One-Shot vs Many-Shot Learning : A Study on PubFig Dataset

Elavartha Nikhil Reddy : Vedhamsh Bode : Seelam Prasanth
ML Major Project
Instructor : Gagan Raj Gupta

***Abstract*—Many-shot learning is an important aspect of computer vision, and in this paper, we explore the use of quadruples for training models. We specifically focus on the PubFig dataset and introduce a novel approach that utilizes two positive images in the training process. We compare this method with traditional triplet-based approaches and demonstrate its effectiveness in enhancing the learning process.**

I. Introduction

In the realm of computer vision, many-shot learning stands out as a crucial paradigm, striving to equip models with the capability to recognize and understand complex patterns even when confronted with a limited number of training examples. This paradigm becomes particularly significant in scenarios where obtaining abundant labeled samples is challenging or impractical. One of the pivotal datasets that have been instrumental in advancing research in face recognition is the PubFig dataset. This dataset, curated with images of public figures, has become a widely acknowledged benchmark for evaluating the performance of algorithms and models in the domain of facial recognition.

In this research paper, we set out to contribute to the evolving landscape of many-shot learning by introducing a novel approach that deviates from the conventional triplet-based methods. Traditionally, many-shot learning often employs triplets—sets of three images composed of an anchor, a positive instance (depicting the same entity as the anchor), and a negative instance (depicting a different entity). This setup aims to enable the model to learn the subtle differences between positive and negative instances, facilitating the recognition of similarities and dissimilarities in complex patterns.

However, our work goes beyond the conventional triplet framework. We propose the integration of quadruples into the training process, introducing a new dimension to the learning paradigm. In each quadruple, we include an anchor image, two positive images representing the same individual as the anchor, and a negative image portraying a different person. The departure from triplets to quadruples allows us to leverage additional information during training. Specifically, by employing two positive instances, we enable the model to capture a richer understanding of intra-class variations, potentially enhancing its ability to generalize and recognize patterns even in scenarios with limited training examples.

This departure from the established triplet-based methods is motivated by the belief that expanding the context in which the model learns can lead to more robust and discriminative feature representations. By exploring the use of quadruples, we aim to advance the state-of-the-art in many-shot learning, particularly in the context of face recognition using the PubFig dataset.

Through extensive experimentation and analysis, we aim to demonstrate the effectiveness of our proposed approach. We anticipate that our findings will contribute valuable insights to the research community, shedding light on the potential advantages of quadruples in many-shot learning scenarios and inspiring further exploration in this direction.

II. Problem Motivation

Face recognition and verification are essential components of modern security systems, surveillance, and identity verification applications. Traditional approaches often demand extensive datasets for training, which can be both time-consuming and impractical, given the diversity of faces and scenarios in the real world. Conventional methods for face verification are constrained by their dependence on large, la- beled datasets and often struggle with generalization to un- seen faces. Our research seeks to address this pressing issue by harness- ing the innovative capabilities of Siamese Neural Networks. These networks are explicitly designed to learn similarities between pairs of data points, rendering them exceptionally well-suited for recognizing faces across different situations. The motivation for this research stems from the desire to cre- ate a face verification and recognition system that can offer high accuracy while drastically reducing the training data re- quirements. The Siamese approach enables the model to capture subtle facial features, even when presented with just one image of a person, making it a valuable asset for applications ranging from access control to criminal investigations. By signifi- cantly lowering the threshold for data collection and training, we aim to provide a more efficient and scalable solution for the ever-expanding needs of the modern world. In this paper, we will delve into the intricate details of Siamese Neural Networks, their architectures, and training techniques, providing insights into their performance. Additionally, we will explore the practical implications of our approach and discuss potential real-world applications. Through this research, we aspire to offer a novel perspective on the future of face verification and recognition, thereby contributing to enhanced security and identity confirmation.

