A Project Report

on

Design of Face detection and recognition system using Machine Learning

Submitted in partial fulfillment of the requirement for the award of the degreeof

Bachelor of Technology in Computer Science and Engineering

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr. HradeshKumar
Assistant Professor
Department of Computer Science and Engineering

Submitted By

18SCSE1140018–Gaurav Sharma

18SCSE1140056–Aryan Yadav

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA

DECEMBER - 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **"Design of face detection and recognition system using Machine learning "** in partial fulfillment of the requirements for the award of the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** Submitted in the **School of Computing Science and Engineering** of Galgotias University, Greater Noida, is an original work carried out during the period of **JULY-2021 to DECEMBER-2021**, under the supervision of **Mr.Hradesh Kumar, Assistant Professor**, **Department of Computer Science and Engineering** of School of Computing Science and Engineering ,Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any otherplaces.

18SCSE1140018–Gaurav Sharma

18SCSE1140056–Aryan yadav

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor

(Mr.Hradeshkumar,

Assistant Professor)

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **18SCSE1140018–Gaurav Sharma, 18SCSE1140056–Aryan Yadav** has been held on_____________________and his/her work is recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Dean

Date:

Place:

ABSTRACT

Today's pandemic situation has transformed the wayof educating a student.Education is undertaken remotely through online platforms.In addition to the way the online course contents and online teaching, it has also changed the way of assessments. In online education, monitoring the attendance ofthe students is very important asthe presence of students is partof a good assessment for teachingand learning. Educational institutions have adopting online examination portals for the assessments of the students. These portals make useof face recognition techniques to monitor the activities of the students and identify the malpractice done by them. This is done by capturing the students' activities through a web camera and analyzing their gestures and postures. Image processing algorithms are widely used in the literature to perform face recognition. Despite the progress made to improve the performance of face detection systems, there are issues such as variations in human facial appearance like varying lighting condition, noise in face images, scale, pose etc., that blocks the progress to reach human level accuracy. The aim of this study is to increase the accuracy of the existing face recognition systems by making use of SVM and Eigenface algorithms. In this project, an approach similar to Eigenface is used for extracting facial features through facial vectors and the datasets are trained using Support Vector Machine (SVM) algorithm to perform face classification and detection. This ensures that the face recognition can be faster and be used for online exam monitoring.

<u>Contents</u>

List of Figures

Introduction

In our education system, exams play a major role in assessing and evaluating a student's knowledge. Exams, tests, and marks are some of the important aspects employed by educational institutions to evaluate the performance of the students. Most of the exams have strict set of rules and procedures that the student has to follow in order to write the examinations. Theserules ensure that the students write their examinations in a proper manner and also to make sure that there is no cheating and malpractice in the examinations. Educational institutions are very much capable of conducting examinations and monitoring the students effectively.In today's pandemic situation, it is impossible to conduct offline examinations. Hence, educational institutions are adapting to online classes and online examinations. Onlineclasses and online examinations are the only possible practices in the current situation. Monitoring students during online classes and examinations becomes a difficult process. Thisproblem can be solved by using effective face recognition systems

LITERATURESURVEY

A face recognition system using Eigenface method wasproposed by Dhavalsinh to monitor the attendance of the students, where the face acts as the main index. Eigenface is a set of eigenvectors used in face recognition and detection. it is used to determine the variation among multiple faces by performing a statistical analysis on the facial images. Sirovich and Kirby designed the Eigenfaces approach to do facial recognition and the same was used by Matthew Turk and Alex Pentland for face classification. Kranthikiran and Pulicherla made use of Eigenfaces and Principal Component Analysis (PCA) to perform face detection for campus surveillance. Continuous face biometric recognition has been used by Fayyoumi and Zarrad in developing a prototype for conducting online examinations. The prototype has been evaluated by obtaining feedbacks from different experts through a surveyusing a five-point Likert scale. The proposed system contains a question bank to assist the instructors in generating different tests randomly. Kamencay et al. suggested a face recognition system using Convolutional Neural Network (CNN). The authors used OLR dataset comprising 400 diverse entities (40 categories/10 images for every category) to carry out the experiments and validate their results. The detection accuracy of the suggested method has been compared with the three popular image recognition approaches like PCA, Local Binary

Patterns Histograms (LBPH) and KNN. In comparison withthese methods, the proposed CNN-based method performs better by achieving an identification accuracy of about 98.3%. Traoré et al. used a multimodal biometric framework to authenticate the participants in online examinations. Theframework consists of three modalities such as mouse dynamics, keystroke dynamics, and face biometrics to check theauthenticity. This framework has been included as a module in ExamShield which is an online exam monitoring tool. Continuous face biometric recognition has been used by Fayyoumi and Zarrad in developing a prototype for conducting online examinations. The prototype has been evaluated by obtaining feedbacks from different experts through a surveyusing a five-point Likert scale. The proposed system contains a question bank to assist the instructors in generating different tests randomly. Zhu et al. addressed various challenges in face detection systems by developing a novel approach namely Contextual Multi-Scale Region-based Convolution Neural Network (CMS-RCNN) which consists of two components:

1) region proposal component and) the region-of-interest (RoI) detection component. The proposed system deals with tiny face regions by grouping multi-scale information in both the components and also allows explicit body contextual reasoning. Zehenguo Yuan addresses the facial occlusion and improved the detection accuracy by developing a visual attention guidance model that guides in highlighting the

visible area in an occluded face. This model avoids setting the additional parameters by using an activation map that predicts the location and scale of the face The above methodsare the ones that are currently available for face recognition. But there is a need for a faster and reliable method that could be used for monitoring the students during their examinations. In this project, an approach similar to Eigenface is used for extracting facial features through facial vectors and the datasets are trained using an Support Vector Machine (SVM) model which is one of the popular machine learning algorithms. This ensures that the face recognitioncan be faster and can be easily used..

