Face Recognition using Landmark Estimation and Convolution Neural Network

Neelam Patel[1], Namrata Sharma[2]

[1]PG Student, Dept. of Computer Science and Engineering, Sushila Devi Bansal College of Engineering, Indore
[2]Asst. Professor, Dept. of Information Technology, Sushila Devi Bansal College of Engineering, Indore

---***---

Abstract - *Face recognition is a particular situation in which objects are recognized. It has gained special attention in recent years due to a broad range of applications such as robot-human interaction, gesture control, surveillance, security, and monitoring of people. We tried to make it usable in real time and we used the LFW dataset to test the system, which is a set of random images available for a person on the internet. Completely different from normal datasets collected under controlled circumstances. Additionally, because CNN needs a lot of information, we have used a big dataset; we have accomplished an increase in both training and testing. We checked the official LFW information page showing the outcomes of nearly all techniques and only a few techniques crossed the 90% mark. Up to 95 percent of the maximum precision we have attained is seen, but it can be time consuming compared to others. There is still a great deal of room for enhancement and it is to be achieved using various algorithm permutations.*

Key Words: Face Recognition, Convolutional Neural Network, Landmark Estimation, LFW Dataset, Recognition Accuracy

1. INTRODUCTION

1.1 Motivation

Our main motive for this project was our interest in implementing distinct algorithms and techniques of pre-processing to fix important issues with facial recognition issue that has huge and meaningful uses of many domains. There are countless and substantial applications of this studies, including facial recognition, facial disfigurement medical diagnosis and even a person's monitoring, surveillance and mobile apps. We were significantly influenced by the notion that we were working to solve an open and crucial issue with all these possible apps. [1]

We specifically chose the Facial Key Point Detection Kaggle technique, which enabled us to test a broad range of methods and neural net models to fix the location issue which is otherwise simple. Recognition of facial main points is a difficult issue given differences in both faces and photo situations. [2] Facial characteristics vary by size, place and expression, while the illumination and visual angle vary with the picture circumstances. In conjunction with the need for very precise coordinated projections (e.g. the precise corner of an eye), these abundant differences lead us to think that the job is excellent to work on. [3]

CNN and HOG detectors have been developed primarily in the form of pythons for the formation of convolutional neural networks.[4][5] The real-time python interpreter program has been used for testing face detection, overlap removal, detection, facial normalization and face recognition over many months, as distinct modules and as a consistent scheme. OpenCV was developed for computer effectiveness and focused on real-time apps. A large number of structures and functions have been used to manipulate the matrix and efficiently process images in the OpenCV library. The library of Intel Performance Primitive (IPP) has been licensed for the acceleration of the image in real time.

2. Tools Used

2.1 OpenCV

(Open Source Computer Vision Library) is released under a BSD license and hence it's free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. [6]

Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics. [7]

2.2 TensorFlow

TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well. [8]

2.3 Python V3.7

Python is a great object-oriented, interpreted, and interactive programming language. It is often compared (favorably of course) to Lisp, Tcl, Perl, Ruby, C#, Visual Basic, Visual Fox Pro, Scheme or Java... but it's actually simple and easy to work on. [9]

Python combines remarkable power with very clear syntax. It has modules, classes, exceptions, very high level dynamic data types, and dynamic typing. There are interfaces to many system calls and libraries, as well as to various windowing systems. New built-in modules are easily written in C or C++ (or other languages, depending on the chosen implementation). Python is also usable as an extension language for applications written in other languages that need easy-to-use scripting or automation interfaces. [10]

3. Proposed System

The general architecture of any recognition system involves these basic steps:

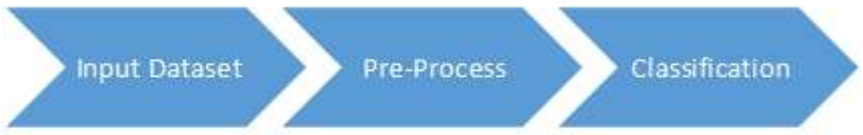

Fig-1: General Architecture of Recognition System

3.1 Method 1: Using HOG and SVM

In our system we started with basic recognition system and used LFW dataset because it is equivalent to real time data and is not taken in constrained condition. Most basic dimensionality reduction technique PCA is being used to reduce calculation and operation time.

Input Dataset: LFW

Pre-Process: Eigen-Face (PCA)

Classifier: Support Vector Machine

A strong classifier SVM is being used to get higher recognition accuracy. Results are being presented in next section..

3.2 Method 2: Further Improvement using CNN

Because of the big variation between face poses, the intuition behind this is; face recognition is hard to tackle. If the CNN were to receive the pictures as they are, it would have to cope with individuals in all ways, background noise, distinct facial positions in the picture, etc. Therefore, first attempting to decrease this variability by centering faces in the picture makes sense.

