[[1]](#footnote-1)

**Surgically altered face recognition using Biometrics**

Guneev Kaur Shatadiya Saha

*Abstract*—Face recognition has been one of the most interesting and important research fields in the past two decades. Altering facial appearance using surgical procedures have become obstruction for face recognition algorithms. These procedures amend the facial features and skin texture thereby providing a makeover in the appearance of face. The most challenging task for face recognition in these application scenarios is the development of robust face recognition systems. This paper proposes tow algorithms: a multimodal bio-metric feature extractor algorithm and a multiple granular algorithm to improve the current method of matching the face images before and after plastic surgery. Also, we will discuss the efficiency of the algorithms after the experiment.

# **INTRODUCTION**

Plastic surgery procedures can significantly alter facial appearance, thereby posing a serious challenge even to the state-of-the-art face matching algorithms. These procedures provide a proficient and enduring way to enhance the facial appearance by correcting feature anomalies and treating facial skin to get a younger look. Apart from cosmetic reasons, plastic surgery procedures are beneficial for patients suffering from several kinds of disorders caused due to excessive structural growth of facial features or skin tissues. These procedures amend the facial features and skin texture thereby providing a makeover in the appearance of face.

Facial plastic surgeries are typically performed either locally or globally:

• **Local Surgery:**

To correct defects, anomalies or to improve general skin texture, e.g., to correct congenital defects such as cleft lip and palate, to improve nose structure, chin, etc.

• **Global Surgery:**

To reconstruct the complete facial structure for example, for patients with severe burns. Though facial plastic surgeries can be misused by criminals to avoid law-enforcement, typically the goal of these surgeries in not to create a new identity.

Our proposed methodologies for surgically altered face detection is for local surgery.

# **RELATED WORK**

Matching post-surgery images with pre-surgery images becomes an arduous task for automatic face recognition

algorithms.

Several researches have been carried out accordingly, some of them are discussed below:

Aggarwal et al. [1] proposed sparse representation approach on local facial fragments to match surgically altered face images. But in this approach the main disadvantage is that, it requires multiple samples of data. Also, the identification accuracy is less (21.5% - 40%).

Singh et al. [2] analyzed several types of local and global Feature Extraction Matching using distance matching square algorithm Identification and Verification Feature Extraction plastic surgery procedures and their effect on different face recognition algorithms. They have experimentally shown that the nonlinear variations introduced by surgical procedures are difficult to address with current face recognition algorithms. The performance of their system is subjected to the neutral expression and proper illumination. If we include other covariates such as pose, expression, and illumination, the performance deteriorates.

F. Li. and H. Wechsler et.al., [3] This paper advocates robust part based face recognition using boosting and transduction. The face representation used spans a multiresolution (golden resolution) grid that captures partial information at different scales to accommodate different surveillance scenarios including human identification from distance.

M. De Marisco, M. Nappi et.al., [4] Proposed the use of region-based strategies for addressing the problem of face recognition after plastic surgery. FARO (Face Recognition against Occlusions and Expression Variations) divides the face into relevant regions and code them independently since different region gives different information. FACE (Face Analysis for Commercial Entities) applies a localized version of image correlation index.

Himanshu S. Bhatt et.al., [5] in his research presents a multi objective evolutionary granular computing based algorithm for recognizing faces altered due to plastic surgery procedure, the proposed algorithm starts with generating non-disjoint face granules where each granule represents information at different resolution and sizes. Two feature extractors, namely Extended Uniform Circular Local Binary Pattern (EUCLBP) and Scale Invariant Feature Transform (SIFT) are used for extracting discriminating features from face granules. Later, different responses are unified in an evolutionary manner using a multi objective genetic approach for improved performance.

We followed the following research paper:

Ms. Abha R. Gulhane Dr. S. A. Ladhake Prof. S. B. Kasturiwala et.al.,[1] which discusses about Surgically Altered Face Images Recognition Using Multimodal Biometric Features. The paper proposes a method of surgically altered face detection using Local Binary Pattern for feature extraction and matching using Matlab.

We adopted the main idea of the paper and implemented Local Binary Pattern in Python for feature extraction and matching the images. Along with it, we also tried different prediction algorithms using OpenCV classifiers and face recognizers and compared the accuracy of the algorithms in face detection as well as in correct prediction of post-surgery face labels.

