**“HOME SECURITY USING FACE RECOGNITION TECHNOLOGY”**

BY

G.SHARAN (17311A19F1)

V.SAI MADHAV (17311A19H0)

G.SHRIPADH RAO (17311A19H6)

***ABSTRACT***

*Face is the easiest way to penetrate each other's personal identity. Face recognition is a personal identification system that uses a person's personal characteristics to identify a person's identity. The human facial recognition process basically consists of two stages, namely face detection, where this process occurs much faster in humans, except in situations where the object is located a short distance, the next.*

*Introductions, which recognize a face as individuals. The stage is then repeated and developed as a model for facial image recognition (facial recognition), one of the biometrics technology studied and developed by experts. There are two types of methods that are currently popular in the developed face recognition pattern and face recognition with deep learning.*

***Keywords--*** *Face recognition, Eigen face, deep learning.*

1. **INTRODUCTION**

Facial recognition is the act of identifying an already known object as a known or unknown face. The problem of facial recognition is often confused with the problem of face recognition. Face recognition, on the other hand, is to use a database of faces to validate this input face for the purpose of deciding whether a "face" belongs to an acquaintance or unknown person.

***1.1 FACE RECOGNITION***

***1.1.1 Neural Network***

Neural networks are gaining more attention in many pattern recognition problems such as OCR, object recognition, and autonomous robot driving. Since face detection can be considered as a two class pattern recognition problem, various neural network algorithms have been proposed.

The advantage of using neural networks for face detection is:

The feasibility of training a system to capture complex squared conditional densities of facial patterns. However, one drawback is that network architecture is massively tuned to achieve exceptional performance (number of layers, number of nodes, learning rate, etc.). In the early days the most hierarchical neural network was proposed by Agui et al. The first stage consisting of two parallel sub networks in which the input intensity values ​​from the original image are filtered.

The inputs to the second-stage network have outputs from the sub-network and extracted feature values. An output in the second stage shows the presence of a face in the input field. Prop and Samal developed one of the earliest neural networks for facial recognition. Their network consists of four layers with 1,024 input units, 256 units in the first hidden layer, eight units in the second hidden layer and two output units. Ferraud and Bernier presented a detection method using the auto associative neural network.

The idea is based on that showing an auto associative network with five layers capable of performing a non-linear principal component analysis. One auto associative network is used to detect frontal-view faces and the other is used to detect faces up to 60 degrees to the left and right of the frontal view.

Thereafter Lin et al. Presented a face recognition system using probabilistic decision-based neural networks (PDBNN). The architecture of PDBNN is similar to a radial basis function (RBF) network with modified learning rules and probabilistic interpretation.

1. **TECHNICAL OBJECTIVES**

***2.1 Working of Neural Network***

Deep learning systems are modeled after neural networks in the neocortex of the human brain, where high-level cognition occurs. In the brain, a neuron is a cell that transmits electrical or chemical information. When connected with other neurons, it forms a neural network. In machines, neurons are virtual - basically bits of code that run statistical registers. Enough string of these virtual neurons together and you get a virtual neural network.

While traditional machine learning algorithms are linear, intensive learning algorithms stack up in a hierarchy of increasing complexity and abstraction. To understand deep learning, imagine a child whose first word is a dog. The child learns what a dog is (and is not) pointing to objects and saying the word dog. The parents say, "Yes, he is a dog," or, "No, he is not a dog." As the child continues to point to objects, he or she becomes more aware of the features that all dogs have.

Without knowing what the child does, a complex abstraction (dog concept) is clarified by constructing a hierarchy in which each level of abstraction is created with knowledge that was derived from the preceding layer of the hierarchy .

Each algorithm in the hierarchy applies a nonlinear transformation to its input and learns to use it as an output to construct a statistical model. Iterations continue until the output reaches an acceptable accuracy level. The number of processing layers through which the data must pass, which induces a label depth.

