# Neural Network For Recognizing Human Faces


Sangeetha Verkot
*Computer and Mathematical Sciences Department*
*Lewis University*
Illinois, USA
sangeethamverkot@lewisu.edu

Brian Dutremble
*Computer and Mathematical Sciences Department*
*Lewis University*
Illinois, USA
brianrdutremble@lewisu.edu

Meena Makan
*Computer and Mathematical Sciences Department*
*Lewis University*
Illinois, USA
meenamakan@lewisu.edu

Justice Ike
*Computer and Mathematical Sciences Department*
*Lewis University*
Illinois, USA
justiceoike@lewisu.edu



***Abstract*— This paper provides an overview of a neural network-based face recognition system and the techniques in which machine learning is used to teach the network how to function optimally. The paper then describes the way in which that system functions, detailing the processes involved in its three main steps: facial detection, feature evaluation, and facial recognition. The system uses technologies available in the Open Source Computer Vision (OpenCV) library and methodology developed in Python to implement them. We then present the results of our attempts to use the Harr Feature-based Cascade Classifier versus those using the Caffe deep learning framework.**



***Keywords*— Face detection, Face Recognition, Neural Networks**


## I. Introduction

Many systems like video surveillance, biometrics, and human-computer interface rely on reliable and successful face detection and recognition. Face detection is not straightforward because facial features vary from person to person. The race, gender, age, and other physical characteristics of an individual must be considered, which presents a challenge for computer vision. The objective of the project is to investigate methods to recognize and identify faces. The function of face detection is to identify a face in an image, while face recognition makes the decision regarding whose face it is using an image database. In this project, we investigate a dataset consisting of faces and compare the Haar Feature-based Cascade Classifier with neural network techniques utilizing the Caffe deep learning framework. Three tasks for face recognition will be applied: Acquisition – the detection and tracking of face-like images, Feature Extraction – the segmentation, alignment and normalization of the face images, and Recognition – the identification of a facial expressions. This paper outlines some of the research and algorithms used in face detection and recognition, then details the typical processes in face recognition. The research section details the types of techniques, neural network algorithms, and statistical methods used in face recognition, and the idea creativity and complexity section looks at techniques in OpenCV and the Caffe model. In the idea feasibility section, we look at challenges of face recognition and how neural networks can alleviate these challenges. Lastly, we present our python code, dataset and results.

## II. Related Work in the field of facial recognition

A. Study of face Recognition Studies by Pritpal Singh [2] reviewed various methods and techniques used in face recognition research studies. Major techniques used are Template matching, Neural Networks, Fischer Face, Geometrical feature matching. Minor techniques used are Principal Component Analysis (PCA) and Multilayer Perception (MLP). The steps in general face recognition are 1). Face detection, 2). Features Extraction and 3).Face recognition.

B. Face Detection and Recognition using Viola-Jones algorithm and Fusion of PCA and Artificial Neural Network (ANN) by Narayan Deshpande and Dr. S. Ravishankar [1] has a two stage approach. The first stage detects the human face in an image using the Viola-Jones algorithm. The second stage uses a fusion of Principle Component Analysis and Feed Forward Neural Network for face recognition. Their proposed approach provided a 94 percent accuracy in image identification.

C. Facial Expression Recognition Using Weighted Mixture Deep Neural Network Based on Double-Channel Facial Images by Yang, Caoi, Ni, and Zhangi [6] experimented with changes in facial expression recognition (FER) using a weighted mixture deep neural network which processes grayscale and their corresponding LBP images. The expressions angry, disgust, fear, happiness, sadness, and surprise were evaluated. The recognition accuracies of the different facial expressions were above 90%.

D. Fusion of Local Binary Patten (LBP) and Histogram of Oriented Gradients (HOG) using Multiple Kernel Learning (MKL) for Infrared Face Recognition by Zhihua Xie, Peng Jiang, Shuai Zhang [5], proposed a fusion algorithm of LBP and HOG for infrared face recognition. The LBP operator is adopted first to extract the texture feature of an infrared face, then the edge features of the original infrared face are extracted by using the HOG operator. Finally, MKL is applied to fuse the texture features and edge features. With the MKL fusion, a 98.6 percent recognition rate was achieved.