# **BACKGROUND**

## Different Stages of Face Recognition

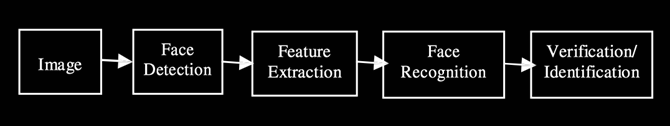


Fig 1: Diagram showing different stages involved in Face Recognition

### **Face Detection:** To detect the face from the given image.

### **Feature Extraction:** Extract features from the detected face.

### **Face Recognition:** Recognizes the face based on the extracted features.

### **Verification/Identification:** Verify or identify the faces from the training data set.

# **IV. PROPOSED METHODOLOGIES**

Our proposed methodologies for identifying the local surgically altered face recognition are:

i. Matching algorithm using Local Binary Pattern (LBP).

ii. Prediction algorithm by predicting the labels of post-surgery images.

**V. Matching algorithm**

**A. Local BINARY PATTERN**

Local binary pattern is a nonparametric method and it has aroused demanding interest in image processing, computer vision and related applications. LBP is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

The most important property of the LBP operator in real-world applications is its robustness to monotonic gray-scale changes caused, for example, by illumination variations. Another important property is its computational simplicity, which makes it possible to analyze images in challenging real-time settings.

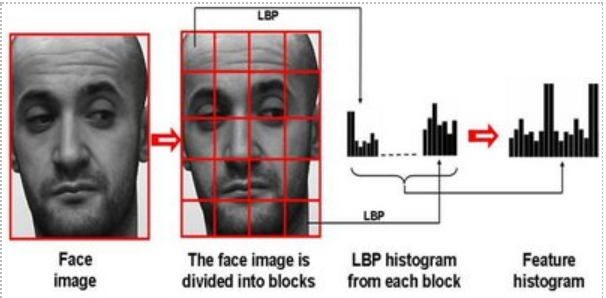
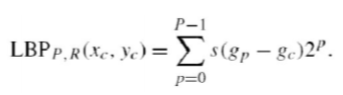


Fig 2: Figure explaining Local Binary Pattern

Consider a 3\*3 pixel with (XC, YC) intensity value be GC and local texture as T = t (G0, G1, G2, G3, G4, G5, G6, G7) where Gi (i =0, 1, 2, 3, 4, 5, 6, 7) corresponds to the grey values of the 8 surrounding pixels. These surrounding pixels are threshold with the center value GC as t (s (G0 - GC) .... s (G7 - GC)) and the function s(x) is defined as, s(z) = 1, z≥ 0 = 0, z.



## **STEPS INVOLVED:**

The steps of implementation in Python are as follows:

1. Dataset collection of pre-surgery and post-surgery images
2. Preprocessing
3. Grayscale Conversion
4. Feature Extraction using LBP [scikit-learn]
5. Matching using chi-square Distance
6. Calculation of Face Recognition Rate(Accuracy)

### **Dataset collection:**

### 100 publicly available pre-surgery images and post-surgery images from plastic surgery operation cases are collected from various hospitals websites located in United States. The images are labelled manually. There is only one image per person in pre-surgery as well as in posy-surgery.

Fig 3: Figure showing Pre-surgery and Post-surgery

images

### **Preprocessing of images:**

### All the images are preprocessed using SciPy image processing modules like resizing and normalizing.

### **Grayscale Conversion:**

### LBP works on grayscale images. All the images are converted to grayscale using OpenCv library.

### **Feature Extraction using LBP**:

### Local Binary Pattern Texture Classification function is used from scikit-learn toolkit which gives output of a (N, M) array

### **Matching:**

### The output of each post-surgery image is matched with all the pre-surgery images using KF means algorithm. It generates a score for all pre-surgery images for one post-surgery image. The image with the lowest score is the matched image.