RESEARCHMETHODOLOGY

In this case study, we aim to use ML techniques Educational institutions conduct online exams for a large number of people and it is difficult to monitor the students manually. The proposed system focuses on designing a suitable face detection and recognition model for monitoring the students during online examinations. Here, Eigenface method is used for extracting the facial features through facial vectors and the datasets are trained using an SVM model to improve the detection accuracy. The flowdiagram for the proposed system is given in Figure 1.At first, the vector values are extracted from the images in the dataset. The extracted embeddings are then passed to the SVM classifier to train it for recognizing the faces from the input images or video frames. A. Modules Description The modules identified and implemented in the proposed system are listed below:

1. Extracting embeddings from the images

2. Training the SVM model

3. Recognizing faces from static images and video frames 1) Extracting Embeddings from Images: In this module, the feature vectors are extracted from the images. The 128 D feature vectors are called as embeddings. A Caffe based DL face detector has been used to locate faces in an image. This module makes use of Pytorch based embedder to extract the embeddings from the images in the dataset. The extracted embeddings are then stored in a pickle file in an encoded format. 2) Training the SVM Model: The extracted 128-D embeddings from the previous step each face. But to recognize different faces, we need to train a "standard" machine learning model (such as an SVM, k-NN classifier, Random Forest, etc.) on top of the embeddings.

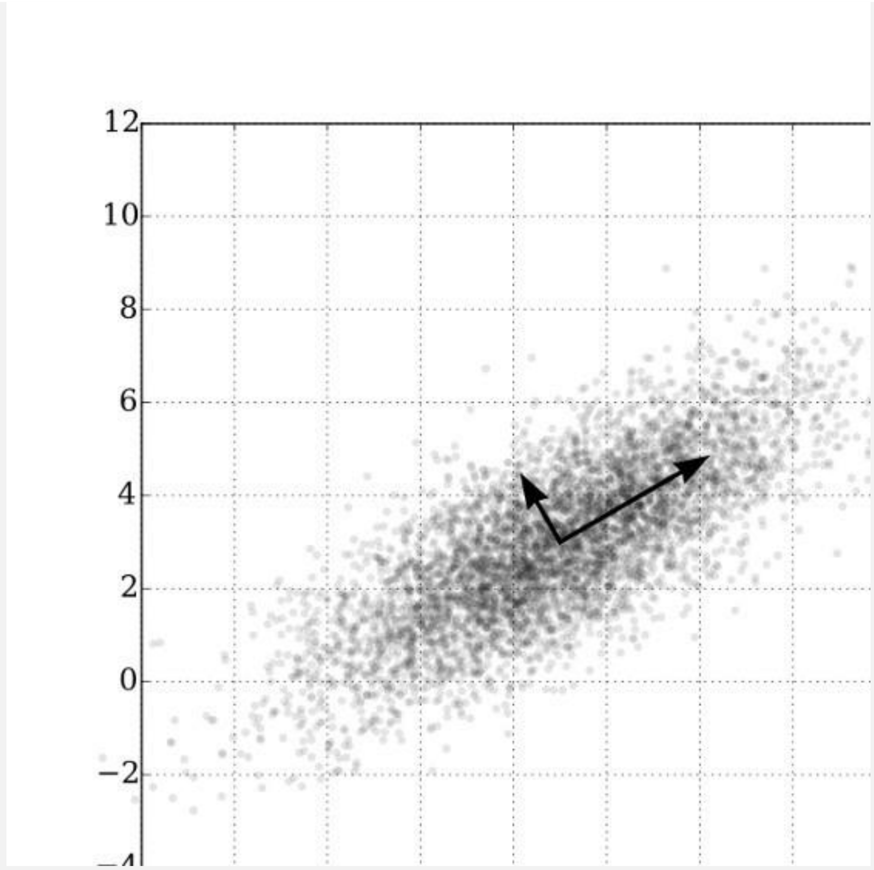
In this article, we will learn to use **Principal Component Analysis** and **Support Vector Machines** for building a facial recognition model.

First, let us understand what **PCA** and **SVM** are:

Principal Component Analysis:

Principal Component Analysis (PCA) is a machine learning algorithm that is widely used in exploratory data analysis and for making predictive models. It is commonly used for **dimensionality reduction** by projecting each data point onto only the first few principal components to obtain lower-dimensional d ata while preserving as muchof the data's variation as possible.

.



PCAofaMultivariateGaussianDistribution

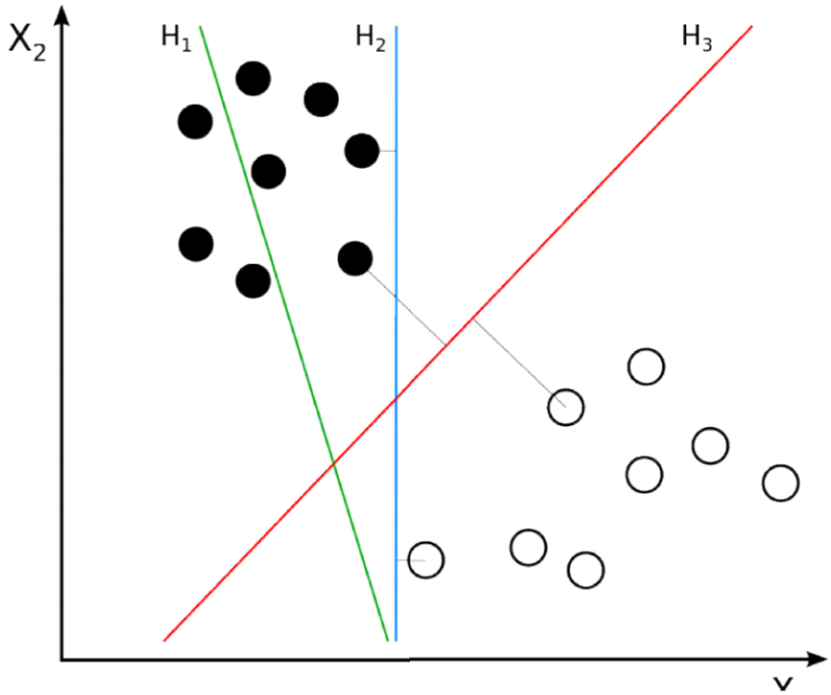
suppose we have a problem at hand for which we are collecting data. Our dataset results in several variables, many features, all of which affect the results in different aspects. We may choose to eliminate certain features, but that would mean losing information, and we don'twant that, right?

So another method of **reducing the number of features** (reducing data dimensionality) is to create new features by extracting the important information and dropping the least important ones. In this way, our information will not be lost and we will have reduced features, and therewill be fewer chances of overfitting our model.