E. Face Detection and Face Recognition using Open Computer Vision Classifiers by Lahiru Dinalankara [4] showed that using Haar-cascades for face detection worked extremely well even when subjects wore spectacles. Real time video speed was satisfactory as well devoid of noticeable frame lag. Considering all factors, Local Binary Pattern Histogram (LBPH) combined with Haar-cascades can be implemented as a cost effective face recognition platform.

F. FaceNet: A Unified Embedding for Face Recognition and Clustering by Schroff, Kalenichenko, and Philbin from Google Inc. [3], research a method based on learning a Euclidean embedding per image using a deep convolutional network. They achieved accuracy levels of 99.63 percent. A system called FaceNet was used. This system learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. Upon this space being produced, the face recognition, verification, and clustering tasks were implemented using standard techniques with FaceNet embeddings as feature vectors.

## III. BACKGROUND

Facial recognition is a widely researched area in computer science. Attempts were made as early as the 1970s, starting with a feature vector of human faces, followed by Principal Component Analysis (PCA) for feature extraction. Using PCA, Eigenface was developed in 1991 by Mathew Turk and Alex Pentland and is considered a major milestone in the technology. Later in 1994, Local Binary Pattern Histogram (LBPH) analysis was introduced for facial recognition. In 1996, the Fisherface method was developed; it uses Linear discriminant analysis (LDA) for dimensional reduction and that can identify faces in different illumination conditions. In 2001, Viola and Jones introduced a face detection technique that utilizes Haar cascades and ADABoost (their detection framework selects a small number of critical visual features from a larger set and yields efficient classifiers). In 2007, Naruniec and Skarbek developed a face recognition technique using Gabor Jets. In this project, we focus on neural network method using deep learning with the Open Computer Vision (OpenCV) library.

## IV. METHODOLOGY

A person is recognized by his or her distinct features like the eyes, nose, cheeks, and forehead, and the relationships between those features. Different facial expressions change these facial feature relationships and provide measurable differences with which the OpenCV-based system can be taught. OpenCV is an existing library developed by Intel in 1999 consisting of programming functions mainly aimed at real-time computer vision. The OpenCV release 3.4 has a trainer and face recognizer class for face recognition. Using this library, together with OpenCV deep neural networks module (DNN), we prepare the training data using a collection of images. The face detection, training, and face recognition processes then follow. A general face recognition system includes the steps: face detection; feature extraction; and face recognition.

### A. Face Detection

Face detection is a computer technology that is based on learning algorithms to locate human faces in digital images. Face detection takes images/video sequences as input and locates face areas within these images. This is done by separating face areas from non-face background regions. Facial feature extraction locates important feature (eyes, mouth, nose, and eyebrows) positions within a detected face. Feature extraction simplifies face region normalization where detected face aligned to coordinate framework to reduce the large variances introduced by different face scales and poses.

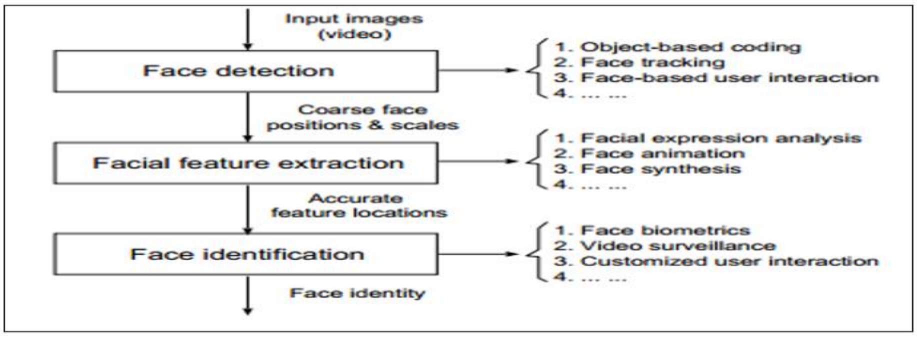


Figure 1. Framework of a Face recognition system [8]