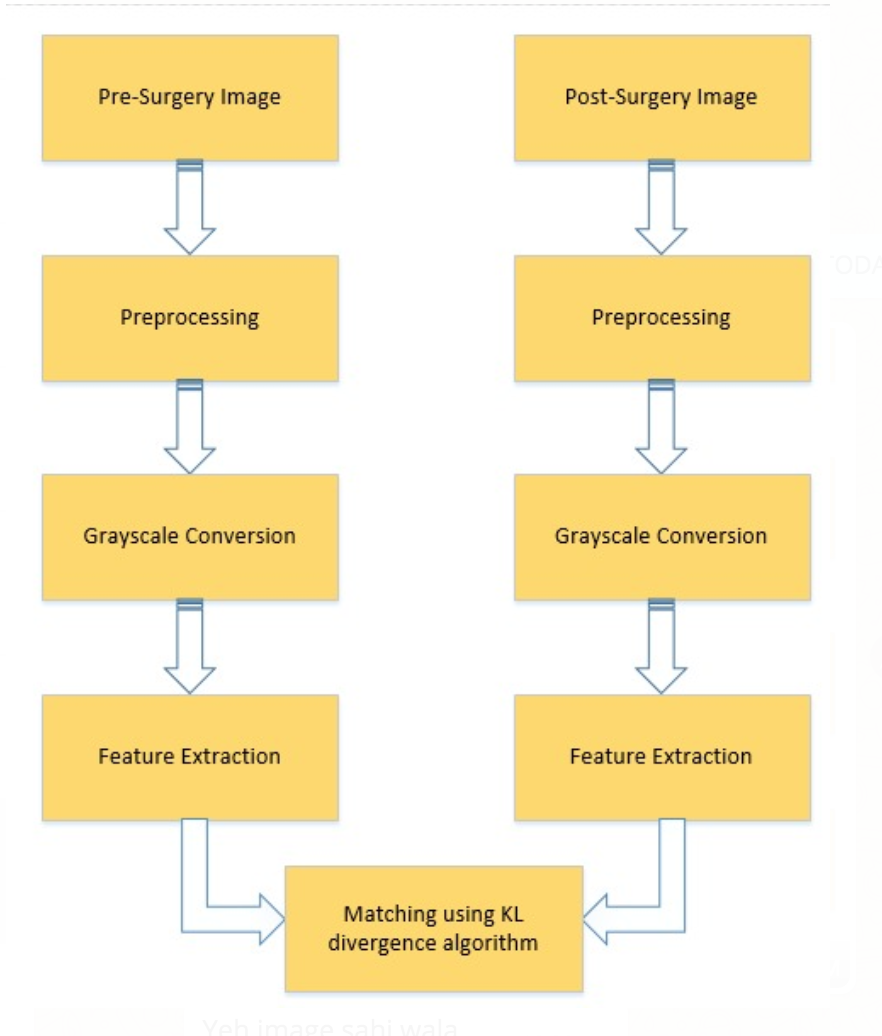


Fig 4: Diagram showing different steps involved in the Matching Algorithm.

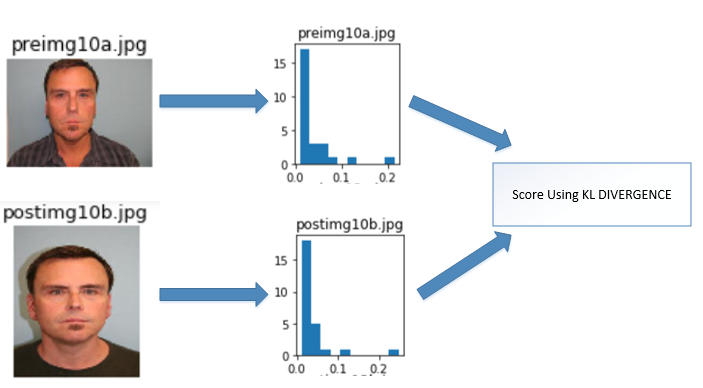


Fig 5: Diagram showing generation of histogram and calculation of score using KL divergence algorithm

The histogram of each post-surgery image is matched with all the pre-surgery images using Kullback-Leibler divergence algorithm and gives put a score as shown.

