

# Face Recognition: A Shallow CNN Approach

Leonard Yulianus  
Indiana University  
Bloomington, IN, USA  
lyulianu@umail.iu.edu

Ramya Rao  
Indiana University  
Bloomington, IN, USA  
ramrao@umail.iu.edu

## Abstract

*This project investigates the problem of face recognition on a small dataset. We propose Wideception, a shallow but wide convolutional neural network architecture based on inception module by Szegedy et al. [1]. In our experiment, Wideception is able to provide a reasonable accuracy of about 47.5% with just 20 training images per class while keeping the network architecture simple. The main distinctions of Wideception are the hierarchical structure of the convolutional layers and the elimination of the fully-connected layer which not only reduces overfitting but also significantly reduces the number of parameters and training time of the network compared to a traditional convolutional neural network with same number of layers as Wideception.*

## 1. Introduction

Faces represent complex multidimensional meaningful visual stimuli and developing a computational model for face recognition is difficult [2]. Face recognition is a difficult task because face changes over time. People look different with different hairstyle, on different emotions, from different angles. Generally each person has eyes, ears, nose and mouth but there are fine details in these organs that shape and differentiate a person from others. Despite the difficulty, face recognition is very popular in Computer Vision and has a wide range of applications, such as surveillance and verification. Our project implements a face recognition system which identifies an image of a person into one of the known classes with respect to the training dataset. We use a convolutional neural network for classification as it has shown many good results and considered the current state-of-the-art technique in object recognition.

The main objective of our project is to design a convolutional neural network which performs notably better than random chance with very few training images. The motivation behind using very few training images is to apply our face recognition system on real-world applications such as suspect identification during crime scene investigation or searching for a

missing person given surveillance images. Since the privilege of obtaining a large number of images might not be plausible in such applications, we need to design a convolutional neural network architecture that leverages the limited dataset to the best extent. Our network is based on inception module [1] which is proposed by GoogLeNet, the winner of ILSCVR 2014 ImageNet classification challenge.

## 2. Related Work

There are many methods for object recognition in Computer Vision, convolutional neural network is only one of the methods. As we design our CNN-based face recognition system, we mention few very closely related works with respect to the inception module in this section.

The use of Gabor filters of different scales and orientations followed by max pooling showed better results with respect to the image classification task as this resonates more with the biological system as experimented by Serre et al. [3]. This provides basis for the work by Szegedy et al. [1] who use the above mentioned idea by using convolutional filters of different sizes to account of scale invariance in their architecture for image classification and detection and this was mimicked in our architecture.

The Network in Network paper by Lin et al. [4] propose a deep neural network with multiple micro neural networks. The generalised linear model of the convolutional filter is passed to a multilayer perceptron which is a non linear function approximator [5] to provide better capability to differentiate between classes which are linearly inseparable. This is used by Szegedy et al. [1] as a layer of 1x1 convolutions which have dual purpose of dimensionality reduction and decreased width and depth of the network, with no penalty on the accuracy, which is also used in our architecture to reduce the number of parameters in the network.

## 3. Methods

### 3.1. Dataset

The dataset used in our experiment is taken from “Celebrities on the Web” [6], which contains 202,792 images from

1,583 people. Out of this dataset, we selected 8 people: Barack Obama, Brad Pitt, Cristiano Ronaldo, David Beckham, Emma Watson, Hillary Clinton, Keira Knightley, and Scarlett Johanson as our main classes. For each class we collected 20 different images for training and another 20 images for testing.

We also picked 20 images from 20 different people as a negative class. The reason for including the negative class is to provide the flexibility for the network to identify a face that belongs to a person which is not of the main classes. Figure 1 demonstrates the variability in our dataset from each class.



Figure 1. Few images from each classes, the last two images belong to the negative class

### 3.2. Pre-processing

Since the original dataset is of varied size, we resized and cropped the images in our dataset into 128x128 pixel. We artificially expanded our training dataset during the training phase by randomly flipping the image horizontally, upscaling the image to 150x150 pixel, and randomly cropping 128x128 pixel patches out of the 150x150 pixel image. This dataset expansion was done automatically on each batch of training dataset by caffe [7], the framework used in our experiment.

We also experimented augmenting our dataset even further by applying random projective transformation on the training images, the effects of these transformations are elaborated in the subsequent sections. Figure 2 shows the results of projective transformation.