The main steps of face detection system are shown in Figure 2. Some systems detect and locate faces at the same time, while others first perform a detection routine and then, if positive, try to locate the face. Some tracking algorithms may also be needed. Face detection separates image windows into two parts: one containing faces, and one containing the background. The process is difficult because while commonalities exist between faces, they vary in terms of age, skin color and expression. Variances in lighting, image quality, and geometry can also create problems. Image pre-processing is sometimes necessary to adapt the input image to the algorithm prerequisites. Once the images have been properly prepared, some algorithms analyze the image as it is, while others try to extract certain relevant facial regions.

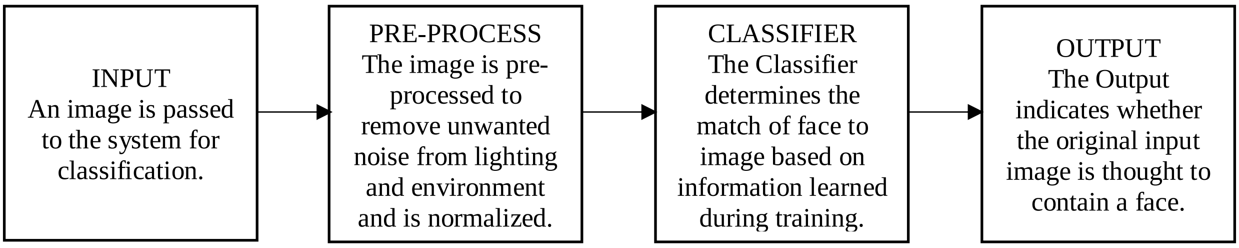


Figure 2. A General Face Detection System [11]

## B. Feature Extraction

We can typically recognize a person we know, even when a person is wearing glasses or has grown a beard, but these features represent a challenge to the computers. Feature extraction's main task is to extract information from photographs. It involves the procedure of extracting essential information from a face image, which is used for a later step of identifying the subject with an acceptable error rate. This process must be efficient in terms of computing time and memory usage. The typical steps of feature extraction are: dimensionality reduction, feature extraction, and feature selection. These steps may overlap. Dimensionality reduction can be a consequence of the feature extraction and selection algorithms, and is an essential task in any pattern recognition system. The performance of a classifier depends on the amount of sample images, number of features, and classifier complexity. A false positive ratio of a classifier does not increase as the number of features increases. However, added features may degrade the performance of a classification algorithm. This may happen when the number of training samples is small relative to the number the features.