Therefore, the first stage would be to have all faces in the center of the picture and turn them to look at the front. Doing so makes analyzing them much easier. In our scheme, we made use of the landmark estimation for face frontalization as; it supplied a reliable application that worked quickly enough to meet our needs.

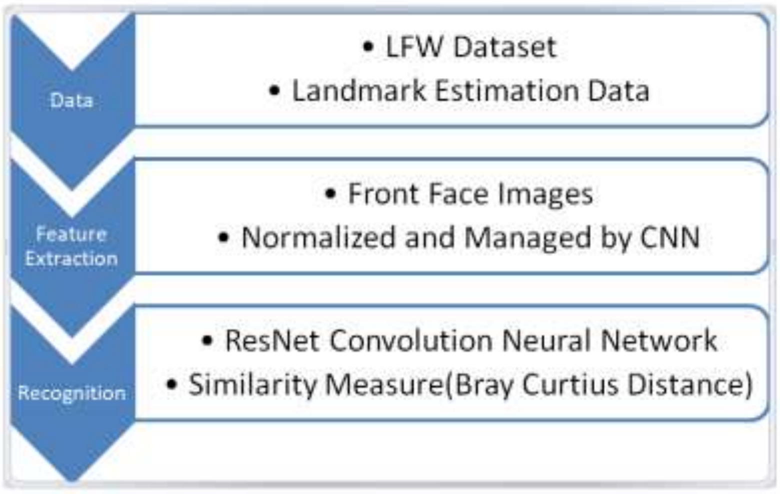
Data
• LFW Dataset
• Landmark Estimation Data
Feature Extraction
• Front Face Images
• Normalized and Managed by CNN
Recognition
• ResNet Convolution Neural Network
• Similarity Measure(Bray Curtius Distance)

Fig-2: Proposed Architecture

4. Result Analysis

We have applied sklearn pipeline and preprocessing with MinMaxScaler. Training is done with a fully connected layer and 100 hidden units with tensorflow backend.

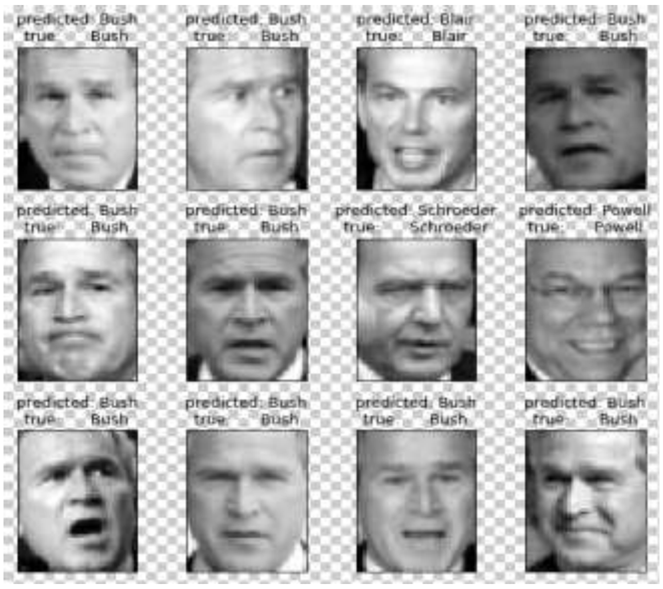
predicted: Bush true: Bush
predicted: Bush true: Bush
predicted: Blair true: Blair
predicted: Bush true: Bush
predicted: Bush true: Bush
predicted: Bush true: Bush
predicted: Schroeder true: Schroeder
predicted: Powell true: Powell
predicted: Bush true: Bush
predicted: Bush true: Bush
predicted: Bush true: Bush
predicted: Bush true: Bush

Fig-3: Image from LFW dataset correctly recognized method I

The above results show that the people in dataset are predicted successfully. Few options are difficult to identify and in turn reduces the overall accuracy. The system performance is shown in table below. The overall accuracy of the system was 85%.