Support Vector Machine:

Support Vector Machine (**SVM**) is a supervised machine learning model used for two-group classification problems. After giving an **SVM** model set of labeled training data for each category, they’re able to categorize new test data.

.



SVM classifies data based on the plane that maximizes the margin. TheSVM decision boundary is straight. SVM is a really good algorithm for image classification. Experimental results show that SVMs a hieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback. This is also true for image segm ntation systems, including those using a modified version SVM that uses the privileged approach.

Face detection techniques

Face detection is a computer technology that determines the location and size of a human face in the digital image. The facial features are detected and any other objects like trees, buildings and bodies are ignored from the digital image. It can be regarded as a specific case of object-class detection, where the task is finding the location and sizes of all objects in an image that belongs to a given class. Face detection, can be seen as a more general case of face localization. In face localization, the task is to identify the locations and sizes of a known number of faces (usually one). Basically, there are two types of approaches to detect

facial part in the given digital image i.e. feature based and image based approach. Feature based approach tries to extract features of the image and match it against the knowledge of the facial features. While image based approach tries to get the best match between training and testing images. The following methods are commonly used to detect the faces from a

still image or a video sequence (Fig. 4).

Features based approaches:-

Active shape model:-

Active Shape Model (ASM) focus on complex non-rigid features like actual physical andhigher level appearance of features. Main aim of ASM is automatically locating landmarkpoints that define the shape of any statistically modelled object in an image. For examples,

in an image of human being face, extracted features such as the eyes, lips, nose, mouth andeyebrows. The training stage of an ASM involves the building of a statistical facial modelcontaining images with manually annotated landmarks. ASMs is classified into three groupsi.e. Snakes, Point Distribution Model (PDM) and deformable templates.

Snakes:-The first type uses a generic active contour called snakes (Kass et al. 1988). Snakes are used to identify head boundaries. In order to achieve the task, a snake is first initialized at the proximity around a head boundary. It then looks onto nearby edges and subsequently assumes the shape of the head. The evolution of a snake is achieved by minimizing an energy function, Esnake (analogy with physical systems), denoted as:

Esnake Einternal + Eexternal

where Einternal and Eexternal are internal and external energy functions.

Internal energy is the part that depends on the intrinsic properties of the snake and definesits natural evolution. The typical natural evolution in snakes is shrinking or expanding. The external energy counteracts the internal energy and enables the contours to deviate from the natural evolution and eventually assume the shape of nearby features-the head boundary at astate of equilibria. Two main considerations for forming snakes i.e. selection of energy terms and energy minimization. Elastic energy (Erik and Low 2001) is used commonly as internal energy. Internal energy is varying the distance between control points on the snake, through which we get contour, an elastic-band characteristic that causes it to shrink or expand. On

other side external energy relay on image features. Energy minimization process is done

by optimization techniques such as the steepest gradient descent, which needs the highestcomputations. Fast iteration methods by greedy algorithms are also used. Snakes have somedemerits like contour often become trapped onto false image features and another one is thatsnakes are not suitable in extracting non convex features.

Point distribution model:-

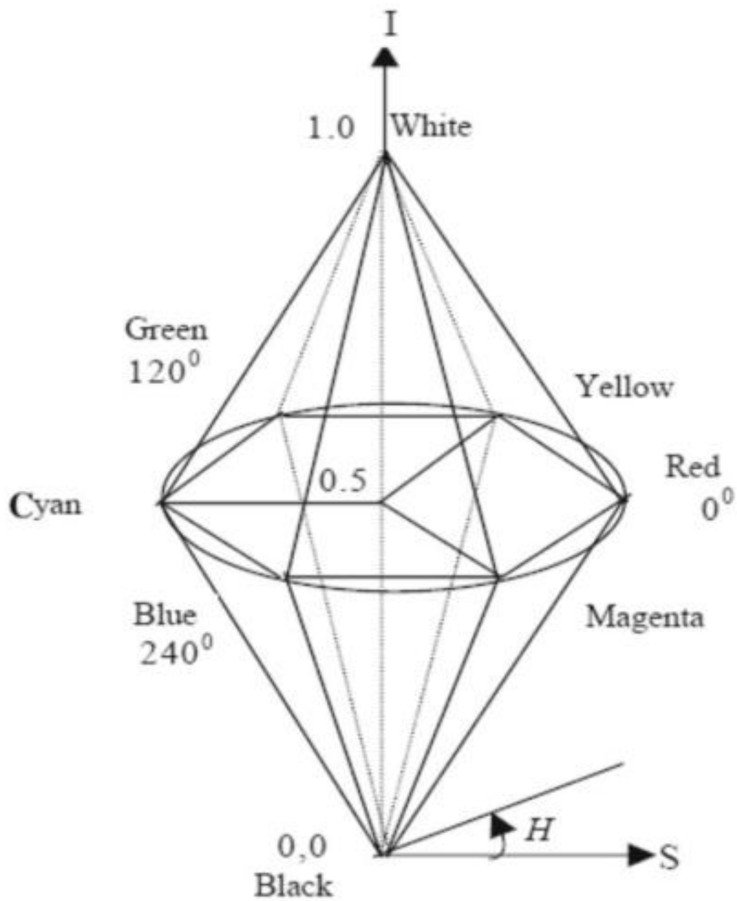
Point Distribution Model (PDM) was developed independent of computerized image analysis, and developed statistical models of shape (Erik and Low2001). The idea is that once one can represent shapes as vectors, after that they can applystandard statistical methods to them just like any other multivariate object. These models learn allowable constellations of shape points from training examples and use principal components to build model, known as Point Distribution Model (PDM). These have been used in diverse ways, for example for categorizing Iron Age broaches. The first parametric statistical shape model for image analysis based on principal components of inter-landmark distances were presented (Cootes et al. 1992). Based on this approach, they released a seriesof papers that cumulated in what we call the classical Active Shape Model.

Deformable templates

SDeformable templates take into account a priori of facial featuresto improve the performance of snakes (Yuille et al. 1992). Locating a facial feature boundarydoes not constitute an easy task because the local evidence of facial edges is difficult toorganize into a sensible global entity using generic contours. The low brightness contrastaround some of these features also makes the edge detection process problematic. Conceptof snakes is a step further by incorporating global information of the eye takes to improvethe reliability of the extraction process. Deformable templates approaches are designed tosolve this problem. Deformation is based on narrow valley, edge, peak, and brightness. Otherthan face boundary, salient feature (eyes, nose, mouth and eyebrows) extraction is a greatchallenge of face recognition.