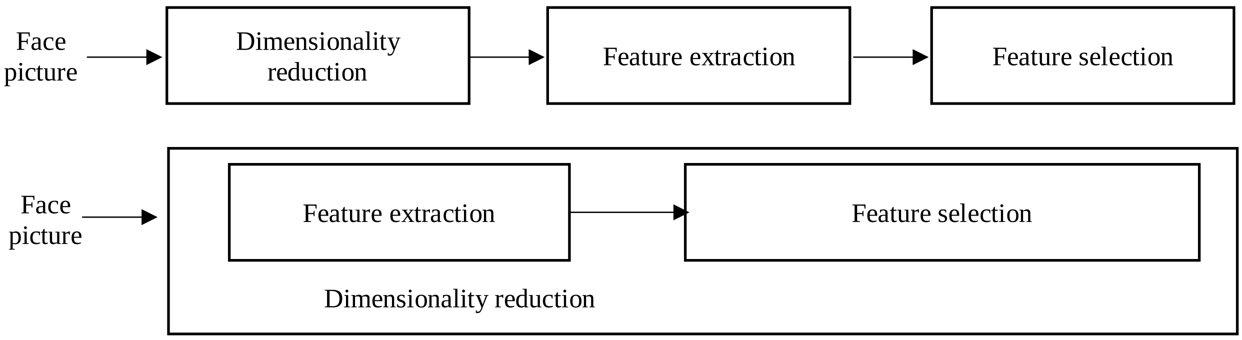

Figure 3. Feature Extraction Process

### C. Feature Classification

Once the features are extracted and selected, the next step is to classify the image. Classifiers have a large impact on the success of face recognition. Classification algorithms usually involve some learning - supervised, unsupervised or semi-supervised. Unsupervised learning is the most difficult approach, as there are no tagged examples. Many face recognition applications do include a tagged set of subjects and are thus considered to be utilizing supervised learning methods. Where systems have a small dataset, or the acquisition of new tagged samples is infeasible, semi-supervised learning is required. Three concepts are key in building a classifier - similarity, probability and decision boundaries.

## V. RESEARCH

Some of the main face detection approaches are:

- The knowledge-based method uses ruled-based methods that encode our knowledge of human faces.
- The featured-based method uses the features of faces for detecting texture and skin color. Results generated by such algorithms can be corrupted due to illumination, noise, and occlusion.
- The template matching method uses algorithms that compare input images with stored patterns of faces and features. The performance of this method may be affected by variations in scale, pose, and shape.
- The appearance-based method is a template matching method whose pattern database is learnt from a set of training images. These templates are predefined by experts. Statistical analysis and machine learning techniques can be used to find the relevant characteristics of face and non-face images.

Neural Network approaches for face detection and recognition include Neural networks with Gabor filters, with Hidden Markov Models and Fuzzy neural networks. Some of these methods use neural networks just for classification. One approach is to use decision-based neural networks and there are methods that perform feature extraction using neural networks. We provide a synopsis of the research conducted using ANN below.

Lawrence et. al used self-organizing map neural network and convolutional networks. Self-organizing maps (SOM) are used to project the data in a lower dimensional space and a convolutional neural network (CNN) for partial translation and deformation invariance. Overall CNN classification methods are not optimal in terms of computational time and complexity as their classification performance is limited and is costlier to implement in practice [11]. Neural networks with Gabor filters - Bhuiyan et al. (2007) proposed a neural network method combined with Gabor filter. Their algorithm uses a multilayer perceptron with back-propagation algorithm. Followed by a pre-processing step, each image is normalized in terms of contrast and illumination. The noise is reduced by a "fuzzily skewed" filter that works by applying fuzzy membership to the neighbor pixels of the target pixel. The median value is used as the 1 value membership, and reduces the extreme values, taking advantage from median filter and average filter. Then, each image is processed through a Gabor filter. The filter is represented as a complex sinusoidal signal modulated by a Gaussian kernel function. Although the algorithms main purpose is to face illumination variations, it shows a useful neural network application for face recognition. [11]. Neural networks and Hidden Markov Models - Hidden Markov Models (HMM) is a statistical tool used in face recognition. Fuzzy neural networks – In 2009, Bhattacharjee et al. developed a face recognition system using a fuzzy multilayer perceptron (MLP) [9]. The idea behind this approach is to capture decision surfaces in non-linear manifolds, a task that a simple MLP can hardly complete. [11].

In a typical pattern classification in artificial neural networks, the aim is to approximate a mapping $f: \mathcal{R}^{l\,x\,w} \rightarrow \mathcal{L}$ where $\mathcal{L}$ is a set of class labels, *l* and *w* are integers. The object $\mathcal{R}$, to be classified is represented by an *l* x *w* array of uniform type which corresponds to cells in a *l* x *w* grid. This can be applied to classification of images where an object is an *l* x *w* array of which the elements are the intensity of pixels in an image[11].