Figure 2. Random projective transformation, leftmost image is the original

### 3.3. Architectural Details

Our Wideception architecture is based on inception module (shown in Figure 3). Instead of concatenating the feature maps from each branch of convolution layers, we feed the feature maps into another sub inception modules. The feature

maps from all the sub inception modules are concatenated and fed into the classifier where we get a probability associated with each class of how confident the network is that the presented face belongs to a certain class. The network also includes dropout before concatenating the output of the four sub inception modules to reduce overfitting in our network. Figure 4 shows Wideception architecture. Table 1 has all the details related to the layers in our network.

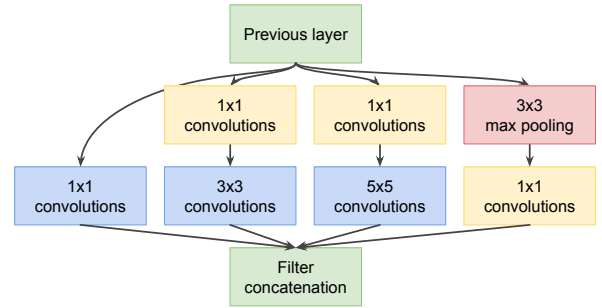


Figure 3. Original inception module with dimension reduction

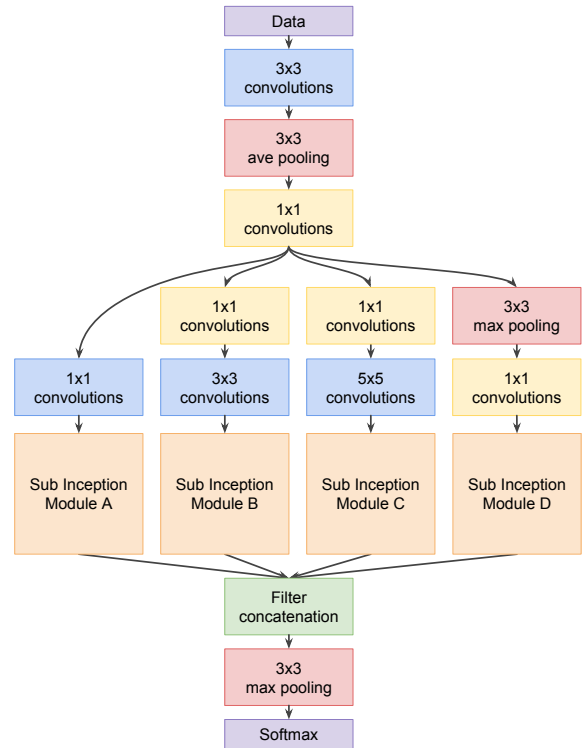


Figure 4. Wideception architecture

### 3.4. Training Methodology

We trained our network with stochastic gradient descent using caffe [7] on Big Red II. We used batch size of 10 images,

Layer	Size	# outs	Out size	# params
Conv 1	3x3/2	3	64x64	84
Ave pool	3x3/2	3	32x32	–
Conv 2	1x1/1	8	32x32	32
Main/Conv 1	1x1/1	4	32x32	36
Sub A/Conv 1	1x1/1	2	32x32	10
Sub A/Conv 2a	1x1/1	4	32x32	20
Sub A/Conv 2b	3x3/1	10	32x32	370
Sub A/Conv 3a	1x1/1	4	32x32	20
Sub A/Conv 3b	5x5/1	10	32x32	1010
Sub A/Max pool	3x3/1	4	32x32	–
Sub A/Conv 4	1x1/1	10	32x32	50
Sub A/Concat	–	32	32x32	–
Main/Conv 2a	1x1/1	4	32x32	36
Main/Conv 2b	3x3/1	4	32x32	148
Sub B/Conv 1	1x1/1	10	32x32	50
Sub B/Conv 2a	1x1/1	4	32x32	20
Sub B/Conv 2b	3x3/1	2	32x32	74
Sub B/Conv 3a	1x1/1	4	32x32	20
Sub B/Conv 3b	5x5/1	10	32x32	1010
Sub B/Max pool	3x3/1	4	32x32	–
Sub B/Conv 4	1x1/1	10	32x32	50
Sub B/Concat	–	32	32x32	–
Main/Conv 3a	1x1/1	4	32x32	36
Main/Conv 3b	5x5/1	4	32x32	404
Sub C/Conv 1	1x1/1	10	32x32	50
Sub C/Conv 2a	1x1/1	4	32x32	20
Sub C/Conv 2b	3x3/1	10	32x32	370
Sub C/Conv 3a	1x1/1	4	32x32	20
Sub C/Conv 3b	5x5/1	2	32x32	202
Sub C/Max pool	3x3/1	4	32x32	–
Sub C/Conv 4	1x1/1	10	32x32	50
Sub C/Concat	–	32	32x32	–
Main/Max pool	3x3/1	8	32x32	–
Main/Conv 4	1x1/1	4	32x32	36
Sub D/Conv 1	1x1/1	8	32x32	40
Sub D/Conv 2a	1x1/1	4	32x32	20
Sub D/Conv 2b	3x3/1	8	32x32	296
Sub D/Conv 3a	1x1/1	4	32x32	20
Sub D/Conv 3b	5x5/1	8	32x32	808
Sub D/Max pool	3x3/1	4	32x32	–
Sub D/Conv 4	1x1/1	8	32x32	40
Sub D/Concat	–	32	32x32	–
Main/Concat	–	128	32x32	–
Max pool	3x3/2	128	16x16	–
Softmax	–	9	–	–