E = Ev + Ee + Ep + Ei + Einternal

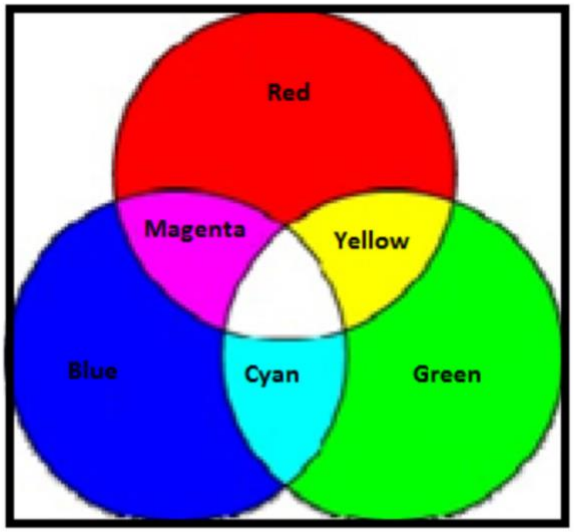
where Ev, Ee, Ep, Ei, Einternal are external energy due to valley, edges, peak and imagebrightness and internal energy.



Low level analysis :-

Skin color base Color is an important feature of human faces. Using skin color as a feature for tracking a face has several advantages. Color processing is much faster than other facial features. Under certain lighting conditions, color is orientation invariant. This property makes motion estimation much easier because only a translation model is needed for motion estimation. Tracking human faces using color as a feature has several problems like the color representation of a face obtained by a camera is influenced by many factors like, ambient light, object movement, etc. suggested simplest skin-color algorithms for detecting skin pixels have been suggested by Crowley and Coutaz in (1997). The perceived human color varies as a function of the relative direction to the illumination. Pixels for skin region can be detected using a normalized color histogram, and can be normalized for changes in intensity on dividing by luminance (Fig. 5).

Conversion of an [R, G, B] vector into an [r, g] vector of normalized color which provides a fast processing of skin detection. This algorithm fails when there are some more skin regions like legs, arms, etc. Skin color classification algorithm with YCbCr color space is also introduced. Few researchers have noticed that pixels belonging to skin regions have similar Cb and Cr values. So, the thresholds be

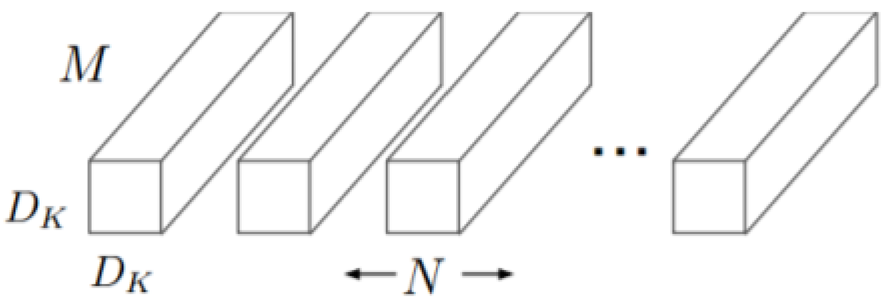


chosen as [Cr1, Cr2] and [Cb1, Cb2], a pixel is classified to have skin tone if the values [Cr, Cb] fall within the thresholds. The skin color distribution gives the face portion in the color image. This algorithm is also having the constraint that the image should be having only face as the skin region. A color predicate in HSV color space to separate skin regions from background has been defined (Kjeldsen and Kender 1996). Skin color classification in HSI color space is the same as YCbCr color space but here the responsible values are hue (H) and saturation (S). Similar to above the threshold be chosen as [H1, S1] and [H2, S2], and a pixel is classified to have skin tone if the values [H, 123 A. Kumar et al. Fig. 6 RGB color model S] fall within the threshold and this distribution gives the localized face image. Generally, three different face detection algorithms are available based on RGB, YCbCr, and HIS color space models. For implementation of these algorithms there are basically three main steps are required as the following:

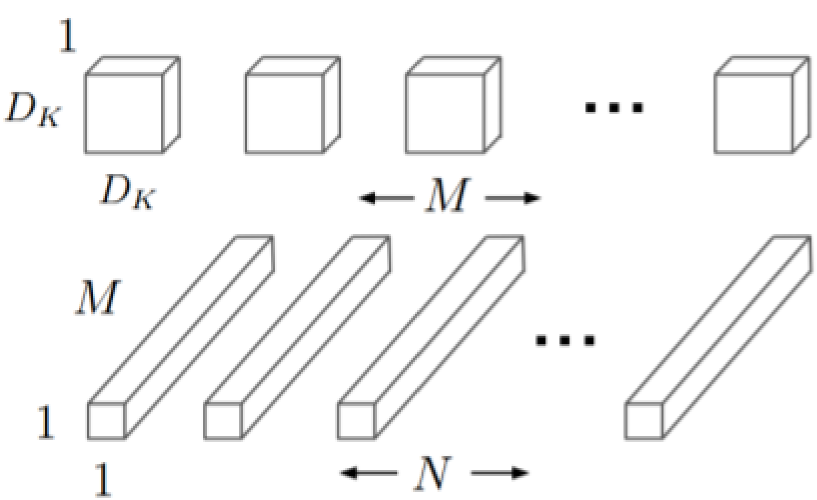
- Classify the skin region in the color space,
- Apply threshold to mask the skin region and
- Draw bounding box to extract the face image.