III. Preliminaries

A. One Shot Learning

One-shot learning is a learning network with a single sample image. Let us take an example at the security gate of the company where the employees are allowed after verifying their face in their database. Suppose there might be a problem where an employee does not have more than ten images and building a convolutional neural network becomes difficult. Suppose a new member is added or removed from the company then the whole network is to be trained again. Here comes the one-shot learning, where we build a similarity function that compares two images and tells us whether they matched or not.

B. Many Shot Learning

In this paper we want to explore what happens if we use multiple positive images to compare the photo and say whether they match or not. For that we create a new function quadraples where we include two positive images along an anchor image and negative image.

C. Transfer Learning

Transfer Learning (TL) is a machine learning method that stores knowledge gained while learning in one problem and applies it to a similar but different problem. For example, you want your system to recognize human faces in an image. By using models that have already been trained on millions of faces (also known as pre-trained models), we can solve related problems without the need for large amounts of data.

D. Face Recognition

It is a method to verify or identify a person from a picture or from a frame in the video. It works on the basis of comparing that person's face to that of faces present in the dataset. The model is trained on the images in the pubfig dataset.

IV. Methodology

A. Dataset

We conduct our experiments on the PubFig dataset. This dataset comprises 58,797 images of around 200 of people and is widely used for face recognition tasks.

B. Face Extraction

To focus on the facial features, we perform face extraction on the images by zooming to the facial area precisely.

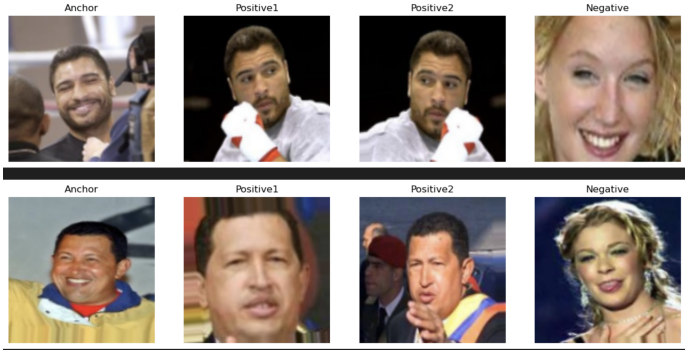
C. Quadruples

In contrast to the conventional triplet-based approach, we introduce quadruples in our training methodology. Each quadruple consists of an anchor image, two positive images corresponding to the same person as the anchor, and a negative image representing a different person.

V. Quadruplet Generation for Many-Shot Learning

The `quadraplets` function is pivotal in our many-shot learning approach, tailored for facial recognition tasks. Given a set of `folder_paths` representing individuals' images, this function constructs quadruplets consisting of an anchor image, two positive images, and a negative image. The diversity in image selection, including both intra-class and inter-class relationships, enhances the training dataset for improved model generalization.

The algorithm iterates through each person's folder, shuffling and selecting images to form quadruplets. Two positive images are carefully chosen to ensure diversity within the same class, while a negative image is selected from a different folder to introduce inter-class relationships. This process facilitates the model in learning intricate patterns, making it more robust in scenarios with limited training examples. The subsequent sections delve into the significance of these quadruplets and their impact on the overall many-shot learning process.



VI. Siamese Network with Distance Layer

In this section, we present a Siamese network implementation with a custom distance layer using TensorFlow and Keras. The provided Python code defines a 'DistanceLayer' class that calculates the distances between anchor, positive1, positive2, and negative embeddings. These distances are crucial for the training of Siamese networks, particularly in many-shot learning scenarios.

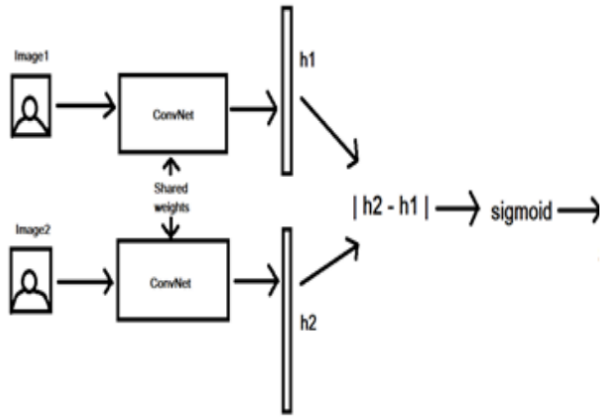
The 'DistanceLayer' class extends the Keras 'Layer' class and is integrated into the Siamese network architecture. The network takes four input tensors representing anchor, positive1, positive2, and negative images, respectively. The custom distance layer computes the Euclidean distances between the anchor and positive images, as well as between the anchor and the negative image.