Table 1. Wideception architecture detail

learning rate of 0.0001, with momentum of 0.9. The learning rate was decreased by 10% every 2,000 iterations. We trained our network for 50,000 iterations and saved the trained model

every 10,000 iterations for evaluation.

## 4. Results

Along with our network structure, we also experimented with Eigenfaces and AlexNet. For comparison, We included AlexNet as CNN counterpart and Eigenfaces as traditional method.

### 4.1. Wideception

We trained our network twice, first using only random flipping and cropping on the training dataset. Secondly, we trained on dataset which are augmented by random projective transformation. Figure 5 shows the accuracy for the first experiment. Figure 6 shows the accuracy for the second experiment.

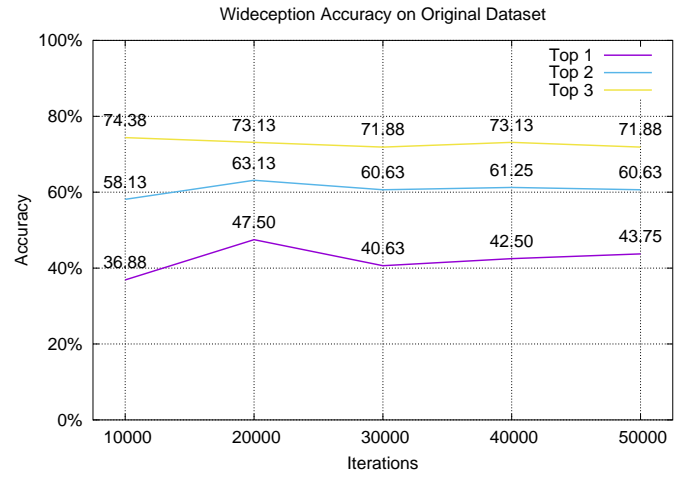


Figure 5. Accuracy on Original Dataset

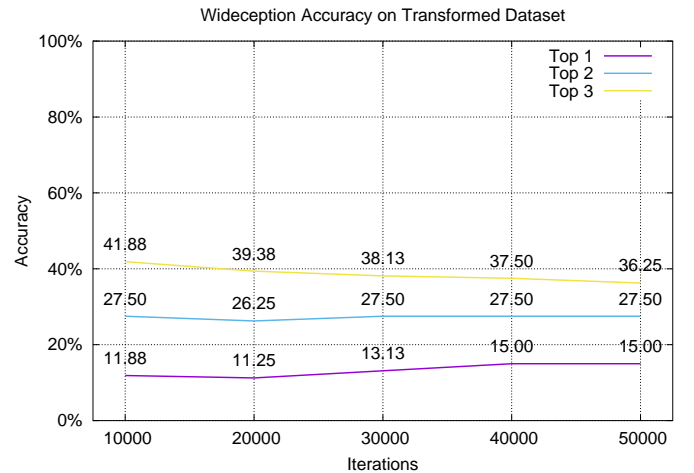


Figure 6. Accuracy on Transformed Dataset

Table 2 shows the confusion matrix for our network using the model from iterations 20,000.