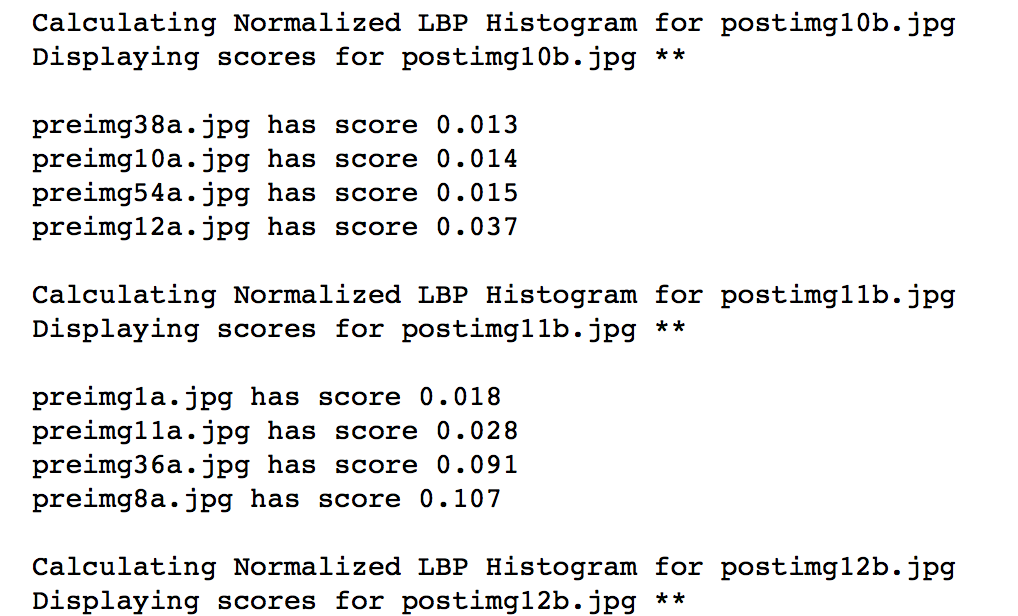


Fig 6: Score calculation in the Matching Algorithm.

The pre-surgery image with the minimum score is the correct match for the post-surgery image

**C. RESULTS:**

Owing to the dataset used for this algorithm which consists of only one image per person, this proposed algorithm gave a very low matching accuracy of 16%.

We further decided to move with method of predicting the post-surgery images based on their pre-surgery images using OpenCV library.

# **VI. PREDICTION ALGORITHMS**

**A. OpenCV**

OpenCV library is used for Face prediction algorithms. It involves two main steps:

**1) Data Collection:**

collecting facial data of person you want to identify.

**2) Recognizer Training**:

It involves feeding of data and their corresponding names to the recognizer so that it can learn.

Preparing the training data using cascade classifiers:

* LBP cascade
* HAAR cascade.

**2)Face Recognizer:**

In this step, new faces will be feed to model and checking that whether model is able to recognize it or not.

Face Recognition Algorithms which provide different functions to recognize a given face:

* FisherFaceRecognizer
* EigenFaceRecognizer
* LBPHFaceRecognizer

We performed our experiments with three different combinations of Classifier and Recognizer on same dataset, our pre-surgery images as training dataset and post-surgery images as testing dataset:

* LBPCascade Classifier with LBPHFaceRecognizer
* LBPCascade Classifier with EigenFaceRecognizer
* HAARCascade Classifier with LBPHFaceRecognizer

**B. STEPS INVOLVED:**

The steps of implementation in Python are as follows:

1)Dataset collection of pre-surgery and post-surgery images

2)Grayscale Conversion

3)Loading the required classifiers.

4)Face detection using the classifiers

5)Training the pre-surgery (training) dataset using recognizer.

6)Prediction of label of post-surgery (testing) image using the recognizer.

