COMP9444 Project Summary
Deep Face Recognition

Nam Anh Dang $z5310809$ | Purnjay Peshawaria $z5215940$ | Matthew Su $z5257413$ | Lyn Wang $z5313427$ | Danver Zhao $z5317086$

August 2022

1 Introduction

Facial recognition research has found extensive applications for convolutional neural networks (CNNs). Our project aimed to demonstrate and analyse the performance of state-of-the-art CNNs in the multiclass classification of faces. We investigate architectures that allow a network to be trained accurately on small-scale datasets, which is helpful when limited resources exist for creating and processing large training sets.

2 Methods

2.1 Siamese network

We implemented and trained a Siamese network [2] using 2 VGG CNNs with 8 layers each. Thus, the architecture of the network consisted of multiple convolutional layers, with the ReLU activation function and max pooling on each layer. The model receives a pair of face images as input and outputs their feature vectors. Rather than triplet loss, our Siamese network was trained with a contrastive loss function which was applied to the outputted feature vectors of the 2 CNNs. To test the network, we fix one of the input images and iterate through all other face images, computing the Euclidean distance between each pair of output vectors, where we expected the fixed image to have the shortest distance from an image of the same identity.

2.2 VGG Face

We used as our next model a Pytorch implementation of the VGG Face CNN, based on VGG-16 introduced in [6], to improve upon the VGG CNN that was used as the subnetworks for our Siamese model.

The Visual Geometry Group found in [4] that training the model is more efficient when the network is bootstrapped as a linear classifier. We copied the weights and biases from a pre-trained VGG model, and our model used these parameters to learn the identities from our training set.

After the weights were loaded into the model, the linear classifier output layer was removed and replaced with a new layer, the embedding layer, to represent the feature vector of an inputted image. As per [4], the network was trained to learn a face embedding using the triplet loss function. Every identity in the training set had at least one triplet generated where an image of the identity was used as the anchor. Positives were selected at random from other face images with the same label as the anchor, while negatives were selected at random from the remaining face images. For validation, an additional set of triplets was randomly generated.

The network was functionally tested by inputting a new face image, obtaining the feature vector as output, then computing its Euclidean distance from the feature vectors of every other image in the test dataset. The 10 face images that were closest in Euclidean space to the new face image were then displayed, with the expectation that the other images of the same identity would be among those 10.

3 Experimental setup

We used an $80-20$ split of the Pins Face Recognition dataset from Kaggle [7] as the training and validation sets for our VGG Face implementation. The dataset has 105 unique identities, or classes, with at least 95 images of each identity, totalling 17640 face images sourced from Pinterest. The test dataset was sourced from Labelled Faces in the Wild dataset (LFW) [3], which contains 13233 images of 5749 people, as well as images of our own faces. Stochastic gradient descent optimisation was used with a learning rate of 0.025.

For training and testing the Siamese network, we used the Pins and LFW datasets, and the AT&T Database of Faces [1][5]. The AT&T dataset contains 400 images of 40 identities — 10 images per identity. Adam optimisation was used with a learning rate of 0.0005.

4 Results

4.1 Siamese network

No. of epochs	Distance from `purnjay_3.jpg`	Distance from closest image	Closest image is match
25	0.0607	0.0607	True
50	0.3151	0.0555	False
100	0.1543	0.0056	False
150	1.0273	0.1354	False

Table 1: Siamese model output when testing with an image of a team member, `purnjay_1.jpg`, as the fixed input. The closest image was a match only when the model was trained for 25 epochs.

We found that the Siamese model was unable to identify a person if presented with the side of their face, but could identify a straight portrait, after 25 training epochs. The model could identify the side of the face with 50, 100, and 150 epochs in some instances, but often failed to identify the match, as in Table 1. Thus, our Siamese network is less accurate than state-of-the-art models; although we expected the model to perform better on limited data than a VGG CNN with 8 layers, our model may need to be trained on a larger dataset to see an improvement in accuracy.

4.2 VGG Face

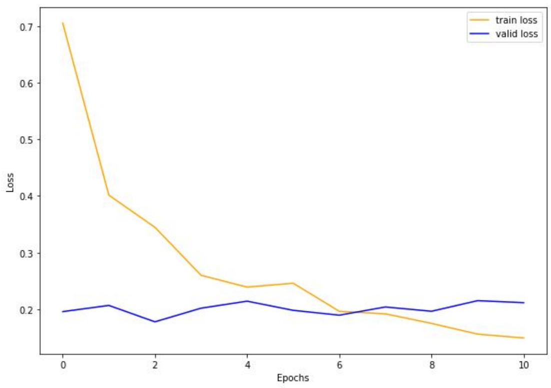

Figure 1: Training and validation loss for the VGG Face model.

The VGG Face model was trained for 10 epochs. The accuracy of the model at identifying an input's match among its 10 nearest neighbours was 0.417. Its accuracy for 100 nearest neighbours was 0.661, while for 1 nearest neighbour, it was 0.186. This accuracy was lower than implementations of VGG Face in the literature and of other state-of-the-art models, and can be attributed to our limited data, since the Pins dataset only contains 105 identities.

5 Conclusions

The models created from our study had low accuracy which do not make them deployable in real-world applications. The aim of our study was to successfully train a network using small-scale datasets, however, this proved to be a challenge; the study could have benefited from a larger dataset with more identities and more images per identity. Future work must involve the curation of a larger dataset, to improve overall accuracy of both models as well as accuracy for recognising side profiles. Further hyperparameter tuning, involving experimentation with more learning rates, could also be explored.

References

[1] *AT&T Database of Faces (previously Olivetti Research Laboratory Face Database)*. https://github.com/paviro/MMM-Facial-Recognition-Tools/tree/master/training_data/negative.

[2] J. Bromley et al. "Signature verification using a siamese time delay neural network". In: (1994). DOI: https://doi.org/10.1142/S0218001493000339.

[3] G. B. Huang et al. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. https://people.cs.umass.edu/~elm/papers/lfw.pdf. 2008.

[4] O. M. Parkhi, A. Vedaldi, and A. Zisserman. "Deep Face Recognition". In: (2015). DOI: https://doi.org/10.5244/c.29.41.

[5] F. Samaria and A. Harter. "Parameterisation of a stochastic model for human face identification". In: (1994). DOI: https://doi.org/10.1109/ACV.1994.341300.

[6] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: (2014). DOI: https://doi.org/10.48550/arXiv.1409.1556.

[7] B. Toy, V. Shanawad, and S. Ishii. *Pins Face Recognition*. https://www.kaggle.com/datasets/hereisburak/pins-face-recognition. 2020.