	BO	BP	CR	DB	EW	HC	KK	SJ	Others
Barack Obama	8	2	2	2		2	1		3
Brad Pitt	2	9				7		1	1
Cristiano Ronaldo	2	1	12	2		2		1	
David Beckham	1		3	11		2		1	2
Emma Watson	1				4		8	3	4
Hillary Clinton	2					14		3	1
Keira Knightley		2		1		1	10	4	2
Scarlett Johansson				1		5	1	8	5
Others									

Table 2. Confusion matrix for Wideception (iterations = 20,000)













Class	Correct – Top 1		Correct – Top 3		Incorrect	
Emma Watson						
	Emma: 0.90	Emma: 0.65	Keira: 0.63 Emma: 0.29 Others: 0.03	Keira: 0.24 Obama: 0.22 Emma: 0.19	Others: 0.86 Keira: 0.07 Crist: 0.03	Others: 0.66 David: 0.13 Hillary: 0.09
Hillary Clinton						
	Hillary: 0.90	Hillary: 0.90	Scarlett: 0.86 Hillary: 0.10 David: 0.01	Obama: 0.48 Hillary: 0.32 David: 0.12	Others: 0.55 Crist: 0.38 Obama: 0.03	

Figure 7. Classification result example

## 4.2. Eigenfaces

We resized our training images to a resolution of 20x20 and 50x50. After extracting the mean face of all the training images, we performed Eigen decomposition and obtained Eigen values and Eigen vectors. We selected K values of the most significant Eigen values and used the corresponding Eigen vectors to extract features from the training set and passed it to an SVM (Support Vector Machine). We used the same Eigen vector obtained during the training and multiplied resized (20x20) version of the test images to SVM and obtained the accuracy.

The plot of Accuracy vs. sensitivity to parameter K, with training images re-sized to 20x20 is shown in Figure 8. The values in the x-axis are different values chosen for K and y-axis shows the accuracy calculated at that K value.

The plot of Accuracy vs. sensitivity to parameter K, with training images re-sized to 50x50 is shown in Figure 9. The values in the x-axis are different values chosen for K and y-axis shows the accuracy calculated at that K value

## 4.3. Fine-tuning AlexNet

We did not train AlexNet from scratch because of the number of parameters it has. Instead we fine-tuned AlexNet by fixing the convolutional layers and only allowing it to learn the weights for the fully-connected layers. Figure 10 shows the accuracy of AlexNet.

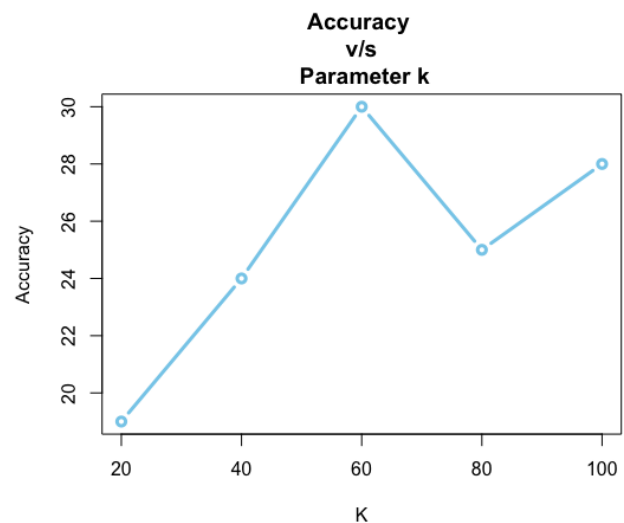


Figure 8. Eigenfaces: accuracy vs. Parameter K for Image size 20x20

## 4.4. Comparison

We wanted to know how our model compares to Eigenfaces and AlexNet. Table 2 shows the comparison for tested models.

