Face Recognition using Deep Learning and TensorFlow framework

Mohamed Elsayed - Omar Fathi
Karima Asad - Hend Mohamed - Marwa Mohamed

May 10, 2017

Abstract

This paper introduced a simple **C**onvolution **N**eural **N**etwork (**CNN**) using TensorFlow framework for image classification by using AT&T and Yale faces datasets.

1 Introduction

CNN is the best way to simulate the human brain, as we can build it by (neurons) which make the network and each neuron can think or do mathematical operations to produce right output or at least close to it, so we need to build a good network to produce best output in our image classification.

2 Problem

The problem is How can we make the machine recognize faces of different people in different challenges

- ⋆ **Input:** Images of different people and their labels.
- ⋆ **Output:** Name of each person on his image.

3 Motivation

There are many algorithms and projects that could make face recognition, but they have problems and they have some of the challenges that mentioned in (4), So we are motivated to make different type of algorithm that can solve these problems, we want to make the machine can detect many faces correctly regardless the problems in the image, and this will be useful for camera to detect more faces regardless illumination or other challenges, it also will be useful for Facebook face recognition that can be useful in detect persons face in the posts and it can offers help for police that it can help them in catching criminals.

4 Challenges

Since this task of recognizing a visual concept is relatively trivial for a human to perform, it is worth considering the challenges involved from the perspective of a Computer Vision algorithm, there are list of challenges below:

1. **Viewpoint variation:** A single instance of an object can be oriented in many ways with respect to the camera.

2. **Scale variation:** Visual classes often exhibit variation in their size (size in the real world, not only in terms of their extent in the image).

3. **Deformation:** Many objects of interest are not rigid bodies and can be deformed in extreme ways.

4. **Occlusion:** The objects of interest can be occluded. Sometimes only a small portion of an object (as little as few pixels) could be visible.

5. **Illumination conditions:** The effects of illumination are drastic on the pixel level.

6. **Background clutter:** The objects of interest may blend into their environment, making them hard to identify.

7. **Intra-class variation:** The classes of interest can often be relatively broad, such as chair. There are many different types of these objects, each with their own appearance.

5 Related Works

1. Deep Face Recognition [1]

2. Learning face recognition from limited training data using deep neural networks [2]

3. Deep Learning Face Representation by Joint Identification-Verification [3]

4. Deep Learning Face Representation from Predicting 10,000 Classes [4]

6 Algorithm

As we learned in neural network or machine learning **"You have input data and its output and you need to produce the program that fit this data"** so it depend on our dataset, the architecture and its hyper parameters for our network so we can assume that the architecture is **"The algorithm"**.
We choose deep learning because it is the best method for now in classification (face recognition).
Deep learning is like a child who learn from his previous mistakes and give best answer (output) based on these mistakes, this is the idea for our decision we need to build a child in our machine to learn what we need to teach for it :).

7 Alternatives

1. **PCA(principal components analysis):** This method is designed to model linear variation. Its goal is to find a set of mutually orthogonal basis functions that capture the directions of maximum variance in the data and for which the coefficients are pairwise decorrelated.
   PCA is guaranteed to discover the dimensionality of the manifold and produces a compact represen-tation.
   PCA was used to describe face images in terms of a set of basis functions, or eigenfaces.
   PCA is an unsupervised technique, so the method does not rely on class information. In our implementation of eigenfaces, we use the nearest neighbor (NN) approach to classify our test vectors using the Euclidean distance.

2. **MPCA:** One extension of PCA is that of applying PCA to tensors or multilinear arrays which results in a method known as multilinear principal components analysis (MPCA).
   Since a face image is most naturally a multilinear array, meaning that there are two dimensions describing the location of each pixel in a face image, the idea is to determine a mulitlinear projection for the image, instead of forming a one-Dimensional (1D) vector from the face image and finding a linear projection for the vector. It is thought that the multilinear projection will better capture the correlation between neighborhood pixels that is otherwise lost in forming a 1D vector from the image.

3. **LDA:** Fisherfaces is the direct use of (Fisher) linear discriminant analysis (LDA) to face recognition.
   LDA searches for the projection axes on which the data points of different classes are far from each other while requiring data points of the same class to be close to each other.
   Unlike PCA which encodes information in an orthogonal linear space, LDA encodes discriminating information in a linearly separable space using bases that are not necessarily orthogonal.
   It is generally believed that algorithms based on

LDA are superior to those based on PCA. When the training data set is small, PCA can outperform LDA, and also that PCA is less sensitive to different training data sets.

4. **ICA:** is a generalization of PCA and it tries to identify high-order statistical relationships between pixels to form a better set of basis vectors. In a similar fashion to PCA and LDA, once the new basis vectors are found, the training and test data are projected into the subspace and a method such as NN is used for classification.

8 Dataset

1. AT&T dataset dataset which contains 40 classes, 10 images for each class and we take one image from each class for testing.
   - Each image take from different position (Viewpoint variation).
   - Some characters wear glasses (Occlusion).
2. Yalefaces dataset which contains 15 classes, 11 images per class and we take one image for testing
   - Each image take from different position (Viewpoint variation).
   - Some characters wear glasses (Occlusion).
   - Some images token with different Illumination.