RGB color model

RGB colors are specified in terms of three primary colors i.e. Red (R), Green (G), and Blue (B) (Fig. 6). In RGB color space, a normalized color histogram is used to detect the pixels of skin color of an image and can be further normalized for changes in intensity on dividing by luminance. This localizes and detects the face (Subban and Mishra 2012). It is the basic color model and all other color models are derived from it. The RGB color model is light sensitive. In comparison with other color models such as YCbCr or HSI it has a major drawback that it cannot separate clearly the mere color (Chroma) and the intensity of a pixel, so it is sometimes difficult to distinguish skin colored regions. These factors contribute to the less favourable of RGB. It is widely used color space for processing and storing digital images because of the fact that chrominance and luminance components are mixed in RGB; it is not widely used in skin detection algorithm. Dhivakar et al. (2015) have proposed a method which consists of two main parts as detection of faces and then recognizing the detected faces. In detection step, skin color segmentation with thresholding skin color model combined with AdaBoost algorithm is used, which is fast and also more accurate in detecting the faces. Also, a series of morphological operators is used to improve the face detection performance. Recognition part consists of three steps: Gabor feature extraction, dimension reduction and feature selection using PCA, and KNN based classification. The system is robust enough to detect faces in different lighting conditions, scales, poses, and skin colors (Fig. 7).



(a)



(b)

A. Training Data

Since we jointly perform face detection and alignment, here we use four different kinds of data annotation in our training process: (i) Negatives: Regions that the Intersection-over-Union (IoU) ratio less than 0.3 to any ground-truth faces; (ii) Positives: IoU above 0.65 to a ground truth face; (iii) Part faces: IoU between 0.4 and 0.65 to a ground truth face; and (iv) Landmark faces: faces labeled 5 landmarks' positions. Negatives and positives are used for face classification tasks, positives and part faces are used for bounding box regression, and landmark faces are used for facial landmark localization. The training data for each network is described as follows: 1) P-Net: We randomly crop several patches from WIDER FACE [24] to collect positives, negatives and part face. Then, we crop faces from CelebA [23] as landmark faces 2) R-Net: We use first stage of our framework to detect faces from WIDER FACE [24] to collect positives, negatives and part face while landmark faces are detected from CelebA [23]. 3) O-Net: Similar to R-Net to collect data but we use first two stages of our framework to detect face.

A. The effectiveness of online hard sample mining

To evaluate the contribution of the proposed online hard sample mining strategy, we train two O-Nets (with and without online hard sample mining) and compare their loss curves. To make the comparison more directly, we only train the O-Nets for the face classification task. All training parameters including the network initialization are the same in these two O-Nets. To compare them easier, we use fix learning rate. Fig. 3 (a) shows the loss curves from two different training ways. It is very clear that the hard sample mining is beneficial to performance improvement.

B. The effectiveness of joint detection and alignment

To evaluate the contribution of joint detection and alignment, we evaluate the performances of two different O-Nets (joint facial landmarks regression task and do not joint it) on FDDB (with the same P-Net and R-Net for fair comparison). We also compare the performance of bounding box regression in these two O-Nets. Fig. 3 (b) suggests that joint landmarks localization task learning is beneficial for both face classification and bounding box regression tasks.

C. Evaluation on face detection

To evaluate the performance of our face detection method, we compare our method against the state-of-the-art methods [1, 5, 6, 11, 18, 19, 26, 27, 28, 29] in FDDB, and the state-of-the-art methods [1, 24, 11] in WIDER FACE. Fig. 4 (a)-(d) shows that our method consistently outperforms all the previous approaches by a large margin in both the benchmarks. We also evaluate our approach on some challenge photos1 . E. Evaluation on face alignment In this part, we compare the face alignment performance of our method against the following methods: RCPR [12], TSPM [7], Luxand face SDK [17], ESR [13], CDM [15], SDM [21], and TCDCN [22]. In the testing phase, there are 13 images that our method fails to detect face. So we crop the central region of these 13 images and treat them as the input for O-Net. The mean error is measured by the distances between the estimated.

RELATED WORK

Commonly used CNNs for feature extraction include a set of fully connected layers at the end. Fully connected layers tend to contain most of the parameters in a CNN. Specifically, VGG16 [10] contains approximately 90% of all its parameters in their last fully connected layers. Recent architectures such as Inception V3 [12], reduced the amount of parameters in their last layers by including a Global Average Pooling operation. Global Average Pooling reduces each feature map into a scalar value by taking the average over all elements in the feature map. The average operation forces the network to extract global features from the input image. Modern CNN architectures such as Xception [1] leverage from the combination of two of the most successful experimental assumptions in CNNs: the use of residual modules [6] and depth-wise separable convolutions [2]. Depth-wise separable convolutions reduce further the amount of parameters by separating the processes of feature extraction and combination within a convolutional layer. Furthermore, the state-of-the-art model for the FER2-2013 dataset is based on CNN trained with square hinged loss [13]. This model achieved an accuracy of 71% [4] using approximately 5 million parameters. In this architecture 98% of all parameters are located in the last fully connected layers. The second-best methods presented in [4] achieved an accuracy of 66% using an ensemble of CNNs.

III. MODEL

We propose two models which we evaluated in accordance to their test accuracy and number of parameters. Both models were designed with the idea of creating the best accuracy over number of parameters ratio. Reducing the number of parameters help us overcoming two important problems. First, the use of small CNNs alleviate us from slow performances in hardware-constrained systems such robot platforms. And second, the reduction of parameters provides a better generalization under an Occam's razor framework. Our first model relies on the idea of eliminating completely the fully connected layers. The second architecture combines the deletion of the fully connected layer and the inclusion of the combined depth-wise separable convolutions and residual modules. Both architectures were trained with the ADAM optimizer [8]. Following the previous architecture schemas, our initial architecture used Global Average Pooling to completely remove any fully connected layers. This was achieved by having in the last convolutional layer the same number of feature maps as number of classes, and applying a softmax activation function to each reduced feature map. Our initial proposed architecture is a standard fully-convolutional neural network composed of 9 convolution layers, ReLUs [5], Batch Normalization [7] and Global Average Pooling. This model contains approximately 600,000 parameters. It was trained on the IMDB gender dataset, which contains 460,723 RGB images where each image belongs to the class "woman" or "man", and it achieved an accuracy of 96% in this dataset. We also validated this model in the FER-2013 dataset. This dataset contains 35,887 grayscale images where each image belongs to one of the following classes {"angry", "disgust", "fear", "happy", "sad", "surprise", "neutral"}. Our initial model achieved an accuracy of 66% in this dataset. We will refer to this model as "sequential fully-CNN". Our second model is inspired by the Xception [1] architecture. This architecture combines the use of residual modules [6] and depth-wise separable convolutions [2]. Residual modules modify the desired mapping between two subsequent layers, so that the learned features become the difference of the original feature map and the desired features. Consequently, the desired features $H(x)$ are modified in order to solve an easier learning problem

