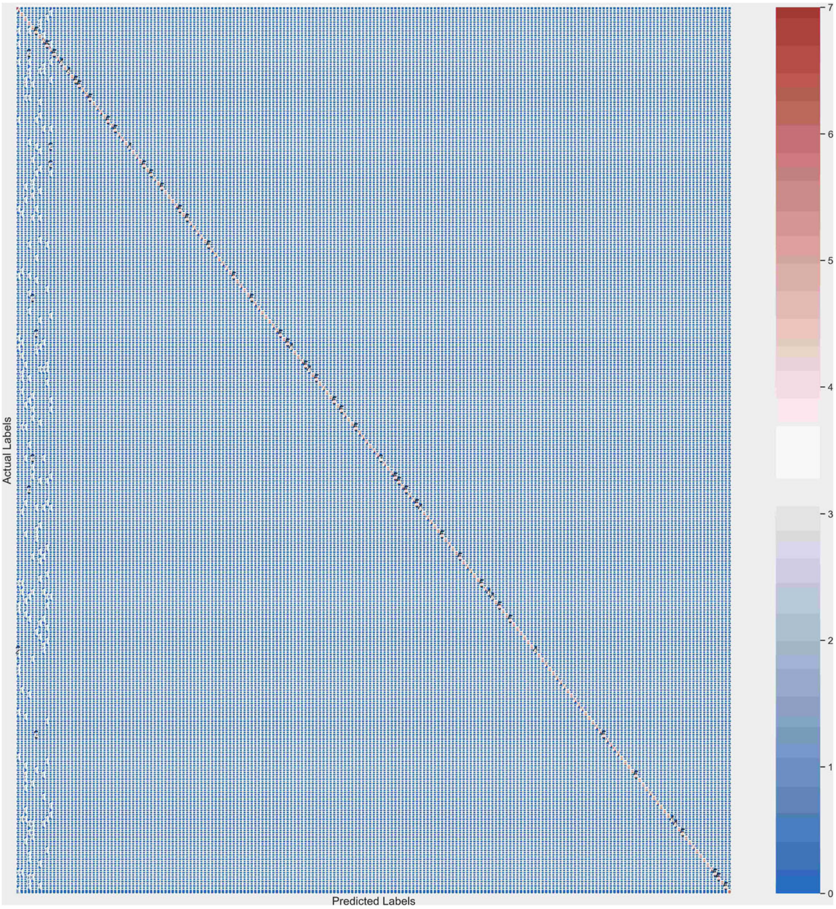

Fig. 10 Confusion matrix for base network with 200 classes

Funding Authors have received no funding for this research.

Data Availability The datasets analyzed during the current study are publicly available and cited appropriately.

Declarations

Conflicts of interest Authors declare that they have no conflict of interest.

