Ensemble Networks for Better Facial Recognition of Bearded Faces

Edward Vendrow
Computer Science
Stanford University
evendrow@stanford.edu

Akash Singhal
Mechanical Engineering
Stanford University
akash13@stanford.edu

Amogh Dixit
Mechanical Engineering
Stanford University
addixit@stanford.edu

Abstract

Face recognition systems such as FaceNet[9] perform poorly when certain facial features are obscured[10]. We propose an ensemble network architecture which combines FaceNet with a specialized secondary network for face recognition in the presence of the facial obscurity, as well as a dispatcher network to decide which face recognition model to use. We apply this architecture to bearded faces, and demonstrate superior performance over standalone face recognition systems. Our architecture extends to an arbitrary number of facial obscurities, indicating a potential for significant improvement to face recognition systems in general.

1 Introduction

In today's age, facial recognition systems have become extremely prevalent and are almost ubiquitous in modern security systems. However, facial recognition models do not perform well for faces with obscured features because it is difficult to incorporate the specific obscurities into a recognition model. As an example, the presence of beards can adversely affect the performance of facial recognition models because beards make it difficult for the model to capture some facial landmarks such as the boundary of the face and other features around the jawline that are part of the identity of a person. Thus, using any number of such obscurities people can bypass facial recognition systems[14], consequently affecting the effectiveness of the technology. For this reason, it is important to enhance the facial recognition models such that they can be used in a reliable manner and not be tricked by obscurities. For this project, we take the images of two people and output whether or not they are the same person. The notion was to improve the accuracy of facial recognition models when the input images are of the same person but one image has a beard and the other does not. In this case, facial recognition does not work as well and our goal was to improve it.

2 Related work

The availability of massive datasets and computing power has led to rapid improvements in facial recognition technology. The current state-of-the-art facial verification and recognition systems such as DeepFace[13] and FaceNet[9] have achieved accuracies higher than 95% on multiple datasets scuh as Labelled Faces in the Wild (LFW), YouTube Faces (YTF) and Social Face Classification (SFC). Such developments have inspired adversarial attacks aimed at circumventing such systems. Research has been done on designing special perturbations that can fool these facial recognition systems. In [14], the authors present attacks that enable a person to impersonate someone or dodge facial recognition systems entirely. Adding specially designed glasses caused 3 separate Deep Neural Networks to misclassify people with very high levels of success (> 80%). The accuracy of facial recognition networks has also been observed to suffer without specialised attacks when certain landmarks are obscured by sunglasses, beards and small masks[15].

,

We propose an ensemble network that attempts to deal with a specific facial obscurity, namely, the presence of beards. This methodology can be extended to other obfuscations by training on datasets with sufficient examples of that particular obfuscation.

3 Dataset and Features

For our project we had to collect a custom dataset and could not directly use other datasets such as LFW or CelebA because we did not want to count just some stubble as beard, which is what CelebA does. Hence, just segregating the CelebA dataset by the beard attribute was not good enough for us. Also, for our specialized secondary network, we needed images of the same person with and without a beard, which is not easily available in CelebA. Hence we mostly downloaded individual images from the internet.

We had two separate parts to our dataset. For the first part of the dataset we used some images from CelebA and downloaded other individual images from the internet (to get more identities). We had a total of 800 images with beards and 800 without. The number of identities was upwards of 400. For the second part of our dataset we collected images of the same person with and without beards. For this, we collected approximately 1650 images with 60 unique identities with each image downloaded individually from the internet. These datasets are different because in the first part of the dataset we are not concerned with the presence of obscurities (other than beards) and we also do not require the image of the same person with and without a beard. This is because we only use this first dataset to train the dispatcher network (checking for the presence of a beard). Hence the presence/absence of other obscurities is irrelevant. For the second part of the dataset, we are careful to collect images without any obscurities (other than beards) as this can affect identification and we also need both bearded and non-bearded images of the same person. This second dataset is also used for the training/testing of the dispatcher network, however it is the only data used for the training/testing of our specialized secondary network.