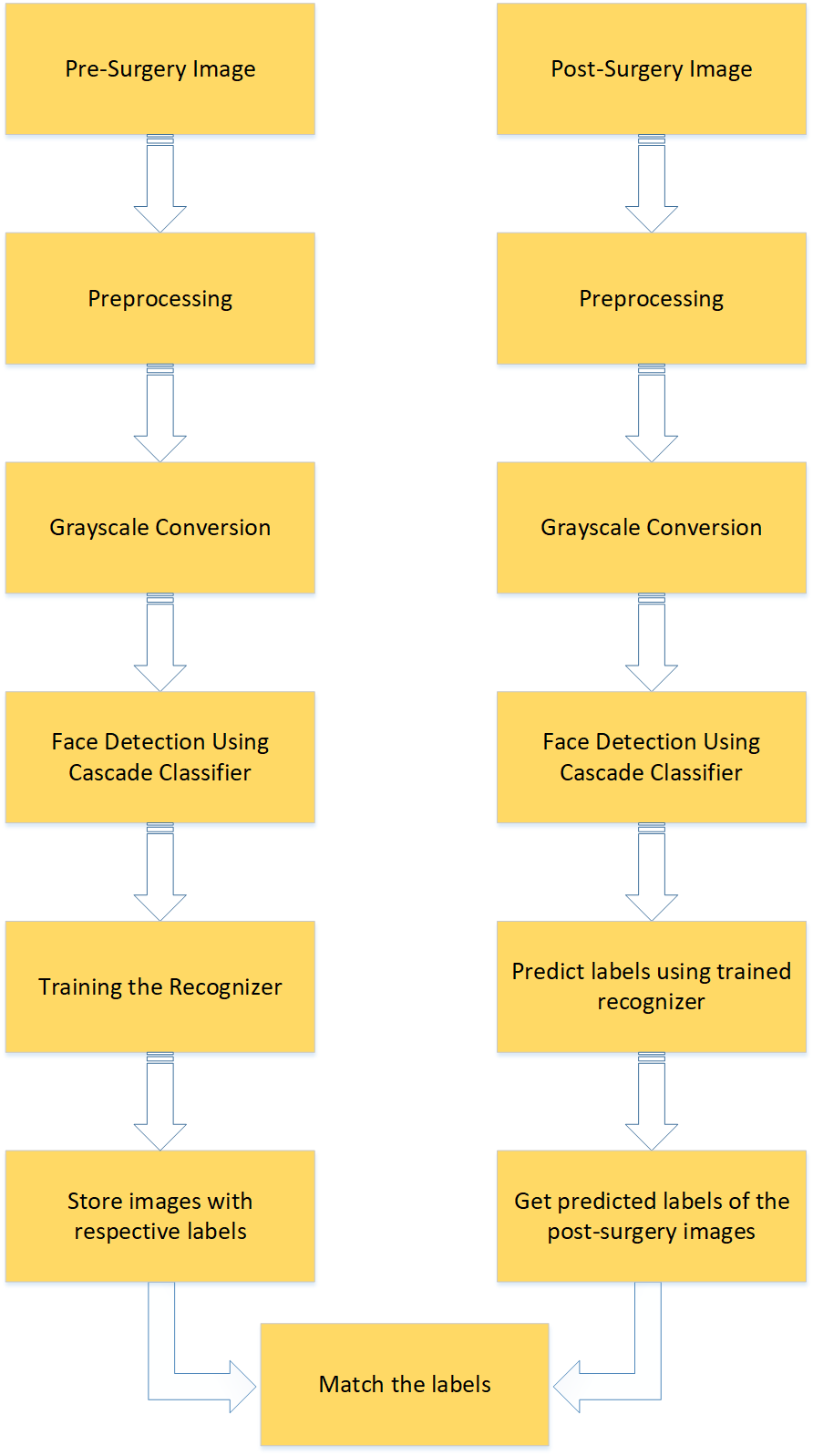


Fig 7: Diagram showing different steps involved in the Prediction Algorithm.

### **Dataset collection**:

Frontal and side pre-surgery and post-surgery images of 28 people from different plastic surgery operation cases are collected from hospitals in United States.

The images are labelled manually in hierarchical manner.

|----------------------s1

| |--------1.png

| |--------2.png

| |--------3.png

|----------------------s2

| |--------1.png

| |--------2.png

| |--------3.png

|----------------------s3

Fig 8: Figure showing the dataset images labelling in hierarchical structure

### **Preprocessing of Images:**

### Images are resized and reshaped (for EigenFaces Recognizer) using OpenCV libraries.

### **Grayscale Conversio**n:

### We have converted the images to grayscale since Cascade classifiers work only on the grayscale images.

### **Loading the required classifiers:**

### Loads a .xml classifier file which can either be HAAR or LBP classifier. LBP features are integer in contrast to Haar features, so both training and detection with LBP are several times faster than with Haar features. Regarding the LBP and Haar detection quality, it depends on training: the quality of training dataset first and training parameters too. It’s possible to train a LBP-based classifier that will provide almost the same quality as Haar-based one.

### **Face detection using the classifiers:**

### detectMultiScale () function of cascade classifier is used to detect faces of different sizes in the input image.

### **Training the pre-surgery dataset:**

### Training the faces detected using FaceRecognizer algorithm.

### **Prediction of label of post-surgery (testing) image:**

The testing images are loaded, and the same process of face detection is performed using the same classifier. The labels of these detected faces are predicted using the predict function of the recognizer.

## **LBPCascade Classifier with LBPHFaceRecognizer**

**LBPCascade Classifier**:

In this predication algorithm, we load LBPCascade\_FrontalFace.xml which contains different nodes. These nodes are connecting dots of a facial layout.