References

1. Schroff, F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. arXiv:1503.03832
2. Yu J, Hu C-H, Jing X-Y, Feng Y-J (2020) Deep metric learning with dynamic margin hard sampling loss for face verification. Signal, Image and Video Processing 14(4):791–798. https://doi.org/10.1007/s11760-019-01612-3. Cited by: 6
3. Harvill J, Leem S-G, Abdelwahab M, Lotfian R, Busso C (2023) Quantifying emotional similarity in speech. IEEE Trans Affect Comput 14(2):1376–1390. https://doi.org/10.1109/TAFFC.2021.3127390. Cited by: 2; All Open Access, Hybrid Gold Open Access
4. Abdallah MS Kim H, Ragab ME, Hemayed EE (2019) Zero-shot deep learning for media mining: person spotting and face clustering in video big data. Electronics (Switzerland) 8(12). https://doi.org/10.3390/electronics8121394. Cited by: 7; All Open Access, Gold Open Access
5. Uzhinskiy AV, Ososkov GA, Goncharov PV, Nechaevskiy AV, Smetanin AA (2021) One-shot learning with triplet loss for vegetation classification tasks. Comput Opt 45(4):608–614. https://doi.org/10.18287/2412-6179-CO-856.. Cited by: 6
6. He M, Zhang J, Shan S, Kan M, Chen X (2020) Deformable face net for pose invariant face recognition. Pattern Recognit 100. https://doi.org/10.1016/j.patcog.2019.107113. Cited by: 42
7. Chen X, Lan X, Liang G, Liu J, Zheng N (2017) Pose-and-illumination-invariant face representation via a triplet-loss trained deep reconstruction model. Multimed Tools Appl 76(21):22043–22058. https://doi.org/10.1007/s11042-017-4782-y. Cited by: 13
8. Guo K, Wu S, Xu Y (2017) Face recognition using both visible light image and near-infrared image and a deep network. CAAI Trans Intell Technol 2(1):39–47. https://doi.org/10.1016/j.trit.2017.03.001
9. Tikoo S, Malik N (2017) Detection of face using viola jones and recognition using back propagation neural network
10. Zhao W, Chellappa R, Phillips PJ, Rosenfeld A (2003) Face recognition: a literature survey. ACM Comput Surv 35(4):399–458. https://doi.org/10.1145/954339.954342
11. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720. https://doi.org/10.1109/34.598228
12. Guo G, Li SZ, Chan K (2000) Face recognition by support vector machines. In: Proceedings fourth IEEE international conference on automatic face and gesture recognition (Cat. No. PR00580), pp 196–201. https://doi.org/10.1109/AFGR.2000.840634
13. Dubey RK, Choubey DK (2023) Deconstructive human face recognition using deep neural network. Multimed Tools Appl 82(22):34147–34162. https://doi.org/10.1007/s11042-023-15107-4
14. Mughaid A, Obeidat I, AlZu'bi S, Elsoud EA, Alnajjar A, Alsoud AR, Abualigah L (2023) A novel machine learning and face recognition technique for fake accounts detection system on cyber social networks. Multimed Tools Appl 82(17):26353–26378. https://doi.org/10.1007/s11042-023-14347-8
15. Samaria F, Young S (1994) Hmm-based architecture for face identification. Image Vis Comput 12(8):537–543. https://doi.org/10.1016/0262-8856(94)90007-8
16. Nefian AV, Hayes MH (1998) Face detection and recognition using hidden markov models. In: Proceedings 1998 international conference on image processing. ICIP98 (Cat. No.98CB36269), vol 1, pp 141–1451. https://doi.org/10.1109/ICIP.1998.723445
17. Juneja K, Rana C (2021) An extensive study on traditional-to-recent transformation on face recognition system. Wirel Pers Commun 118(4):3075–3128. https://doi.org/10.1007/s11277-021-08170-3
18. Taskiran M, Kahraman N, Erdem CE (2020) Face recognition: past, present and future (a review). Digital Signal Processing 106:102809. https://doi.org/10.1016/j.dsp.2020.102809