For the division of our training and testing sets, we use 3043 images for training the dispatcher network and test it on the remaining 200 images. Whereas, for training and testing our specialized network we use 52 identities ($\sim$ 1400 images) from our set of 60 to train the network and test it on the remaining 8 identities ($\sim$ 250 images). We also train the conventional FaceNet model on two datasets. We use the Labelled Faces in the Wild (LFW) dataset for getting a baseline accuracy of the model and further test in on the 8 identities ($\sim$ 250 images) that form the test set for our specialized Secondary Network.

We aligned and cropped the images in our dataset using Multi-task Cascaded Neural Network (MTCNN). We also augmented our dataset through standard procedures of cropping, rotation, and flipping. An example of the augmentation is seen in Figure 1.

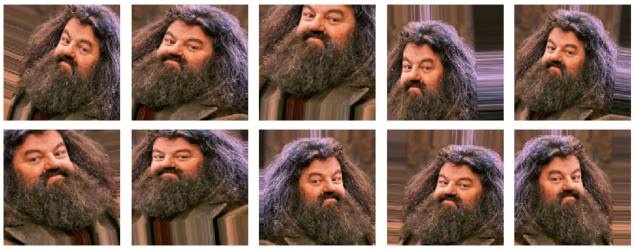

Figure 1: Data augmentation applied to an example image. Augmented every image to 10 using this.

4 Methods

4.1 HagridNet Architecture

Whereas typical face recognition systems use an end-to-end recognition network, we use an ensemble network architecture to achieve high accuracy face recognition by combining existing face recognition network with a specialized network. In our case, this specialized network is a fine-tuned version of

FaceNet which recognizes bearded faces at a higher accuracy. In order to decide which of the two recognition network to use, a dispatcher network recognizes the presence or absence of a beard with a sigmoid output, which is used to determine the recognition network.

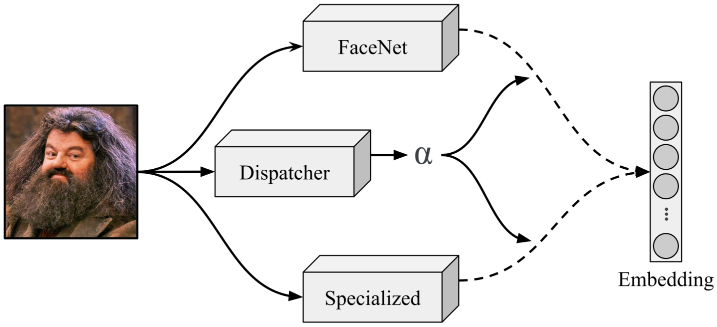

Figure 2: The HagridNet architecture. The dispatcher output α represents the probability of a beard being present.

Like many other deep learning based face recognition systems, FaceNet operates by outputting an embedding representing the identity of the input image. Specifically, FaceNet outputs a 128-dimensional vector. Then, the identity of two faces is determined by the distance between their respective embeddings, determined by the norm of their difference:

$$\text{Similarity} = ||f_1 - f_2||_2$$

Where embeddings f_1, f_2 correspond to the network outputs of the two faces being compared. Since both FaceNet and the specialized network necessarily have somewhat different embeddings, the same recognition network is used for both images. Specifically, if either face has a beard, the specialized network is used. Otherwise, the regular FaceNet is used.

4.2 Dispatcher Network: Beard Detector

The dispatcher network determines which network is used for facial recognition. In our case, the dispatcher network must recognize the presence of a beard in an image. We present a few methods for implementing this network.

4.2.1 Support Vector Machine

The Support Vector Machines (SVM) learning algorithm is among the best supervised learning algorithms for discriminative classification. SVMs are capable of operating in very high-dimensional spaces using kernelization, a method by which to apply SVMs to very high or even infinite dimensional feature spaces. Since our images are 224x224 in size with 3 color channels, the dimension of each data point is $224 \cdot 224 \cdot 3 = 150,528$ dimensional. Running an SVM on this high dimensional of a feature set would not compute in a reasonable time. Instead, we applied kernelization with the Nystroem approximation to reduce our data to 3000 dimensions. The result SVM had an accuracy of 0.53, only slightly better than guessing.