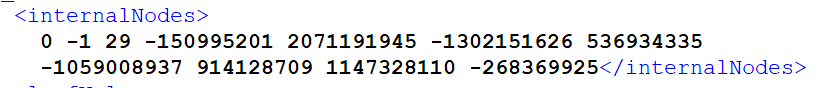


Fig 9: Image showing LBPCascafeClassifier\_frontal.xml

These internal nodes are the rectangles of the trained face images (i.e. these areas have distinct features and can be used to differentiate faces from non-face images). For lbpcascade\_frontalface.xml has 139 rectangles. Each rectangle's x, y points are multiplied with a constant number to make additional three rectangles, so one rectangle represents four rectangles. In internalNode, the first two numbers 0 and -1 represents left and right.

They represent left leaf Value and right leaf Value. The third one is the feature index. If we put those 139 rectangles into an array, that feature index refers to the array index. The last eight numbers represent corner point subtractions from four rectangles. These are calculated from the integral images, so the numbers are quite big.

In this way, the xml classifier file detects the faces from the images.

**LBPHFaceRecognizer:**

In this case, LBPHFaceRecognizer is loaded using createLBPHFaceRecognizer () function. This is used to train the recognizer with our training images with detected faces and associated labels. The train () function stores the detected faces with the labels such that faces of a same person has a same label.

Predict () function of the recognizer predicts the label of the test image (with detected face) along with a confidence value.

If the actual label of the image matches with the predicted image, it states that the post-surgery image is correctly recognized and labelled.

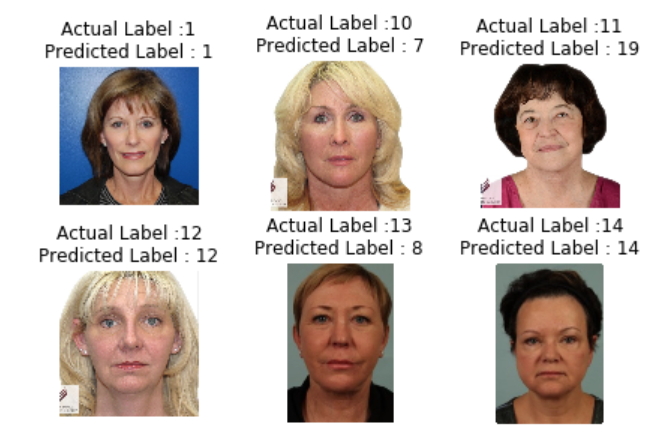


Fig 10: Images showing Actual label and the Predicted label of the Post-Surgery images as predicted by the Face Recognizer predict function.

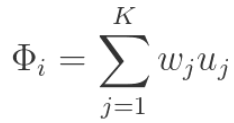
**D. LBPCascade Classifier with EigenFaceRecognizer**

The cascade classifier is the same as described in the previous bit.

Here we used, EigenFaceRecognizer for training and predicting the images. The functions train () and predict () works similarly as described above except that they work on Eigen Faces.

**Eigen Face Recognition:**

The idea behind Eigen Faces is to represent a face as a linear combination of a set of basis images as:



Where \displaystyle \Phi_i  represents the i^{th} face with the mean subtracted from it, w_j represent weights and u_j  the eigenvectors.

Here we want to find a set of images (called Eigenfaces, which are nothing but Eigenvectors of the training data) that if we weigh and add together should give us back an image that we are interested in.

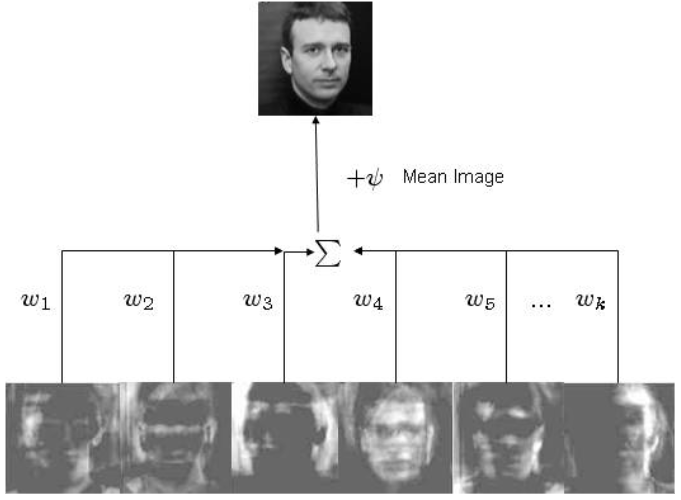


Fig 11: Image showing generation of a face by applying weights to the Eigen Face

In the above figure, a face that was in the training database was reconstructed by taking a weighted summation of all the basis faces and then adding to them the mean face. These Eigenfaces (ghost-like) were prepared using images from the MIT-CBCL database (after adjusting the brightness).