19. Xie W, Wu H, Tian Y, Bai M, Shen L (2022) Triplet loss with multistage outlier suppression and class-pair margins for facial expression recognition. IEEE Trans Circuits Syst Video Technol 32(2):690–703. https://doi.org/10.1109/TCSVT.2021.3063052
20. Tan Z, Liu A, Wan J, Liu H, Lei Z, Guo G, Li SZ (2022) Cross-batch hard example mining with pseudo large batch for id vs. spot face recognition. IEEE Trans Image Process 31:3224–3235. https://doi.org/10.1109/TIP.2021.3137005
21. Zhou C (2019) Measure face similarity based on deep learning. KTH, Skolan för elektroteknik och datavetenskap (EECS)
22. Nguyen TT-L, Le D-L, Nguyen V-D (2022) Siamese network in face verification online learners. In: 2022 RIVF international conference on computing and communication technologies (RIVF), pp 279–282. https://doi.org/10.1109/RIVF55975.2022.10013795
23. Wu H, Xu Z, Zhang J, Yan W, Ma X (2017) Face recognition based on convolution siamese networks, pp 1–5. IEEE https://doi.org/10.1109/CISP-BMEI.2017.8302003
24. Mathi R, Mothukuri JV, Pasumarthy VA, Suja P, Subramani R (2022). Face recognition in different scenarios using siamese network. https://doi.org/10.1109/gcat55367.2022.9971852
25. Sáez Trigueros D, Meng L, Hartnett M (2018) Enhancing convolutional neural networks for face recognition with occlusion maps and batch triplet loss. Image Vis Comput 79:99–108. https://doi.org/10.1016/j.imavis.2018.09.011
26. Holkar A, Walambe R, Kotecha K (2022) Few-shot learning for face recognition in the presence of image discrepancies for limited multi-class datasets. Image Vis Comput 120:104420. https://doi.org/10.1016/j.imavis.2022.104420
27. Zeng K, Ning M, Wang Y, Guo Y(2020) Hierarchical clustering with hard-batch triplet loss for person re-identification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13657–13665
28. Yuan Y, Chen W, Yang Y, Wang Z (2020) In defense of the triplet loss again: learning robust person re-identification with fast approximated triplet loss and label distillation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) Workshops
29. He Z, Su W, Bi Z, Wei M, Dong Y, Xu G (2019) The improved siamese network in face recognition, pp 443–446. IEEE. https://doi.org/10.1109/ICICAS48597.2019.00099
30. Steven Hendryli J, Herwindiati DE (2020) Siamese network's performance for face recognition. In: 2020 IEEE international conference on sustainable engineering and creative computing (ICSECC), pp 141–145. https://doi.org/10.1109/ICSECC51444.2020.9557529
31. Sharma A, Gautam R, Singh J (2023) Deep learning for face mask detection: a survey. Multimed Tools Appl 82(22):34321–34361. https://doi.org/10.1007/s11042-023-14686-6
32. Kamil MHM, Zaini N, Mazalan L, Ahamad AH (2023) Online attendance system based on facial recognition with face mask detection. Multimed Tools Appl 82(22):34437–34457. https://doi.org/10.1007/s11042-023-14842-y
33. Rafidison MA, Rakotomihamina AH, Rafanantenana SHJ, Toky RFM, Raoelina MMN, Ramafiarisona HM (2023) Neural networks contribution in face mask detection to reduce the spread of covid-19. Multimed Tools Appl 82(21):32559–32581. https://doi.org/10.1007/s11042-023-14920-1
34. Heidari M, Fouladi-Ghaleh K (2020) Using siamese networks with transfer learning for face recognition on small-samples datasets, pp 1–4. IEEE. https://doi.org/10.1109/MVIP49855.2020.9116915
35. Sharma S, Kumar V (2021) Performance evaluation of machine learning based face recognition techniques. Wirel Pers Commun 118(4):3403–3433. https://doi.org/10.1007/s11277-021-08186-9
36. Li Y, Lu Z, Li J, Deng Y (2018) Improving deep learning feature with facial texture feature for face recognition. Wirel Pers Commun 103(2):1195–1206. https://doi.org/10.1007/s11277-018-5377-2
37. Sharma R, Patterh MS (2015) A new hybrid approach using pca for pose invariant face recognition. Wirel Pers Commun 85(3):1561–1571. https://doi.org/10.1007/s11277-015-2855-7
38. Petpairote C, Madarasmi S, Chamnongthai K (2021) 2d pose-invariant face recognition using single frontal-view face database. Wirel Pers Commun 118(3):2015–2031. https://doi.org/10.1007/s11277-020-07063-1
39. Chen W, Chen X, Zhang J, Huang K (2017) Beyond triplet loss: a deep quadruplet network for person re-identification. CoRR https://doi.org/10.48550/arXiv.1704.01719
40. Taigman Y, Yang M, Ranzato M, Wolf L (2014) Deepface: Closing the gap to human-level performance in face verification. In: 2014 IEEE conference on computer vision and pattern recognition, pp 1701–1708. https://doi.org/10.1109/CVPR.2014.220

Authors and Affiliations

Sushant Jain[1] · Amit Pundir[2] · Sanjeev Singh[1] · Geetika Jain Saxena[2]

Sushant Jain
jain.sushant@gmail.com

Amit Pundir
amitpundir@mac.du.ac.in

Sanjeev Singh
sanjeev@south.du.ac.in

[1] Institute of Informatics and Communication, South Campus, University of Delhi, Delhi 110021, Delhi, India

[2] Maharaja Agrasen College, University of Delhi, Delhi 110096, Delhi, India