$F(X)$ such that:

$$H(x) = F(x) + x$$

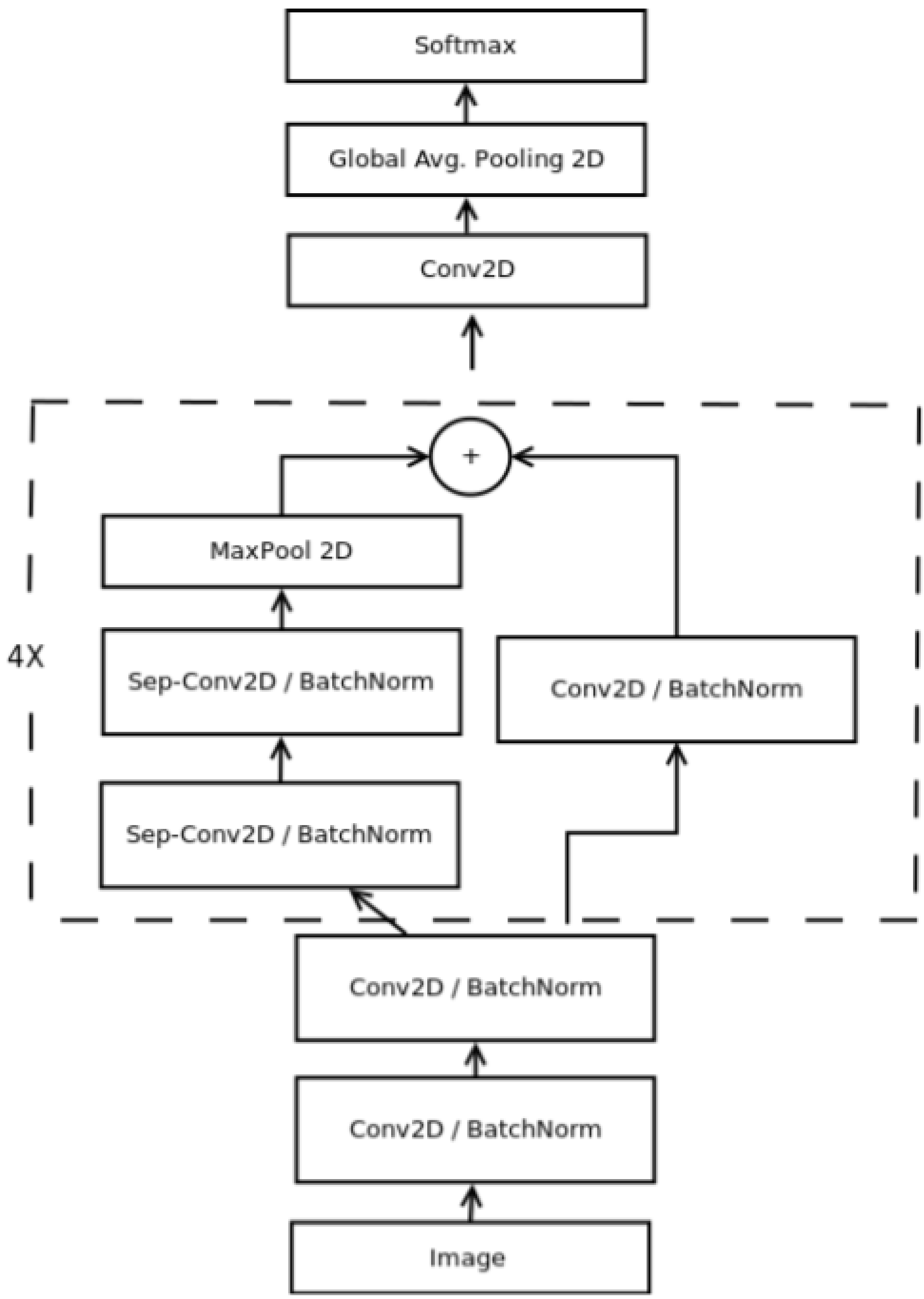


Fig. 3: Our proposed model for real-time classification.

Since our initial proposed architecture deleted the last fully connected layer, we reduced further the amount of parameters by eliminating them now from the convolutional layers. This was done trough the use of depth-wise separable convolutions. Depth-wise separable convolutions are composed of two different layers: depth-wise convolutions and pointwise convolutions. The main purpose of these layers is to separate the spatial cross-correlations from the channel crosscorrelations [1]. They do this by first applying a $D \times D$ filter on every M input channels and then applying N $1 \times 1 \times M$ convolution filters to combine the M input channels into N output channels. Applying $1 \times 1 \times M$ convolutions combines each value in the feature map without considering their spatial relation within the channel. Depth-wise separable convolutions reduces the computation with respect to the standard convolutions by a factor of 1 N + 1 D2 [2]. A visualization of the difference between a normal Convolution layer and a depth-wise separable convolution can be observed in Figure 4. Our final architecture is a fully-convolutional neural network that contains 4 residual depth-wise separable convolutions where each convolution is followed by a batch normalization operation and a ReLU activation function. The last layer applies a global average pooling and a soft-max activation function to produce a prediction. This architecture has approximately 60, 000 parameters; which corresponds to a reduction of 10× when compared to our initial naive implementation, and 80× when compared to the original CNN. Figure 3 displays our complete final architecture which we refer to as mini-Xception. This architectures obtains an accuracy of 95% in gender classification task. Which corresponds to a reduction of one percent with respect to our initial implementation. Furthermore, we tested this architecture in the FER-2013 dataset and we obtained the same accuracy of 66% for the emotion classification task. Our final architecture weights can be stored in an 855 kilobytes file. By reducing our architectures computational cost we are now able to join both models and use them consecutively in the same image without any serious time reduction. Our complete pipeline including the openCV face detection module, the gender classification and the emotion classification takes 0.22 ± 0.0003 ms on a i5-4210M CPU. This corresponds to a speedup of 1.5× when compared to the original architecture of Tang. We also added to our implementation a real-time guided back-propagation visualization to observe which pixels in the image activate an element of a higher-level feature map. Given a CNN with only ReLUs as activation functions for the intermediate layers, guided-back propagation takes the derivative of every element (x, y) of the input image I with respect to an element (i, j) of the feature map f L in layer L. The reconstructed image R filters all the negative gradients; consequently, the remaining gradients are chosen such that they only increase the value of the chosen element of the feature map. Following [11], a fully ReLU CNN reconstructed image in layer l is given by:

$$R^{l}_{i,j} = (R^{l+1}_{i,j} > 0) * R^{l+1}_{i,j}$$

Challenges in face detection:-

• Challenges in face detection, are the reasons which reduce the accuracy and detection rateof face detection. These challenges are complex background, too many faces in images, odd
expressions, illuminations, less resolution, face occlusion, skin color, distance and orientation
etc. (Figure 3).
• Odd expressions Human face in an image may have odd expressions unlike normal, which
is challenge for face detection.