Eigenfaces considers face recognition as a 2-D recognition problem, this assumes that at the time of recognition, faces will be mostly upright and frontal. It transforms the face images in to a set of basis faces, which essentially are the principal components (Eigen Vectors) of the face images.

These Eigenfaces can be used to represent both existing and new faces: we can project a new (mean-subtracted) image on the Eigenfaces and thereby record how that new face differs from the mean face. The Eigen Vectors associated with each Eigenface represent how much the images in the training set vary from the mean image in that direction.

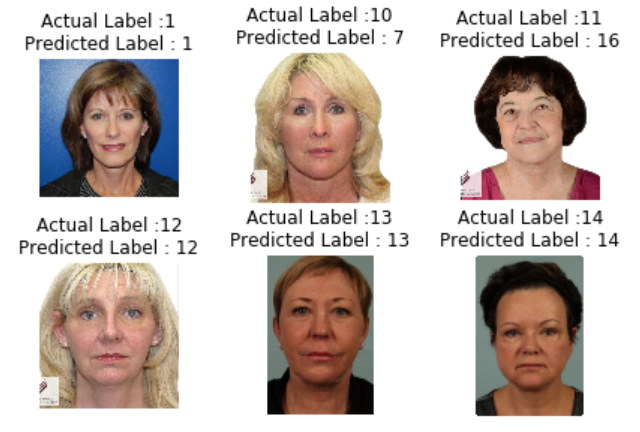


Fig 12: Images showing Actual label and the Predicted label of the Post-Surgery images as predicted by the Face Recognizer predict function.

**D. HAARCascade Classifier with LBPHFaceRecognizer**

**HAARCascade Classifier:**

Object Detection using Haar feature-based cascade classifiers is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects (in our case, faces) in other images.

Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it.

For this, Haar features shown in below image are used. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle. Adaboost is used to select the best features.

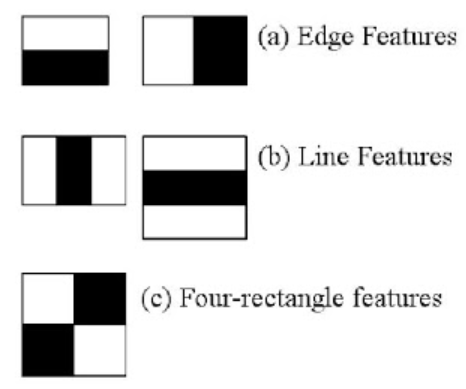


Fig 13: Images showing HAAR classifiers for feature detection

We apply every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. We select the features with minimum error rate. Final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier.

So, the process is we take an image with 24x24 window. Apply 6000 features to it. Check if it is face or not. We initially check if the window is not a face region so that we can discard and not process it again. In this way, face detection is performed using HAAR cascade.

OpenCV comes with HAARCascade Classifier to detect faces from the images. HAARCascade Classifiers are in the form of XML files. In our experiment, we used xml file for frontal face detection.

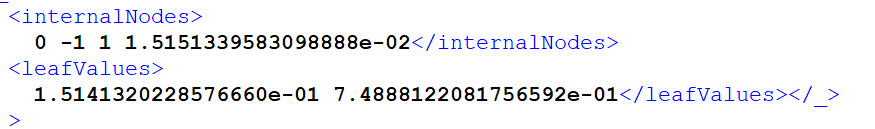


Fig 14: Image showing HAARCascafeClassifier.xml

After face detection, we use the LBPHFaceRecognizer to train the pre-surgery images and detect the labels of the post-surgery images as explained earlier.

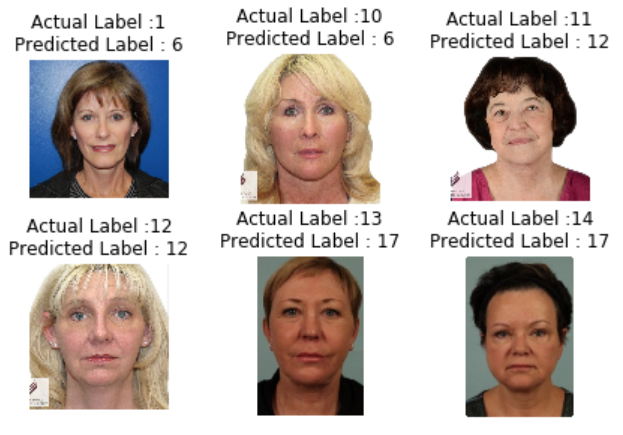


Fig 15: Images showing Actual label and the Predicted label of the Post-Surgery images as predicted by the Face Recognizer predict function.