Statistical approach for recognition algorithms include PCA, Discrete Cosine Transform (DCT-II), Linear Discriminant Analysis, Locality Preserving Projections (LPP), Gabor Wavelet, Independent Component Analysis (ICA), Kernel PCA, Eigenface and Eigenvalues, Euclidean Distance and other algorithms namely genetic algorithms, Bayesian networks, bi-dimensional regression methods, ensembled based and boosting methods. Images of faces are represented as high-dimensional pixel arrays. In statistical approach, the image is represented in terms of d features, hence it is viewed as a vector in a d-dimensional space.

## VI. IDEA CREATIVITY AND COMPLEXITY

Artificial neural networks (ANN) have often been used in the field of image processing and pattern recognition. Face detection is an essential component of face recognition systems because its ability to focus computational resources on the part of an image containing a face is the basis of any algorithm's total efficiency. The process of face detection in images is complex because of variability present across human faces such as: pose, expression, position and orientation, skin color, presence of glasses or facial hair, differences in camera gain, lighting conditions and image resolution. Therefore, it is important to choose a suitable face classification technique that can reliably differentiate between the faces different persons and other information. For our project, we have experimented with OpenCV and deep learning.

The OpenCV has more than 2500 optimized algorithms that can be used to detect and recognize faces. The most common algorithms used in OpenCV are the Haar Cascade Classifier for face detection and the Eigenface and Fisherface algorithms for face recognition. The library has a deep learning (DNN) module which uses Caffe models. Caffe is a deep learning framework made with expression, speed, and modularity in mind. It is developed by Berkeley AI Research (BAIR) and by community contributors [13].

Deep Learning is a subfield of machine learning that focuses on algorithms based on artificial neural networks. Deep networks use a cascade of multiple layers of non-linear processing units for feature extraction and transformation. They are compositional models that are represented as a collection of interconnected layers that work on chunks of data, mimicking the human brain's neurons. Deep learning can learn in different ways, including supervised (e.g. classification) and unsupervised (e.g. pattern analysis) methods, and in both of those methods it learns multiple levels of representation that correspond to different levels of abstraction. The Caffe model defines a net layer-by-layer in its own model schema. The network defines the entire model bottom-to-top from input data to its output. As data flow through the network in forward and backward passes, Caffe stores, communicates, and manipulates the information as what are called blobs. The blob is the standard array and unified memory interface for the framework[13].

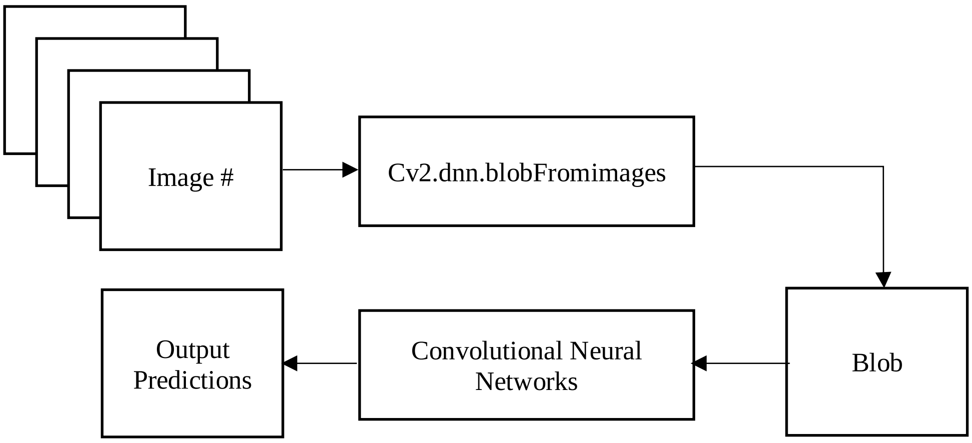


Figure 4. OpenCV Deep Learning (dnn)

A Caffe layer takes input through bottom connections and makes output through top connections. Each layer type defines three critical computations: setup, forward, and backward. The setup initializes the layer and its connections which is done once. The Forward action computes the output by given input from bottom and send to the top. The Backward action computes the gradient with respect to the input by the given gradient with respect to the top output and send to the bottom. A layer with parameters computes the gradient with respect to its parameters and stores internally. Layers have two key responsibilities for the operation of the network: a *forward pass* that takes the inputs and produces the outputs, and a *backward pass* that takes the gradient with respect to the output and computes the gradients with respect to the parameters and to the inputs, which are in turn back-propagated to earlier layers. These passes are simply the composition of each layer's forward and backward[13].