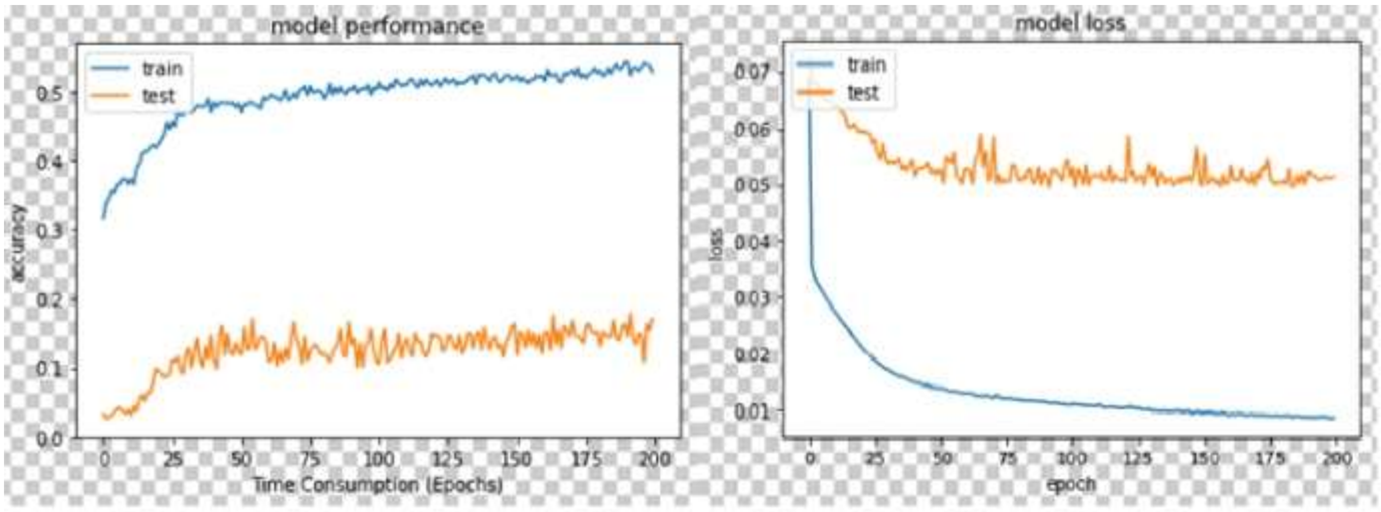

Fig-4 Model Performances and loss in Detection and Recognition

We further applied augmentation; usual techniques of information increase are altering the illumination, rotating, scaling, or image translation. We've only been interested in the first one from these. The other two, as we are already locating, centralizing and resizing the faces, created no sense in our situation. On the contrary, even with the standardization step applied to the feature vector, we felt it might be interesting to use modifications in illumination to train our CNN better.

The data provides keypoints as shown on the image below.

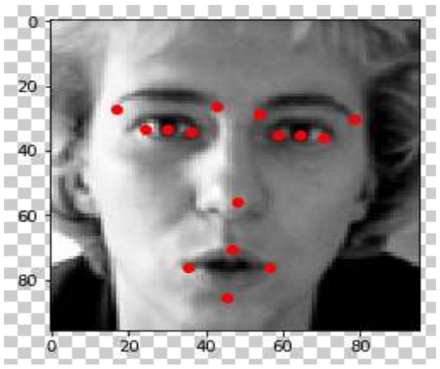

Fig-5: Keypoint Detected

Table-1: Performance in terms of different chosen Parameters

Data	Dropout rate	Distance Measure	Accuracy
Original	0.7	Euclidean	0.814
Original	0.5	Bray Curtis	0.923
Augmented	0.5	Euclidean	0.924
Original	0.5	Bray Curtis	97.48

With regard to the CNN itself, we attempted to modify the setup of their layer and two dropout rates which are 0.5 and 0.7. We can compare their performances. It was evident in this situation that the initial CNN setup was better conducted. We believed we could avoid overfitting this way, given that we had less information to train, but empirical evidence showed that it did not yield any advantage. Experiments and results hand, as they behaved differently with each CNN setup, we did not reach any conclusion about the highest dropout rate.

Table-2: Accuracy of various methods on LFW Dataset

S.No	Methods	Accuracy
1	NReLU[12]	0.8073 ± 0.0134
2	Single LE + holistic[14]	0.8122 ± 0.0053
3	LBP + CSML, aligned[15]	0.8557 ± 0.0052
4	CSML + SVM, aligned[15]	0.8800 ± 0.0037
5	Our Model I(Eigenface + SVM)	0.8467 ± 0.026
6	Our Model II (Landmark Estimation +CNN)	0.9748 ± 0.0042

The overall accuracy in keypoint detection is 93.3%. CNN gives accurate results when we removed the not available (na) data. When we applied this outputs to recognize faces with Bray Curtis distance for similarity measure the overall system accuracy achieved was 97.4%

We have checked the LFW data official website which shows the results of almost all methods and only few methods have crossed the mark of 90%. The maximum accuracy is seen up to 95% we have achieved better accuracy but it may be time consuming compared to others.

To better know where we stand on the graphs on the base of performance, comparative analysis of our suggested model with modern techniques. Comparative analysis of our proposed model with contemporary methods, to better understand where we are standing on the charts on basis of performances.