4.2.2 Logistic Regression

Logistic regression models the problem of beard classification as a linear function over the input image, bounded to $[0, 1]$ by the sigmoid function. That is, given an input image x, we model the probability of the image representing a bearded face by the function

$$P(y = 1) = h_\theta(x) = \sigma(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

which gives us the gradient ascent update rule

$$\theta_j := \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

Since this is a very simple approach which is not expected to work very well with high-dimensional data, we change the images to grayscale to reduce the number of color channels and thus reduce the complexity in our data.

4.2.3 Convolutional Neural Network With Transfer Learning

Convolution neural networks (ConvNets) have demonstrated outstanding performance in computer vision tasks, with the capability to detect edges and high-level features much better than fully-connected networks[7]. We use a model performing high-accuracy face attribute detection[4] based on the MobileNetV2 architecture[8], with pre-trained weights obtained online[5]. To customize this network for beard identification, we remove the last two layers of the pre-trained network and add two fully-connected layers with a sigmoid output. Then, we train the new layers on our custom dataset. We do this because using transfer learning with a pre-trained model generally gives faster and better results while requiring much less data[6].

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Figure 3: The MobileNetV2 architecture we use for beard recognition[8]

5 Experiments/Results/Discussion

5.1 Beard Identification

Method	Test Set Accuracy
SVM	0.53
Logistic Regression	0.70
ConvNet with transfer learning	0.984

Logistic regression was trained with a learning rate of 0.0001 over 100 epochs. We found that higher learning rates gave worse results, while lower learning rates were slower to converge. We paused training after 100 epochs after seeing diminishing training accuracy gains. Our convolutional network

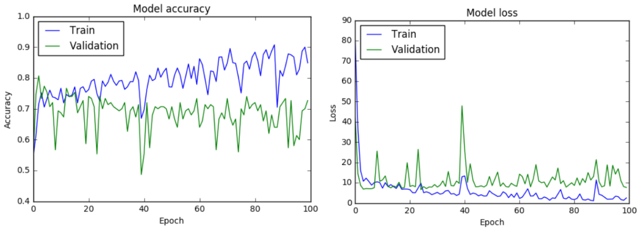

Figure 4: Accuracy and loss training logistic regression

with transfer learning was trained using the Adam optimization algorithm with learning rate 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$. The learning rate was smaller than the default for the same reasons as mentioned above, while we left β_1, β_2 at their default values. The mini-batch size of 64, which was the most we could do without memory allocation issues, and we trained over 50 epochs until we saw little improvement in accuracy and loss.

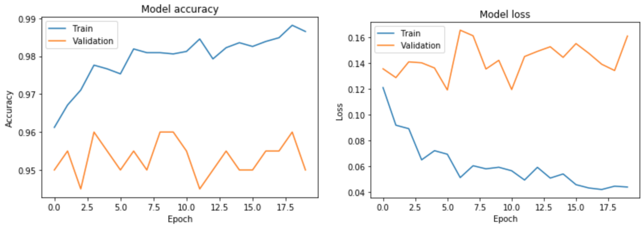

Figure 5: Accuracy and loss training the deep learning model

5.2 Face Recognition

For better face recognition accuracy on bearded faces, we created a specialized network by fine-tuning a pre-trained FaceNet model[11][12] on our beard dataset. We trained the model using 52 of the 60 identities in our custom dataset, for a total of close to 1600 images. To make sure that our specialized network did not overfit to the specific identities of our dataset, we set aside 250 pictures with identities that were not used to train the specialized network. Testing on these images gives the validation rate for the beards dataset.

We trained FaceNet using triplet loss. The loss function is given below:

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

In the table below, the Area Under the Curve (AUC) is a performance metric that represents how well the model is capable of distinguishing between true and false classes. The Equal Error Rate (EER) measures the false positive and false negative rate of the model. The lower the equal error rate value, the better the accuracy of the model.

Model	Dataset	Accuracy	Validation rate	AUC	EER
Baseline	LFW	0.98517±0.00652	0.92500±0.02372	0.998	0.015
Baseline	Beards	0.94286±0.02020	0.15417±0.05853	0.986	0.070
Specialized	Beards	0.98286±0.01245	0.46572±0.09480	0.997	0.013

Figure 6: Model accuracies across datasets. We observe that the baseline model performs well on a standard face dataset, but poorly on beards. Our specialized network performs much better on beards.