The Python code is detailed in section VIII with explanation of the steps followed. The experiment shown concerns human face recognition. We have used an image shown in section IX and the results of recognition in section X.[14]

## VII. IDEA FEASIBILITY

Face recognition has several challenges to overcome. Pose variation and illumination are the two main problems faced by face recognition researchers. Illumination presents difficulties because many algorithms rely on color information to recognize faces and differences in lighting can impact which colors are registered. Pose variation can create many issues, as most face recognition methods are designed to work best with frontal face images. The most successful types of approaches to these solving these problems are either multi-image-based approaches, which require multiple images for training using templates of all the possible pose and lighting variations, or single-model based approaches which solve the problems by using only one image at recognition and several examples of a subject during training. Geometric approaches try to build a sub-layer of pose-invariant information of faces. The input images are transformed depending on geometric measures on those models. Other problems that researchers have encountered in face recognition include: occlusion (where some parts of the face are missing), different formats of the input images (when different cameras or storage media are used), different facial expressions, and the effectiveness of a recognition algorithm (its hit ratio, error rate, computational speed, and memory usage)[15].

Using deep learning with convolutional neural networks (CNN), studies have shown that it is capable of handling facial images that contain occlusions, poses, facial expression and varying illumination levels. Experimental results show that using neural networks with a CNN solution can achieve 99.5 percent recognition. CNN are made up of neurons that have learnable weights and biases. The network ultimately expresses a single score function: from the raw image pixels on one end to class scores at the other. CNNs are specifically designed to identify faces in images without requiring much pre-processing[16].

## VIII. PYTHON PROGRAM

```
In [1]: import cv2
```

```
In [2]: import numpy as np
        import argparse
        face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
        img = cv2.imread('group.jpg')
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
In [3]: faces = face_cascade.detectMultiScale(gray, 1.3, 5)
        for (x,y,w,h) in faces:
            cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
            roi_gray = gray[y:y+h, x:x+w]
            roi_color = img[y:y+h, x:x+w]
        cv2.imshow('image_haar',img)
        cv2.waitKey(0)
        cv2.destroyAllWindows()
```

Figure 5. Python Code for Face Detection Using Haar Cascade Classifier

### Face detection using Load Caffe Framework Models

**Import the necessary packages**

```
In [ ]: import numpy as np
        import cv2
```

**Load our serialized model from disk. It takes two arguments prototxt,caffeModel**

```
In [ ]: print("[INFO] loading model...")
        net = cv2.dnn.readNetFromCaffe("deploy.prototxt.txt","res10_300x300_ssd_iter_140000.caffemodel")
```

**Load the input image and construct an input blob for the image by resizing to a fixed 300x300 pixels and then normalizing it**

```
In [ ]: image = cv2.imread("group.jpg")
        (h, w) = image.shape[:2]
        blob = cv2.dnn.blobFromImage(cv2.resize(image, (300, 300)), 1.0,(300, 300), (104.0, 177.0, 123.0))
```

**Pass the blob through the network and obtain the detections and predictions**

```
In [ ]: print("[INFO] computing object detections...")
        net.setInput(blob)
        detections = net.forward()

        # loop over the detections
        for i in range(0, detections.shape[2]):
            # extract the confidence (i.e., probability) associated with the
            # prediction
            confidence = detections[0, 0, i, 2]
            # filter out weak detections by ensuring the `confidence` is
            # greater than the minimum confidence. showing only faces with more than 50% confidence
            if confidence > 0.5:
                # compute the (x, y)-coordinates of the bounding box for the
                # object
                box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
                (startX, startY, endX, endY) = box.astype("int")

                # draw the bounding box of the face along with the associated
                # probability
                text = "{:.2f}%".format(confidence * 100)
                y = startY - 10 if startY - 10 > 10 else startY + 10
                cv2.rectangle(image, (startX, startY), (endX, endY),
                    (0, 0, 255), 2)
                cv2.putText(image, text, (startX, y),
                cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)

        # show the output image
        cv2.imshow("Output", image)
        cv2.waitKey(0)
```