• Face occlusion Face occlusion is hiding face by any object. It may be glasses, scarf, hand,
hairs, hats and any other object etc. It also reduces the face detection rate.

• Illuminations Lighting effects may not be uniform in the image. Some part of the image may have very high illumination and other may have very low illumination.

• Complex background Complex background means a lot of objects presents in the image,
which reduces the accuracy and rate of face detection.

• Too many faces in the image It means image contains too many human faces, which is challenge for face detection.
• Less resolution Resolution of image may be very poor, which is also challenging for face
detection.
• Skin color Skin-color changes with geographical locations. Skin color of Chinese is different from African and skin-color of African is different from American and so on. Changing
skin-color is also challenging for face detection.

• Distance Too much distance between camera and human face may reduce the detection rate of human faces in image.

• Orientation Face orientation is the pose of face with an angle. It also reduces the accuracy
and detection rate of face detection.

Applications of face detection system

• Gender classification Gender information can be found from human being image.
• Document control and access control Control can be imposed to document access withface identification system.

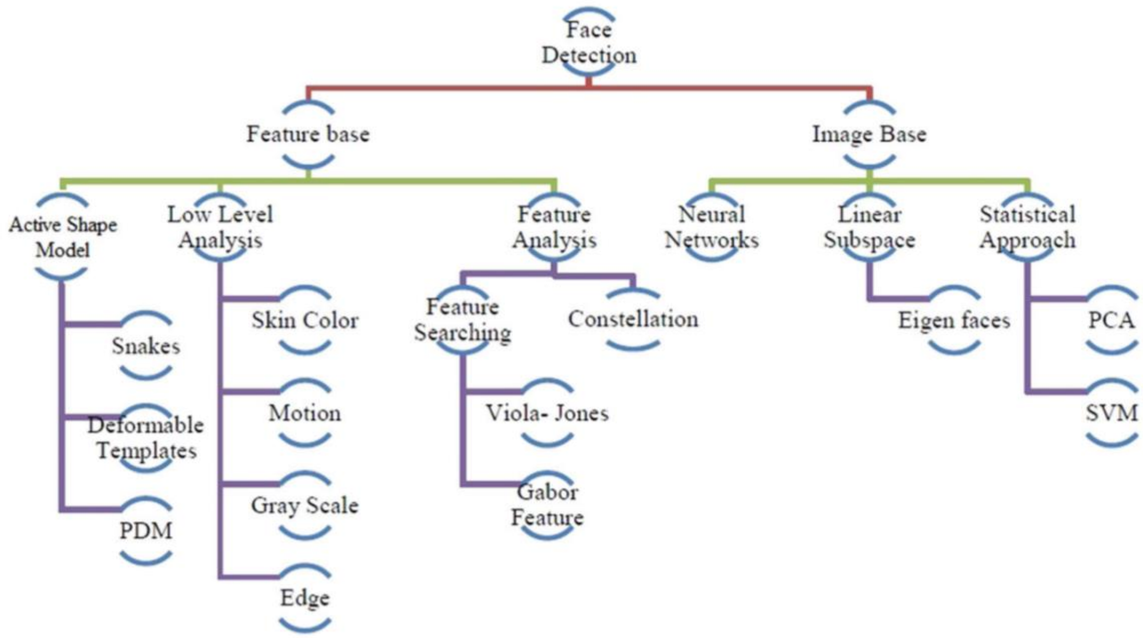
• Human computer interaction system It is design and use of computer technology, focusing
particularly on the interfaces between users and computers.

• Biometric attendance It is system of taking attendance of people by their finger prints or
face etc.
• Photography Some recent digital cameras use face detection for autofocus. Face detection
is also useful for selecting regions of interest in photo slideshows.

• Facial feature extraction Facial features like nose, eyes, mouth, skin-color etc. can be
extracted from image.

• Face recognition A facial recognition system is a process of identifying or verifying a
person from a digital image or a video frame. One of the ways to do this is by comparingselected facial features from the image and a facial database. It is typically used in security
systems.

• Marketing Face detection is gaining the interest of marketers. A webcam can be integratedinto a television and detect any face that walks by. The system then calculates the race, gender, and age range of the face. Once the information is collected, a series of advertisements
can be played that is specific towards the detected race/gender/age



Available facial recognition APIs :-

Kairos Offers a side variety of image recognition solutions through their API. Their API endpoints include identifying gender, age, emotional depth, facial recognition in both photo and video, and more.

• Trueface.ai One flaw with some facial recognition APIs is that they are unable to differ entiate between a face and a picture of a face. TrueFace.ai solves that problem with their ability to do spoof detection through their API.

• Amazon recognition This facial recognition API is fully integrated into the Amazon Web Service ecosystem. Using this API will make it really easy to build applications that make use of other AWS products.

• Face recognition and face detection by Lambda Labs With over 1000 calls per month in the free pricing tier, and only $0.0024 per extra API call, this API is a really affordable option for developers wanting to use a facial recognition API.

• EmoVu by Eyeris This API was created by Eyeris and it is a deep learning-based emotion recognition API. EmoVu allows for great emotion recognition results by identifying facial micro-expressions in real-time.