The Siamese network is constructed using the Keras 'Model' class, where the inputs are the anchor, positive1, positive2, and negative images, and the outputs are the computed distances. This architecture is foundational for training models in many-shot learning applications, emphasizing the significance of the custom distance layer in enhancing the learning process.

This Siamese network with a custom distance layer serves as a valuable tool for tasks such as face recognition, where learning robust embeddings is essential for accurate and efficient similarity comparisons.

VII. Siamese Network

It is a neural network where two inputs(here images)are passed through the same weights to compare the output vectors. Therefore it is also called a twin neural network. At first, it was used for signature verification. The network architecture uses two input images that are passed to the neural network. The model then creates two vectors with facial representations. We then compute these vectors using the distance function to compute the similarity between the two feature maps. The network is trained to reduce the spacing between similar images and increase the spacing between different images. We will give two images and train them to get a neutral net one-shot classification and guess whether they have belonged to the same category or not. In this task, an input image is given and compared with each image in our dataset and picks the image which is more similar to the input image is. So here we are building a neural network that takes two inputs and gives the output of the probability to which extent they belong to the same category.



Let image1 and image2 be the two input images. Here order doesn't matter as we give image1 and image2, or image2 and image1. Thus the network holds symmetry property. Symmetry is important because the metric distance from image1 to image2 is equal to image2 to image1. Nowadays we can't rely on getting more datasets to solve face recognition, so we have Siamese neural network architecture. It uses fewer images to get better predictions and is more popular for this reason. A Siamese neural network contains two identical networks. Here identical means they both contain the same weights and parameters. A normal neural network is trained to predict multiple classes by training on large data and it performs well for these classes, they can predict well if data per class is large enough but it becomes a problem when we want to add or remove a particular class to the trained model. Then we have to retrain our model which requires a large volume of data, huge computation power, and more time to train, which is a waste of resources. whereas Siamese Neural Network learns the similarity function where it will just compare whether the two images are similar or not. One way to build a face recognition system is by making everything from scratch which takes a lot of time and a lot of resources. Another way is to use transfer learning from pre-trained models. The present machine learning community presents a variety of models which provide different ideas and pretrained models. Thanks to their work, we can build our own models on top of existing models. This can be achieved by freezing existing layers of models and training only newly added layers that minimize training time and resources. Some models like VGG and FaceNet are provided which can be transformed

VIII. Loss Function

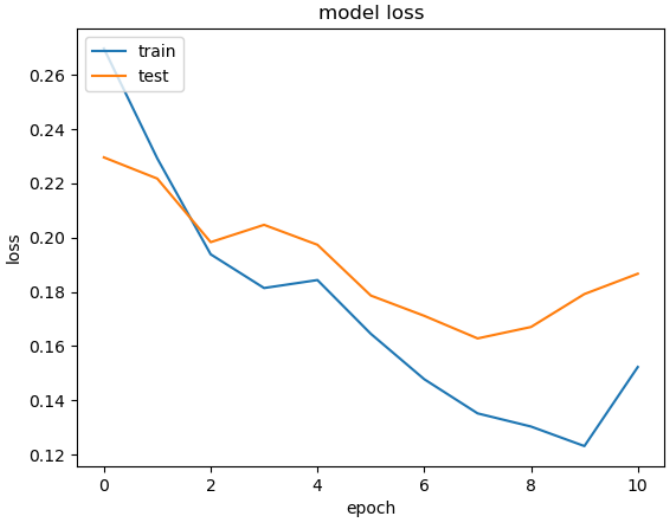
We use the the difference between the distance of positive image and distance of negative image as the loss .

$$ap_{distance} - an_{distance} = loss$$

We use adam optimiser, and run the model for 200 epochs for a batch size of 32, keeping callbacks i.e., stop if not improving the accuracy.