***2.2 Face Recognition using deep learning***

Let's first look at how humans recognize faces. Facial perceptions are very complex because recognition of facial expressions encompasses wide and diverse areas in the brain. Brain imaging studies typically show a large proportion of activity in a region of the temporal lobe, known as the fusiform gyrus, an area that is also known to cause prosopagnosia when damaged (especially when damage on both sides. it occurs). People learn to recognize faces from birth and can clearly distinguish one person from another at around four months of age.



*Figure 2.2: Face Perceptions*

Let's first look at how humans recognize faces. Facial perceptions are very complex because recognition of facial expressions encompasses wide and diverse areas in the brain. Brain imaging studies typically show a great deal of activity in an area of the temporal lobe known as the temporal gyrus, an area that is also known to cause prosopagnosia when damaged. People learn to recognize faces from birth and can clearly distinguish one person from another at around four months of age.

The main thing a person pays attention to is the eyes, cheekbones, nose, mouth and eyebrows, as well as the texture and color of the skin. At the same time, our brain processes the face as a whole and is able to identify a person even up to half of the face. The brain compares the resulting picture with the internal average pattern and finds characteristic differences.

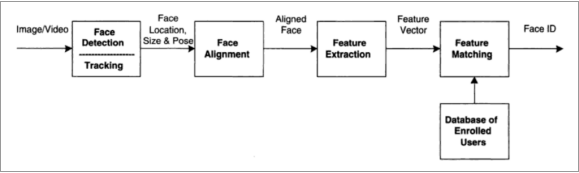
***2.3 Facial recognition system Working***

First, the facial recognition system needs to find a face in the image and illuminate the region. For this, the software can use a variety of algorithms: for example, proportionality and similarity of skin color, selection of shapes in the image and their comparison with facial contours, selection of symmetry using neural networks. The most effective is the Viola-Jones method, which can be used in real time. With this, the system recognizes faces even when rotated 30 degrees.

The method is based on Haar's signs, a set of black and white rectangular masks of various shapes. Masks are superimposed on different parts of the image, and the algorithm combines the brightness of all pixels of the image that are beneath the black and white parts of the mask and then calculates the difference between these values. Next, the system compares the results with the accumulated data and, determining the face in the image, keeps tracking it to select the optimal angle and image quality. For this purpose, motion vector prediction algorithms or correlation algorithms are used.

The most important measurements for facial recognition programs are the distance between the eyes, the width of the nose, the length of the nose, the height and size of the cheekbones, the width of the chin, the height of the forehead, and other parameters.

After that, the obtained data is compared with those available in the database, and, if the parameters match, the individual is identified.



*Figure 2.3: Face Recognition*

Face recognition is actually a sequence of several related steps:

1. First, you need to look at the image and find all the faces on it.

2. Secondly, it is necessary to focus on each face and determine that, despite the unnatural turn of the face or poor lighting, it is the same person.

3. Thirdly, it is necessary to highlight the unique characteristics of the face, which can be used to distinguish it from other people — for example, the size of the eyes, the elongation of the face, etc.

4. In conclusion, it is necessary to compare these unique characteristics of the face with the characteristics of other people you know to determine the name of the person.

The human brain does all this automatically and instantly. In fact, people recognize faces extremely well and, ultimately, see faces in everyday objects. Computers are incapable of such a high level of generalization (at least for the time being …), so you have to teach them every step in the process separately. It is necessary to build a pipeline on which you will find a solution at each step of the face recognition process separately and transfer the result of the current step to the next. In other words, you need to combine several machine learning algorithms into one chain:

• Find the face in the image

• Analyze the facial features

• Compare among the known faces

• Make the predictions

1. **Libraries**

***3.3 Dlib***

Dlib is a general purpose cross-platform software library written in the programming language C ++. Dlib C ++ is a toolkit for building real-world machine learning and data analysis applications. While the library is originally written in C ++, Python bindings are good, easy to use. . Its design is strongly influenced by design considerations by contract and component-based software engineering. Thus, it is, first and foremost, a set of independent software components. It is open-source software released under a boost software license.