Model	Top 1 Accuracy
Eigenfaces (20x20)	30.00%
Eigenfaces (50x50)	32.00%
AlexNet	49.38%
Wideception (transformed data)	15.00%
Wideception (original data)	47.50%

Table 3. Accuracy between models

## 5. Discussion

It is interesting that based on figure 5 and figure 6, it seems that instead of increasing the accuracy, the artificial data created from projective transformation seems to confuse the net-

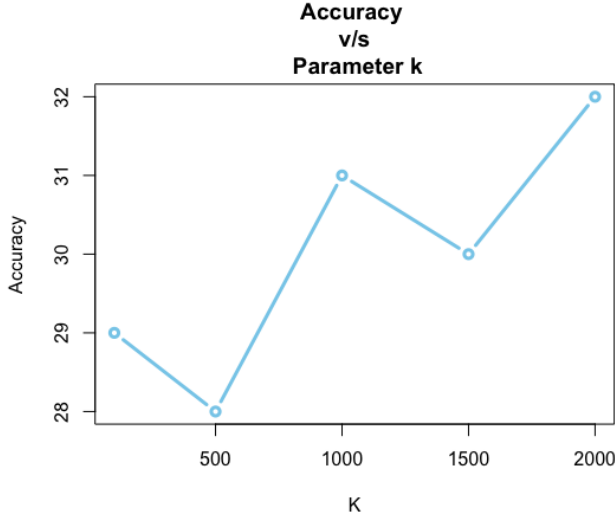


Figure 9. Eigenfaces: accuracy vs. Parameter K for Image size 50x50

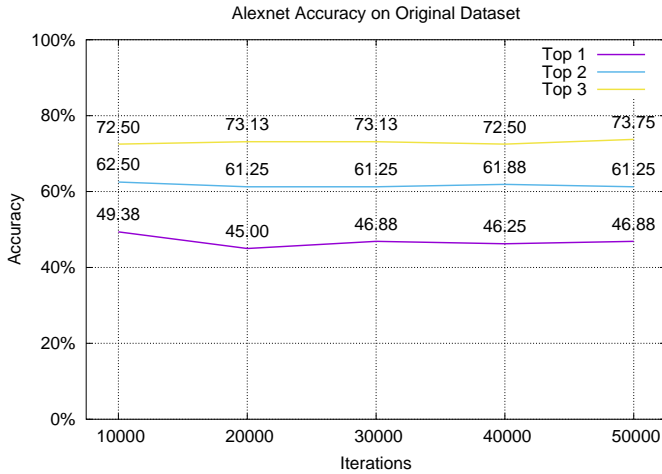


Figure 10. Fine-tuning AlexNet on our dataset

work. We suspect that the transformation might have introduced bias to the network.

By comparing Wideception to one of the traditional methods Eigenfaces, we observe that Eigenfaces is very sensitive to the number of Eigen values that are chosen during the training phase, while our network (Wideception) does not have any such magic parameters. To illustrate the sensitivity of the parameter K, we collected accuracy for different numbers of Eigen vectors and plotted them, which are shown in figure 8 and figure 9. It is clear from the plots that we have to vary the number of Eigen vectors chosen to find the optimal value for each training set. In figure 8, the accuracy is maximum for  $K = 60$ , as  $K$  increases the accuracy is changing but there is no straight forward relation between accuracy and  $K$ , such as directly proportional or inversely proportional. The  $K$  values

also varies with change in the size of the image which is shown in figure 9. We also note that the accuracy from Eigenfaces, with a maximum of 32% with image size 50x50, is about 15 points lower than our network.

Comparing Wideception to AlexNet, our model achieved slightly less accuracy than AlexNet. If we inspect further, there are few things that can be observed. Our network uses way less parameters than AlexNet and might have less degree of freedom to model the input data. Furthermore, AlexNet has already learned “good” filters from other datasets, and we only need to tune the fully-connected layers.

## 6. Conclusion

Data augmentation should be performed carefully. As it is shown in our experiment, data augmentation leads to worse performance.

Shallow convolutional neural network might be a viable alternative in a limited training dataset setting.

Our network architecture, Wideception, gives reasonable results with an accuracy of 47.5%, slightly less than fine-tuned AlexNet model.

## 7. Acknowledgment

This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute, and in part by the Indiana METACyt Initiative. The Indiana METACyt Initiative at IU was also supported in part by Lilly Endowment, Inc.

## References

- [1] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
- [2] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.
- [3] Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomaso Poggio. Robust object recognition with cortex-like mechanisms. *IEEE transactions on pattern analysis and machine intelligence*, 29(3), 2007.
- [4] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.
- [5] Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, DTIC Document, 1961.
- [6] Xiao Zhang, Lei Zhang, Xin-Jing Wang, and Heung-Yeung Shum. Finding celebrities in billions of web images. *IEEE Transactions on Multimedia*, 14(4):995–1007, 2012.

- [7] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.