CS230

Heavy-Lifting Pre-training for Multi-facial Recognition
Project Category: Computer Vision

J.D. Kelly
Department of Electrical Engineering
Stanford University
jdkelly@stanford.edu

Gilbert L. Rosal
Department of Computer Science
Stanford University
rosalg@stanford.edu

Troy H. Lawrence
Department of Computer Science
Stanford University
troylaw@stanford.edu

1 Introduction

Facial recognition performed by computers (the classifying of individuals appearing within images) has been a field undergoing research since the 1960s with Woody Bledsoe's team's highly mathematical approach Pandya et al. (2013). With the increase of computing power that came with the passing of time, other approaches aimed to tackle this challenge up to and including the present day. We also mean to improve on the facial recognition task through more rigorous pretraining. Specifically, we pretrain our model on the Labelled Faces in the Wild dataset and then further train it on the PubFig dataset. Each of these two datasets offer different styles of images containing faces and allows our model greater depth for learning. We will continue the development of our model through the addition of object detection to allow for multi-facial recognition, from which we can run our recognition on each image subset containing a face.

2 Related Work

A few decades after the work of Bledsoe, the Defense Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology (NIST) collaborated on the Face Recognition Technology project in the mid-1990s. The culmination of their work was the advanced use of eigenfaces, which could detect faces using low-dimensional linear algebra Phllips et al. (1996). From this point on, artificial intelligence, particularly in the form of convolutional neural networks, achieved much higher success rates in both facial detection and recognition. In 2014 and 2015 papers introducing both DeepFace and FaceNet respectively were published. DeepFace was introduced by Facebook and introduces a nine-layer deep neural network with explicit 3D modeling to derive a race representation Taigman et al. (2014). This model achieved a 97.35% accuracy on the Labeled Faces in the Wild (LFW) dataset. FaceNet, introduced the following year by Google, maps face images to a compact Euclidean space to measure face similarity using triplet loss Schroff et al. (2015). Triplet loss takes one image of a face as an anchor, another image of the same face but in a different setting as the positive, and an image of a completely different face as the negative, and then trains on the difference between the positive and the negative to the anchor. FaceNet achieved a facial recognition rate of 99.63% on the LFW dataset and achieves 95.12% on the YouTube Faces DB. . Most recently,

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

the Arcface proposed by Deng et al. (2021) has reached success rates as high as 99.85% through its introduction of Additive Angular Margin Loss. These models surpass the human score of 97.53% and thus will be difficult to improve on. However we wish to take a different approach at the problem by introducing various datasets, finetuning around them, and then adapting this model to handle object detection.

3 Github Code

Our github repo can be found here:
`https://github.com/rosalg/cs230_final_project`

4 Dataset

Our current approach leverages two datatsets. The first is the LFW dataset previoulsy mentioned. This dataset contains over 13,000 images of approximately 5,750 individuals collected from the web Huang et al. (2007). We leverage this dataset for initial pretraining due to its breadth and popularity in the field. However for further tuning and testing, we have used the PubFig dataset which is a large, real-world face dataset consisting of 58,797 images of 200 people collected from the internet Kumar et al. (2009). This dataset has far more samples and importantly better coverage on the individuals in the dataset with images in various different environments, outfits, and so on, while in many cases the LFW dataset may only have images on one individual from the same event, in the same environment, lighting, and clothes Kumar et al. (2009). This is far more indicative of the intended application of our model. Based on the initial images from the LFW dataset, we used images with dimensions of 250 x 250 pixels for our input, PubFig images were resized slightly from 256 x 356 to fit this dimension. All images were 3 channel RGB, and we normalized the pixels values to between 0 and 1 across all channels. Furthermore we restricted the images we used from LFW to those with at lease two samples per individual, which further reduce the dataset to 9146 images with 1680 individuals represented.

5 Current Approach

As explored extensively in existing literature, we currently use a deep convolutional neural network (DCNN). Our baseline model is inspired by the first model in the FaceNet paper Schroff et al. (2015); this model demonstrated impressive performance and could easily be modified to our needs by altering the network depth due to its structure. Within each convolutional block we used a 2D Convolutional Layer with 64 filters, a kernel size of 2 and Relu activations, along with max pooling, and dropout regularization layers. In order to improve on time-efficiency, which was a primary goal of our project, we added skip connections in-between convolutional blocks. Our model has a final fully connected layer that outputs a 256 dimension image embedding with L2 normalization.

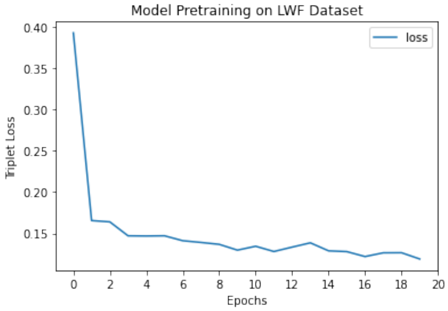
As discussed in our Dataset section, we leverage the LFW dataset for pretraining due to its large coverage and breadth, and further train on the PubFig dataset for final evaluation for more samples on individuals and more representative environments for positive sample comparison.