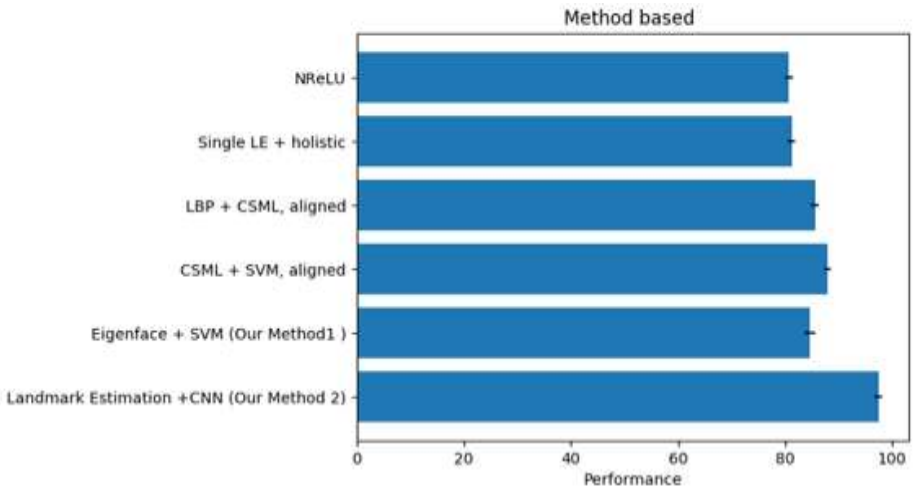

Fig-6: Performance graph of our methods with others

5. Conclusion and Future Work

5.1 Conclusion

The system acquired was widely tested and various combinations of parameters were tested. In the Wild Huang et al., 2007 dataset, we used the Large Face to evaluate the efficiency of our face verification step.

In addition, we have used large dataset because of CNN it requires lot of data; we performed augmentation in both training and testing. The facial recognition portion was tested using two distinct sizes of data sets and we achieved constant outcomes of around 90% precision, achieving a maximum of 85%. These findings are better than we anticipated, allowing for some instances of real-life use. There is space for enhancement; there is still room for improvement. There are many parameters in our scheme related to the earlier point that we were not able to experiment as widely as we wished. Although we attempted to create trials as comprehensive as possible, owing to time limitations we were unable to do so. As such, some parameters still exist that we do not fully comprehend their behavior.

5.2 Future Work

To be sure of the face recognition system works at complete potential, by means of further testing we need to discover the ideal parameter setup. Finally, as has already been said, with regard to the further experiments are possible. We intend to continue working on them so that more characteristics are provided by the internet instrument. And, on the other side, various individuals will have to be recognized at once by the video identification portion. After that, the FR scheme can be used in many other apps, so there is still a lot of job to do with this project.

REFERENCES

[1]Turk, Matthew; Pentland, Alex: Eigenfaces for recognition. In: Journal of cognitive neuroscience 3 (1991), Nr. 1, S. 71–86.

[2] Alex Krizhevsky, Geoffrey E H. Ilya Sutskever S. Ilya Sutskever:ImageNet Classification with Deep Convolutional Neural Networks. (2012). http://dx.doi.org/{kriz,ilya,hinton}@cs.utoronto.ca

[3] Yang, M-H ; Kriegman, D J. ; Ahuja, N: Detecting faces in images: A survey. In: IEEE Transactions on pattern analysis and machine intelligence 24 (2002), Nr. 1, S. 34–58

[4] Taigman, Yaniv ; Yang, Ming ; Ranzato, Marc'Aurelio ; Wolf, Lior:Deepface: Closing the gap to human-level performance in face verification.In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, S. 1701–1708

[5] Fei-Fei Ly, Justin J. Andrej Karpathy K. Andrej Karpathy: CS231n: Convolutional Neural Networks for Visual Recognition. urlhttp:// cs231n.stanford.edu/, 2016

[6] Alex Krizhevsky, Geoffrey E H. Ilya Sutskever S. Ilya Sutskever: ImageNet Classification with Deep Convolutional Neural Networks. (2012). http://dx.doi.org/{kriz,ilya,hinton}@cs.utoronto.ca. – DOI kriz,ilya,hinton@cs.utoronto.ca

[7] Simonyan, Karen ; Zisserman, Andrew: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: ICLR (2015). http://dx. doi.org/{karen,az}@robots.ox.ac.uk. – DOI karen,az@robots.ox.ac.uk

[8]https://www.google.com/search?q=snapchat+features&rlz=1C1CHBD_enIN800IN800&tbm=isch&
source=iu&ictx=1&fir=W5kEw1ET6_zktM%253A%252C3nxhz23VnbhO3M%252C_&vet=1&usg=AI4_kReKxi1EHiGU_lN9kMaA
WWbQvzNrg&sa=X&ved=2ahUKEwjDsPKlvqLjAhUm6nMBHV5gCZ8Q9QEwAHoECAQQAw#imgrc=W5kEw1ET6_zktM

[9] http://www.kscst.iisc.ernet.in/spp/39_series/SPP39S/02_Exhibition_Projects/184_39S_BE_1465.pdf

[10] https://www.bayometric.com/face-recognition-applications/