6 Conclusion/Future Work

We demonstrated that our network architecture achieves superior performance to FaceNet in the case of facial obstruction by beards. This architecture can be extended to any number of facial obstructions using a softmax output from the dispatcher network to trigger one of any number of specialized networks. We believe that using this method, it is possible to achieve strict improvements over existing facial recognition systems. In the future, it would be exciting to explore such an application of our proposed architecture to determine how much of an improvement is achievable.

7 Contributions

7.1 Edward Vendrow

- Developed Deep learning model with transfer learning for beard identification
- Ran FaceNet and fine-tuned on our beard dataset with triplet loss
- Data preprocessing code
- Logistic regression for beard identification
- Support Vector Machine results
- Developed overall network architecture
- Network architecture diagram
- Data collection
- Wrote paper
- Helped with poster

7.2 Akash Singhal

- Collected and arranged over 90% of the dataset
- Helped counter variance problem with proper usage of data
- Helped with some code ideas
- Beard Detection Network Diagram
- Wrote paper
- Made poster

7.3 Amogh Dixit

- Logistic regression and SVM baselines for beard detection
- Helped with data collection and classification into bearded/non-bearded bins
- Helped with some code ideas
- Helped in paper writing and poster creation

References

Github: https://github.com/evendrow/cs229

[1] Tensorflow/Tensorflow. 2015. tensorflow, 2019. GitHub, https://github.com/tensorflow/tensorflow.

[2] Keras-Team/Keras. 2015. Keras, 2019. GitHub, https://github.com/keras-team/keras.

[3] Scikit-Learn/Scikit-Learn. 2010. scikit-learn, 2019. GitHub, https://github.com/scikit-learn/scikit-learn.

[4] Liu, Ziwei, et al. "Deep Learning Face Attributes in the Wild." ArXiv:1411.7766 [Cs], Sept. 2015. arXiv.org, http://arxiv.org/abs/1411.7766.

[5] Anzalone, Luca. Luca96/Face-Clustering. 2019. 2019. GitHub, https://github.com/Luca96/face-clustering.

[6] Yosinski, Jason, et al. "How Transferable Are Features in Deep Neural Networks?" ArXiv:1411.1792 [Cs], Nov. 2014. arXiv.org, http://arxiv.org/abs/1411.1792.

[7] Krizhevsky, Alex, et al. "ImageNet Classification with Deep Convolutional Neural Networks." Commun. ACM, vol. 60, no. 6, May 2017, pp. 84–90. ACM Digital Library, doi:10.1145/3065386.

[8] Sandler, Mark, et al. "MobileNetV2: Inverted Residuals and Linear Bottlenecks." ArXiv:1801.04381 [Cs], Mar. 2019. arXiv.org, http://arxiv.org/abs/1801.04381.

[9] Schroff, Florian, et al. "FaceNet: A Unified Embedding for Face Recognition and Clustering." 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015, pp. 815–23. arXiv.org, doi:10.1109/CVPR.2015.7298682.

[10] “No Beards Allowed: Exploring Bias in Facial Recognition AI.” IDEO Is a Global Design and Innovation Company., https://www.ideo.com/blog/no-beards-allowed-exploring-bias-in-facial-recognition-ai.

[11] Sandberg, David. Davidsandberg/Facenet. 2016. 2019. GitHub, https://github.com/davidsandberg/facenet.

[12] Cao, Qiong, et al. “VGGFace2: A Dataset for Recognising Faces across Pose and Age.” ArXiv:1710.08092 [Cs], May 2018. arXiv.org, http://arxiv.org/abs/1710.08092.

[13] Taigman, Yaniv et al. “DeepFace: Closing the Gap to Human-Level Performance in Face Verification.” 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014): 1701-1708.

[14] Sharif, Mahmood et al. “Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition.” CCS ’16 (2016).

[15] Milich, Andrew et. al. "Eluding Mass Surveillance: Adversarial Attacks on Facial Recognition Models." Stanford University CS229 Website (2018).