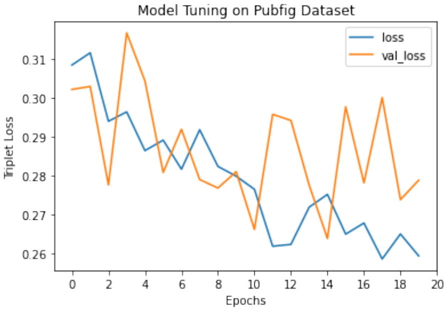
During training we used an Adam Optimizer with 1e-3 learning rate and a Triplet Loss function to encourage the embeddings among the same class, or in this case individual, to be as close as possible while maximizing the distance between images of different individuals.

$$\mathcal{L}(A, P, N) = max(||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + \alpha, 0) \tag{1}$$

Where f(I) is the image embedding output of our model on some image I, A is some anchor image, P is a positive sample image, N is a negative sample, and alpha is some margin we defined as the loss default of 1.

For pretraining on LFW, all data was used for training and a batch size of 32 was selected as the only goal was to better understand the basic image data for image embeddings and speed up training time for the intended evaluation dataset. For development and test on the PubFig evaluation dataset, we used a 60%/20%/20% Training/Validation/Test split while still using the same 32 image batch size. Plots for the loss over training epochs can be seen below.





Of note, we have been significantly constrained by model training time. As we have fine tuned parameters and model architecture we have limited training epochs to ensure iteration time is manageable. We also see that loss when tuning on the PubFig Dataset is abnormal and slightly over-fitting the training set. We will discuss our further steps to address these issues and improve on performance in the Future Work section.

6 Current Performance

We evaluate the model performance on the PubFig dataset. For each class in the Dataset, we have one image that serves as a reference image for each individual. We store image embeddings for all of these reference images. For each test image, we use the model to compute an image embedding for the test sample. We then compare the L2 difference of this embedding with all stored reference images. As long as the smallest difference is below some threshold we have defined, we assign the test image to same class as the reference image embedding it is closest to. If the difference is above our threshold, then we predict that the test face is not in our reference database.

Currently our model's facial recognition accuracy on the PubFig test set is 11.89%. Clearly this significantly under-performs models in existing literature, however, it shows significant learning compared to what would be expected out of a random model, and we expect to continue to see vast improvement gains as we pursue next steps and increase training epochs.

7 Future Work

Our future work will mainly consist of two aspects, prioritized in the order they're given:

1. Improve our model performance
   1.1 Try experimenting with different training and fine-tuning datasets: what's the minimum we can use to reach baseline performance?
      1.1.1 Try training or fine-tuning on the Google Facial Expression Comparison Dataset Vemulapalli and Agarwala (2018)
      1.1.2 Try training or fine-tuning on the Youtube Faces Database Wolf et al. (2011)
   1.2 Try experiment with different triplet loss variants
      1.2.1 What is the impact of trying the same positive and anchor, but changing the negative and having any number of negatives applied to the same positive and anchor examples? (i.e. 5 training dataset examples with the same anchor and positive, but 5 different negatives. Semi-Hard Triplet Loss is another potential loss function leveraged in FaceNet we could explore.)
2. Build object detection specifically for faces
   2.1 Take as input some images with 0 to any number of faced

8 Project Novelty

We used the large LFW dataset for pre-training and further fine tune on the PubFig dataset. We then will adapt our model to perform object detection and run recognization on each face-labelled image subset. By introducing skip layers, we aim to perform such recognition in a less computationally expensive way to return trustworthy results quickly.

9 Contributions

JD Kelly - Model Development/Training; Paper Writing

Troy Lawrence - Data Pre-processing; TA Interfacing; Paper Writing

Gilbert Rosal - Data Pre-processing; Research; Paper Writing

References

Deng, J., Guo, J., Yang, J., Xue, N., Cotsia, I., and Zafeiriou, S. P. (2021). ArcFace: Additive angular margin loss for deep face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.

Kumar, N., Berg, A., Belhumeur, P., and Nayar, S. (2009). Attribute and Simile Classifiers for Face Verification. In *IEEE International Conference on Computer Vision (ICCV)*.

Pandya, J., Rathod, D., and Jadav, J. (2013). A survey of face recognition approach. *International Journal of Engineering Research and Applications*, 3.

Phllips, J., Rauss, P., and Der, S. (1996). *FERET (Face Recognition Technology) Recognition Algorithm Development and Test Results*. Army Research Laboratory.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Vemulapalli, R. and Agarwala, A. (2018). A compact embedding for facial expression similarity. *CoRR*, abs/1811.11283.

Wolf, L., Hassner, T., and Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, pages 529–534.