9 Architecture

Our architecture contains

1. Two convolution Layers
   - First convolution contains 32 filter with 5x5 filter size.
   - Second convolution contains 64 filter with 5x5 filter size.
2. Flatten Layer
   - Convert convolution layer to one dimensional array for fully connected layers.
3. Three Fully connected layers
   - First fully connected contains 2000 neuron (class).
   - Second layer contains 750 neuron (class).
   - Last layer contains 40 neuron (class) for AT&T Faces and 15 for Yale Faces.
4. we use maxPooling with filter 2x2 for AT&T Faces and 5x5 for Yale Faces
5. Learning Rate is equal to 0.001 with ADAM optimizer gradient descent
6. Dropout = 0.75
7. BatchSize = 5

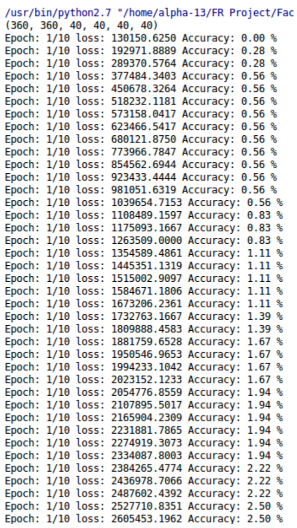
10 Training & Testing

⋆ After training AT&T Faces and Yale Faces we get these results.

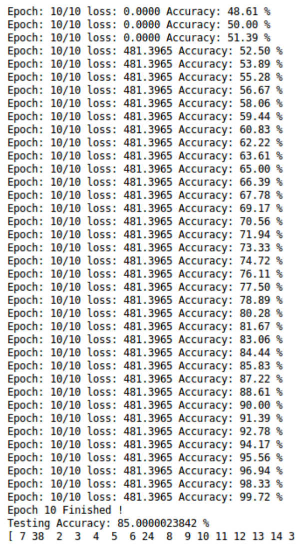
Dataset	N Epoch	Train Accuracy	Test Accuracy
AT&T	10	99.72	85.00%
Yale	15	92.90	80.00%

Table 1: Training Results

11 Output

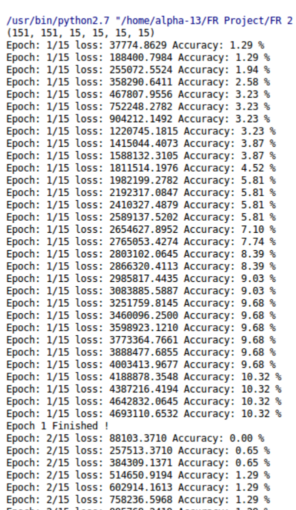


(a)

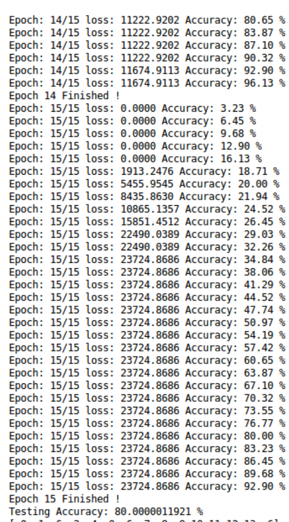


(b)

Figure 1: (a) Start Training (b) End Training



(a)



(b)

Figure 2: (a) Start Training (b) End Training

Figure 3: Testing AT&T dataset using TensorFlow

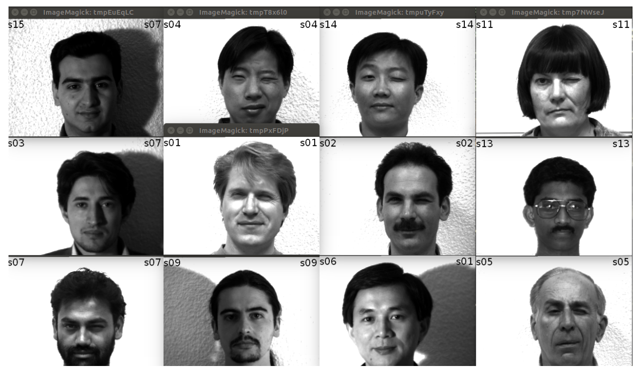


Figure 4: Testing Yale dataset using TensorFlow

12 Analysis

- It works fine because our network structure and hyper parameters are perfect.
- We can make it better in the future Insha'Allah.

13 Contribution Division

1. **Omar:** Build dictionaries that have weights and Biases to define our network, convert images from the convolution layer to the hidden layer as a one dimension array.
2. **Karima:** Build functions that make convolution, maxpool operations and function for convolution layer.
3. **Marwa:** Build a function that make fully connected layer that return the result of multiply weights with layer and add biases.

4. **Hend:** Build a function that make the convolution neural network (architecture) using previous functions.

5. **Mohammed:** Produce accuracy and loss while training and show result.

References

[1] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[2] Xi Peng, Nalini Ratha, and Sharathchandra Pankanti. Learning face recognition from limited training data using deep neural networks. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 1442–1447. IEEE, 2016.

[3] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014.

[4] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.