Figure 6. Python Code for Face Detection Using Caffe Model

## IX. IMAGE DATASET

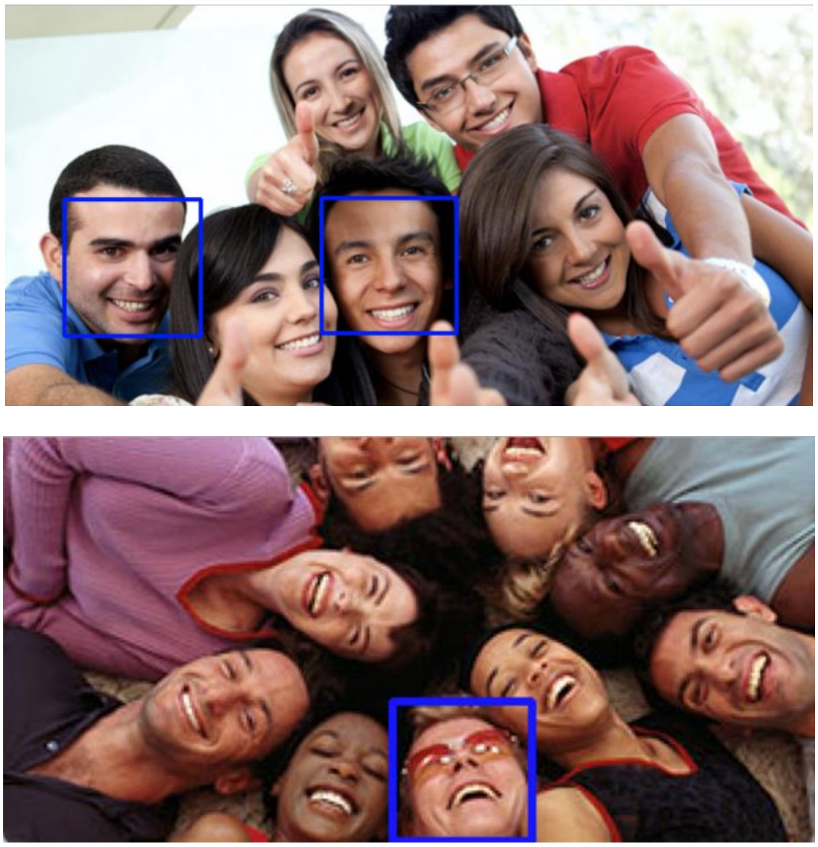

Figure 7. Face Detection Using Haar Cascade


Figure 8. Face Detection and Confidence of Caffe Model

## X. RESULTS

The model was trained to identify a part of an image as a face if it was more than 50 percent confident that it was a face. The DNN model could detect many more faces in the provided images, and did particularly well with the first of the two images. In the first of the two images, the Haar Cascade (Figure 7) showed a match of two faces as compared to Caffe (Figure 8), which identified all faces with recognition levels from 61.7 percent to 100 percent. The second image quality is lower, but the Haar cascade still performed worse as it could match just one face while the Caffe model recognized three faces with confidence levels ranging from 25.8 percent to 97.2 percent.

## XI. CONCLUSION

The face detection model based on deep neural networks, Caffe, consistently outperformed the machine learning based model. The likely reason for this difference in performance is that the Caffe model has multiple hidden layers between the input and the output layers, and each layer of nodes trains on a distinct set of features based on the previous layer’s output. Thus, the deeper into the neural net the processing goes, the more complex the features the nodes can recognize, since they aggregate and recombine features from the previous layer. Such a model likely mimics the process by which the human brain navigates face recognition.