In recent years, much of the development has been focused on building a comprehensive set of statistical machine learning tools and was published in the 2009 Dlib Journal of Machine Learning Research. Since then it has been used in a wide range of domains.

***For this Project we used Dlib’s Frontal Face Detector:***

The frontal face detector works really well in Dlib. It is simple and works just out of the box. This detector is based on a Histogram of Oriented Gradient (HOG) and Linear SVM. Although the HOG + SVM based face detector has been around for some time and has gathered a good amount of users, I'm not sure how many of us saw CNN (Contextual Neural Network) based face detector available in dlib.

You can use Dlib’s Frontal Face detected in 2 ways those are:

1. HOG + SVM

2. CNN

***3.3.1 Histogram Oriented Gradient (HOG):***

Size is one of the important parameters to mark any object. The HOG is an attribute descriptor that is used to mark objects based on their shapes. This technique calculates the histogram (incidence) of each gradient orientation in the mentioned part of the image. As we know that every image has color and intensity. Directional changes in these properties can be called gradient orientation. Below is a phased process of feature extraction which also includes HOG:

***1. Extracting spatial features of the image:***

After resizing the image, we stillpreserve all the features of image. Hence we resize the image, with the help of this, our code will run a bit more fast with no loss of information. This can be done with of open cv function cv2.resize().

***2. Image Colorspace Conversion:***

Converting the image to a particular colorspace to extract information from the chosen color channel:

We can convert image to any of the below color spaces: RGB, HSV, YCrCb, YUV, LUV, HLS.

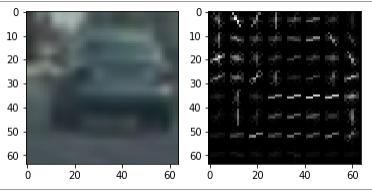
For example, if you want to recognize the image of an object under some shadow, we can use some of their properties like Light or saturation of image to classify the vehicles more effectivly on some shadows. This can be done by using HLS colorspace channel.

***3. Color Histogram Extraction:***

We take color histogram of the image with the help of function numpy.histogram() . This can be very crusial while storing different features of image.

***4. HOG:***

As described above, HOG is the final step which we used in feature extraction process. Function which we have used for HOG is hog(). Below is the visualization of hog feature of an image:



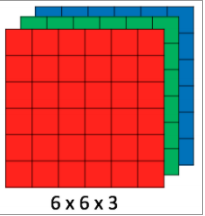
*Figure 3.3.1: Hog feature of a car*

***5. After Applying SVM Training Classifier***

After the feature extraction is done, now comes training the classifier. We have used rbf SVM (Radial basis function in Support Vector Machine). Earlier we tried using Linear SVM model, but there were too many areas where our code was not able to detect vehicles due to less accuracy. Using rbg SVM increased my accuracy to 99.13 %.

***3.3.2 Convolutional Neural Network (CNN):***

CNN image classifiers take an input image, process it and classify it under certain categories (eg Dog, Cat, Tiger, Lion). Computers view an input image as an array of pixels and it depends on the image resolution. Depending on the image resolution, it will see h \* w \* d (h = height, w = width, d = dimensions). E.g., an image of a 6 x 6 x 3 array of matrices of RGB (3 refers to RGB values) and an image of a 4x 4 x 1 array of matrices of a grayscale image.



*Figure 3.3.2: Array of RGB Matrix*

Technically, in-depth learning the CNN model for training and testing, each input image will pass through a series of layers decorated with filters (kernels), pooling, fully connected layers (FC), and the softmax function. To classify an object between 0. And 1. The image below is a complete flow of CNN to process an image and classify objects based on values.