**VII. COMPARISON OF RESULTS**

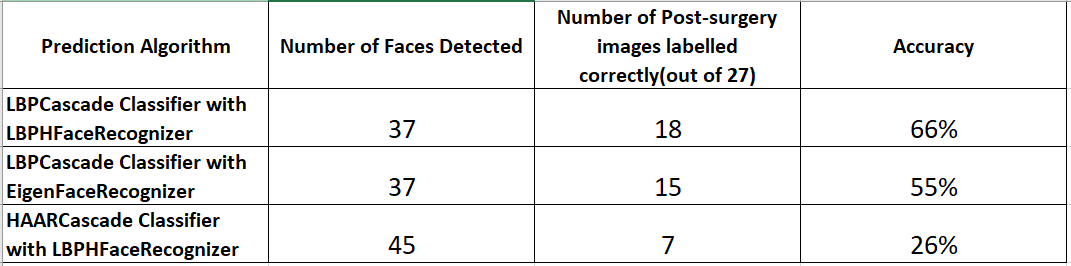


Fig 16: Table showing the results of the above experiments.

The results of the above experiments show that:

a) HAAR classifier detects more faces than the LBPCascade classifier

b) LBP Face Recognizer predicts more number of faces than Eigen Face Recognizer.

**VIII. CHALLENGES**

We faced major challenges in the dataset collection. The images collected as mentioned above are from publicly available websites of different hospitals in United States. The post-surgery images are of any local surgery on face like brow lift, face-lift, jaw lift etc. As we could collect only limited number of available images, we had maximum of two images per person before and after surgery (for a few people, we had three images). Due to the less number of training images (pre-surgery images), our training model was not very accurate. Also, not all the images are frontal images (many images are side wards), face detection rate is low as the features couldn’t be identified correctly. Also, due to the different image sizes of both pre-surgery and post-surgery datasets, we faced some issues in preprocessing of the images to resize all the images to a same size. We tried different face detection and recognizers algorithms so that we could increase the face detection rate, train the model accurately and hence, predict the labels of the post-surgery images correctly and compare the accuracy of the algorithms.

**VIII. REFERENCES**

[1] Ms. Abha R. Gulhane Dr. S. A. Ladhake Prof. S. B. Kasturiwala, “Surgically Altered Face Images Recognition Using Multimodal Biometric Features”, IETE 46th Mid Term Symposium “Impact of Technology on Skill Development” MTS- 2015 Special Issue of International Journal of Electronics, Communication & Soft Computing Science and Engineering, ISSN: 2277-9477

[2] G. Aggarwal, S. Biswas, P. J. Flynn, and K. W. Bowyer, “A sparse representation approach to face matching across plastic surgery,” in Proc. Workshop on the Applications of Computer Vision, 2012, pp. 1–7.

[3] R. Singh, M. Vatsa, H. S. Bhatt, S. Bharadwaj, A. Noore, and S. S. Nooreyezdan, “Plastic surgery: A new dimension to face recognition,” IEEE Trans. Inf. Forensics Security, vol. 5, no. 3, pp. 441–448, Sep. 2010.

[4] F. Li. and H. Wechsler, “Robust part-based face recognition using boosting and transduction,” in Proc. Int. Conf, Biometrics: Theory, Applications and Systems, 2007, pp.1-5.

[5] M. De Marisco, M. Nappi, D. Riccio, and H. Wechsler, “Robust face recognition after plastic surgery using local region analysis”, Vol.6754, pp.191-200.

[6] [Pietikäinen](http://www.scholarpedia.org/article/User:Matti_Pietik%C3%A4inen),Matti ,”Local Binary Patterns”, 2010 [Online], Available:http://www.scholarpedia.org/article/User:Matti\_Pietik%C3%A4inen

[7] [Trivedi](https://onionesquereality.wordpress.com/author/shubhendu09/), Shubhendu.” Face Recognition using Eigenfaces and Distance Classifiers: A Tutorial”,2009[Online], Available:

https://onionesquereality.wordpress.com/2009/02/11/FACE-RECOGNITION-USING-EIGENFACES-AND-DISTANCE-CLASSIFIERS-A-TUTORIAL/

1. . [↑](#footnote-ref-1)