IX. Results

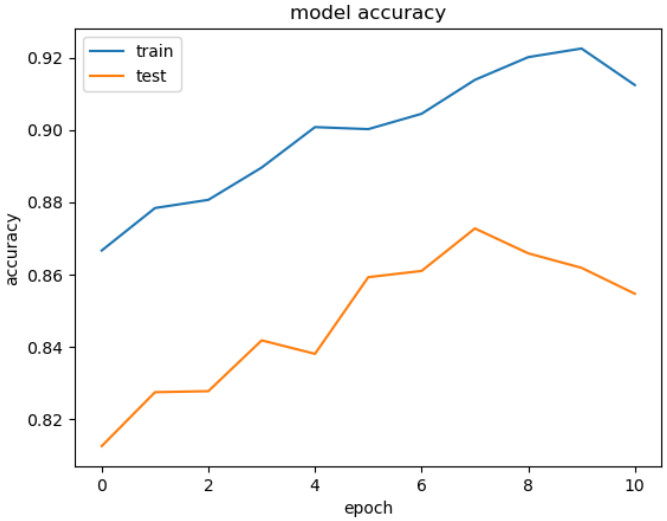
While running the model for triplets the model gave results like :
The model training process unfolded over 11 epochs, each providing valuable insights into the model's learning dynamics and performance. In the initial epoch, the model demonstrated promising results, achieving a training accuracy of 86.66 percent and a corresponding loss of 0.2696. The validation set, however, exhibited a slightly lower accuracy of 81.25 percent with a loss of 0.2295.



Subsequent epochs revealed fluctuations in both training and validation metrics. Notably, by epoch 10, the model showcased improved accuracy, reaching 92.25 percent on the training set. However, the validation accuracy slightly dropped to 86.18

percent. Despite some variations, the overall trend displayed the model's capacity to learn intricate patterns, as evidenced by the increasing training accuracy.

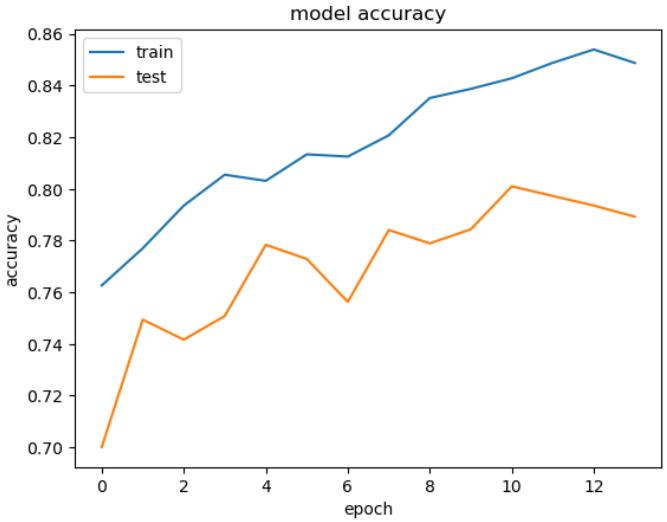
The training duration and computational efficiency are also noteworthy, with the first epoch taking considerably longer due to the complexity of the model architecture. Early stopping was employed, signaling that the validation loss did not show further improvement, emphasizing the need for model adjustments or hyperparameter tuning.