## ACKNOWLEDGMENTS

This project would not have been possible without the support and knowledge of our professor, Dr. Piotr Szczurek, Assistant Professor Math/Computer Science, Lewis University. We would like to thank Dr. Szczurek for leading us through our study of artificial intelligence and for providing guidance as we worked on this project.

We would also like to thank Lewis University for providing the environment and resources necessary for this work, as well as our classmates for enriching that environment with their own important and inspiring work.

Lastly, we are grateful to Intel for creating the Open Source Computer Vision library and to Itseez for maintaining that library. Without OpenCV and its contained algorithms this project would have not been possible.

# REFERENCES


[1] N. T. Deshpande, Dr S. Ravishankar, “Face Detection and Recognition using Viola-Jones algorithm and Fusion of PCA and ANN”, ISSN 0973-6107 Volume 10, Number 5 (2017) pp. 1173-1189.

[2] P. Singh, “Study of Face Recognition Techniques”, vol 04, issue 04, International Research Journal of Engineering and Technology (IRJET), India, Punjabi University Guru Kashi Campus, Apr -2017.

[3] F. Schroff, D. Kalenichenko and J.Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering”, Google Inc., arXiz:1503.03832v3 [cs.CV] 17 June 2015.

[4] L Dinalankara, “Face Detection and Face Recognition using Open Computer Vision Classifiers. Plymouth University, Robotic Visual Perception and Autonomy Faculty of Science and Engineering, August 2017

[5] Z. Xie, P. Jiang, S. Zhang, “Fusion of LBP and HOG Using Multiple Kernel Learning for Infrared Face Recognition,” IEEE 16997372, China, Jiangxi Sciences and Technology Normal University, 29 June 2017.

[6] B. Yang, J. Cao, R. NI, Y Zhang, “Facial Expression Recognition Using Weighted Mixture Deep Neural Network Based on Double-Channel Facial Images,” IEEE 10.1109/ACCESS.2017.2784096, Changzhou, China, 15 December 2017.

[7] P. Viola , M. Jones , “Rapid object detection using a boosted cascade of simple features”, IEEE 990517, May 2004

[8] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman , “Eigenfaces vs. Fisherfaces: recognition using class specific linear projection” , IEEE 598228

[9] Nikolaos Stekas, Dirk van den Heuvel, “Face Recognition Using Local Binary Patterns Histograms (LBPH) on an FPGA-Based System on Chip (SoC)”, IEEE 7529883

[10] F. Zuo (2006) Embedded Face Recognition Using Cascaded Structures, Thesis, Technische Universiteit Eindhoven, China.

[11] O. N. A. AL-Allaf, “Review of Face Detection Systems based Artificial Neural Networks Algorithms”, The International Journal of Multimedia & Its Applications (IJMA) Vol.6, No.1, February 2014

[12] J. Donahue∗, Y. Jia∗, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, “DeCAF:ADeepConvolutionalActivationFeature forGenericVisualRecognition”, arXiv:1310.1531v1 [cs.CV] 6 Oct 2013 , UC Berkeley & ICSI, Berkeley, CA, USA

[13] J. Yangqinge, ‘Caffe’, 2014. [Online]. Available: http://caffe.berkeleyvision.org/tutorial/net_layer_blob.html [Accessed: 4- Mar- 2018].

[14] A. Rosebrock, ‘Face detection with OpenCV and deep learning’ [Online]. Available: https://www.pyimagesearch.com/2018/02/26/face-detection-with-opencv-and-deep-learning/. [Accessed:4-Mar-2018]

[15] R. Gross, S. Baker, I. Matthews, T. Kanade, “Face Recognition Across Pose and Illumination”, 2004. Carnegie Mellon University Handbook of Face Recognition.

[16] LeCun, Yann. "LeNet-5, convolutional neural networks". Online. Available: http://yann.lecun.com/exdb/lenet/ Retrieved 16 November 2013.