*Convolution Layer :*

Convergence is the first layer to extract features from an input image. By learning image features using small sections of input data,pixels maintain therelationship between pixels. It is a mathematical operation that takes an image matrix and 5 inputs such as a filter or kernal.Consider. 5 x 5 whose image pixel value is 0, 1 and filter the matrix 3 x 3 as shown.

Then the convolution of the 5 x 5 image matrix is multiplied by a 3 x 3 filter matrix called the "feature map" as the output below.

*Pooling Layer :*

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains the useful information.

Max pooling take the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

Finally, we have an activation function such as softmax or sigmoid to classify the outputs as cat, dog, car, truck etc.,

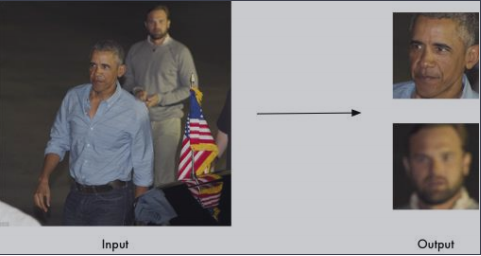
Full CNN architecture

***3.3.3 Face\_Recognition***

Recognize and manipulate faces from Python or from the command line with the world’s simplest face recognition library. Built using dlib’s state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labelled Faces in the Wild benchmark. This also provides a simple face\_recognition command line tool that lets you do face recognition on a folder of images from the command line!

Finding faces in the pictures

Finding all the faces that appear in a picture:



*Figure 3.3.3: Finding faces.*

***3.3.4 Adjusting Tolerance / Sensitivity:***

If you are getting multiple matches for the same person, it might be that the people in your photo look similar and a lower tolerance value is needed to make face comparisons more strict. You can do that with the “--tolerance” parameter. The default tolerance value is 0.6 and lower numbers make face comparisons more strict:

***3.3.5 Speeding up Face Recognition***

Face recognition can be done in parallel if you have a computer with multiple CPU cores. For example if your system has 6 CPU cores, you can process about 6 times as many images in the same amount of time by using all your CPU cores in parallel.

If you are using Python 3.4 or newer, pass in a--cpus <number\_of\_cpu\_cores\_to\_use>parameter:

***3.4 ThingSpeak***

ThingSpeak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications.

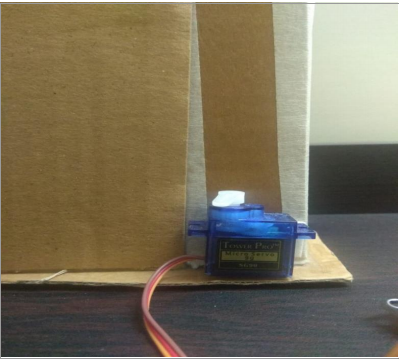
1. **RESULTS**



*Figure 4.1: Face Detecting*



*Figure 4.2: Door closed*

**

*Figure 4.3: Door Unlocked*

1. **CONCLUSION**

It presents the proposed system door access system using face recognition. The system has been successfully tested with Home Door Lock access control by verifying the nominal facial images based on the facial recognition method. The proposed system is fully standalone and wireless to create a reliable, robust, easy to operate and low cost security system. Internet communication is achieved by connecting via USB cellular data card. A battery power source has been provided to successfully build this entire system as a standalone security device. We conclude that the various operations are successfully tested and the results documented.

1. **REFERENCES**

[1] Hteik Htar Lwin, Aung Soe Khaing, Hla Myo Tun, “Automatic Door Access System UsingFace Recognition”, International Journal of Scientific & Technology Research Volume 4, Issue 06, June 2015.

[2] Sadeque Reza Khan, Ahmed Al Mansur, Alvir Kabir, Shahid Jaman, Nahian Chowdhury, “Design and Implementation of Low Cost Home Security System using GSM Network”, International Journal of Scientific & Engineering Research, Volume 3 Issue 3, March 2012.

[3] Shambhavi V. Chippa , Dr. R. R. Dube, 2019, AWS EC2 based Home Security System using Face.