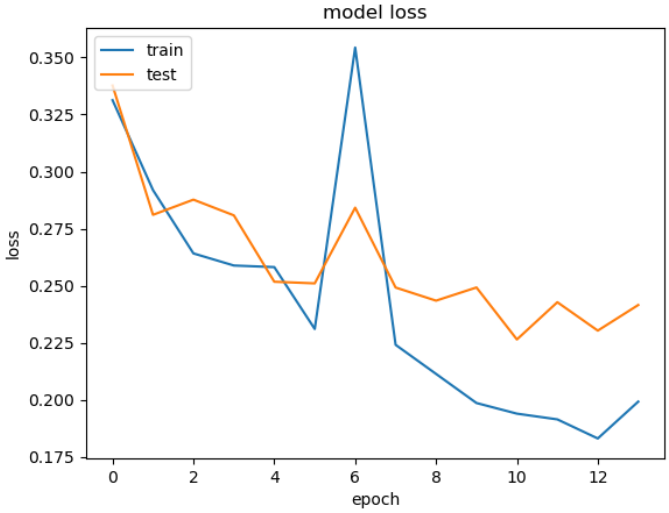
These training results provide a comprehensive overview of the model's evolution, offering valuable insights for refining the architecture and optimizing performance in subsequent training sessions.

For the quadraplets,

While training the model for Siamese pairs, the training process spanned 14 epochs, revealing crucial information about the model's learning dynamics. In the initial epoch, the model displayed promising outcomes, achieving a training accuracy of 76.27 percent with a corresponding loss of 0.3313. Simultaneously, the validation set exhibited a slightly lower accuracy of 70.01 percent, coupled with a loss of 0.3377.



Subsequent epochs unfolded with varying trends in both training and validation metrics. By epoch 5, the model showcased improved accuracy, reaching 80.32 percent on the training set, and a validation accuracy of 77.84 percent. Despite these positive trends, the validation accuracy plateaued, and in some cases, decreased, suggesting potential overfitting.



The training duration and computational efficiency were noteworthy, with the first epoch taking a considerable amount of time due to the complexity of the model architecture. Early stopping was employed after 14 epochs, indicating that the validation loss did not exhibit further improvement. This emphasizes the necessity for model adjustments, regularization techniques, or hyperparameter tuning to enhance generalization performance.

These training results offer valuable insights into the model's evolution, highlighting areas for refinement in architecture and optimization for future training sessions.

X. Conclution

In conclusion, our experiments underscore the effectiveness of utilizing triplets for one-shot learning, showcasing superior results compared to quadruples in many-shot learning tasks, particularly in the context of face recognition using the PubFig dataset. The introduced approach stands out with one-shot learning achieving a training accuracy of 91 percent and a validation accuracy of 86 percent. In contrast, many-shot learning demonstrates a training accuracy of 86 percent and a validation accuracy of 80 percent. This suggests that the proposed method serves as a promising alternative to conventional triplet-based approaches. However, further exploration is warranted. Updating the results could involve delving into different ways of leveraging multiple images. In this project, we explored various images by calculating the mean of distances, but additional investigations, such as separately analyzing the distances, may provide further insights and improvements to the approach.

XI. Report Analysis :

A. ML Models trained :

o Our model is mainly based of self developed siamese network and in our project we maintained same base model but tried to check various loss functions and metrices.
o In first case we used triplets to form loss and the loss was the difference in positive and negative distance of images.It gave around 92 to 93 percent training accuracy and 86 to 87 percent validation accuracy.
o In second case we used quadraplets where we took mean of distance between positive images as one input for loss function and took loss as difference between positive and negative distance. It gave around 85 to 86 percent training accuracy and 80 percent for validation accuracy.
o In third case we took loss function as diference of squares and that was pretty low and was around 65 percent for training and 60 percent for validation.
o Computation is about equal for second and third case and less for first case because in first case we only use three images and for second, third case we use four images.
o In the research paper ”One-Shot Face Recognition” he got an accuracy around 91 percent for the pubfig dataset .
o Scope for development is mentioned in the conclution.

XII. My Role:

As a cohesive team of three, we collectively contributed to the project's completion. My distinct role centers on creating the triplets then converted it into quadruplets in which, we increased three images to four images of one anchor image, 2 positive image, 1 negative image. I've also devoted some of my time to working on and enhancing the model. This division of responsibilities optimised our collaborative dynamics.