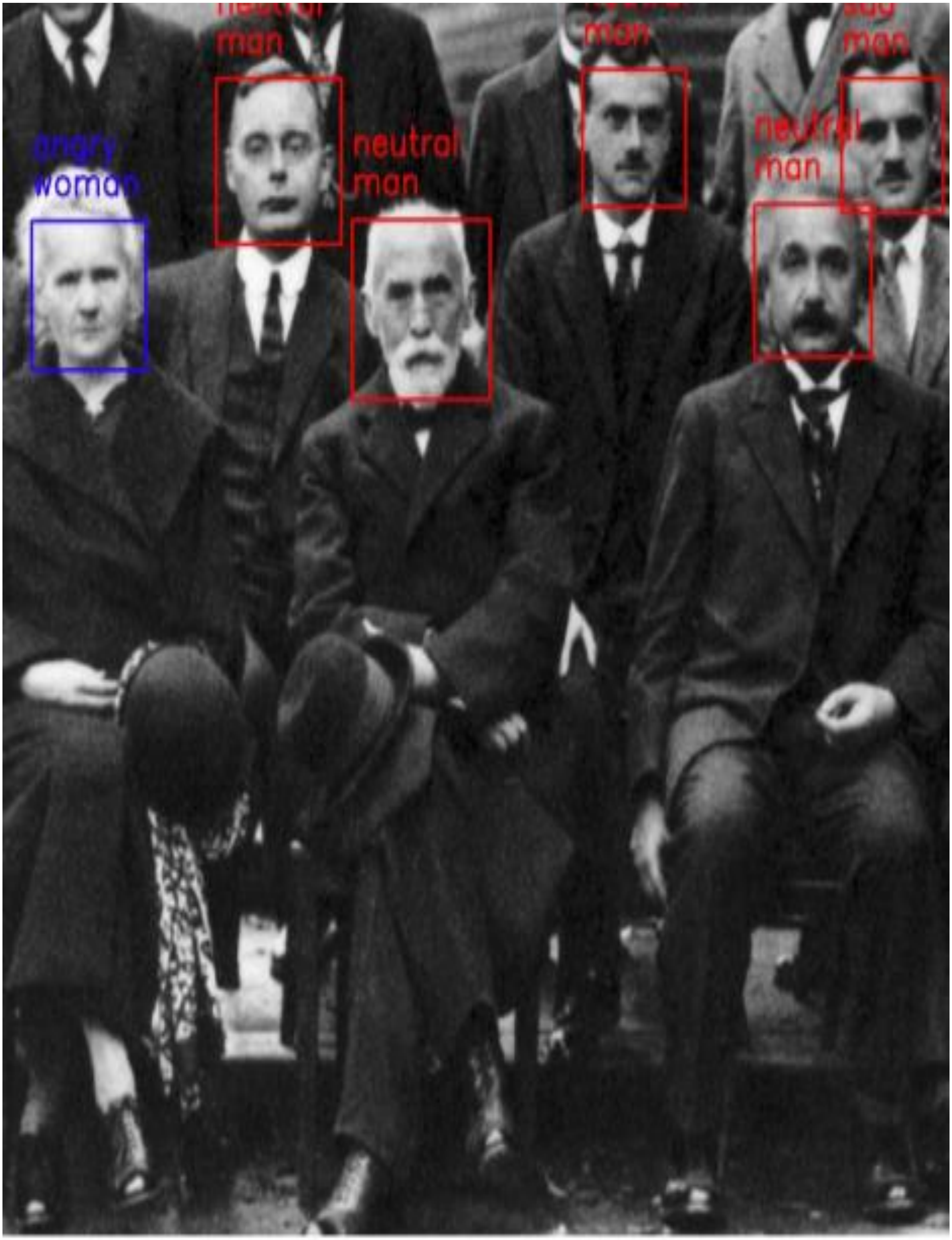
• Microsoft face API One cool feature that I found while doing research on the Microsoft Face API, is that the API has the ability to do “similar face search.” When this API endpoint is given a collection of faces, and a new face as a query, the API will return a collection of similar faces from the collection.

• Animetrics face recognition Using advanced 2D-to-3D algorithms, this API will convert a 2D image into a 3D model. The 3D model will then be used for facial recognition purpose.

• Face++ This API also has an offline SDK for iOS and Android for you to use. The offline SDK does not provide face recognition, but it can perform face detection, comparing, tracking and landmarks, all while the phone does not have cell service.

• Google cloud vision By being integrated into the Google Cloud Platform, this API will be a breeze for you to integrate into applications that are already using other Google Cloud Platform products and services.

• IBM Watson visual recognition Whether it is faces, objects, colors, or food, this API lets you identify many different types of classifiers. If the included classifiers aren’t enough, then you can train and use your own custom classifiers.


man
man
sad
man
angry
woman
neutral
man
neutral
man

Facial Recognition

Faces are high dimensionality dataconsisting of a number of pixels .Datain high dimensionality is difficult to process and can not be visualized using simplet data.

Letusjumptothecodingsegment!

```
import face_recognition as fr
import cv2
import numpy as np
import os

path = "./train/"

known_names = []
known_name_encodings = []

images = os.listdir(path)
for _ in images:
    image = fr.load_image_file(path + _)
    image_path = path + _
    encoding = fr.face_encodings(image)[0]

    known_name_encodings.append(encoding)
    known_names.append(os.path.splitext(os.path.basename(image_path))[0].capitalize())

print(known_names)

test_image = "./test/test.jpg"
image = cv2.imread(test_image)
# image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

face_locations = fr.face_locations(image)
face_encodings = fr.face_encodings(image, face_locations)

for (top, right, bottom, left), face_encoding in zip(face_locations, face_encodings):
    matches = fr.compare_faces(known_name_encodings, face_encoding)
    name = ""

    face_distances = fr.face_distance(known_name_encodings, face_encoding)
    best_match = np.argmin(face_distances)

    if matches[best_match]:
        name = known_names[best_match]

    cv2.rectangle(image, (left, top), (right, bottom), (0, 0, 255), 2)
    cv2.rectangle(image, (left, bottom - 15), (right, bottom), (0, 0, 255), cv2.FILLED)
    font = cv2.FONT_HERSHEY_DUPLEX
    cv2.putText(image, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)


cv2.imshow("Result", image)
cv2.imwrite("./output.jpg", image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

The following code example is taken from thes klearnd ocumentationoneigenfaces. We will go through the code step by step to understand itsintricaciesandresults.WhatwewilldoistousePCAtoreducethehighdimensionalityof dataandthenfeeditintotheSVMclassifiertoclassifythepictures. ImportRelevantLibrariesandModulesFirstoff,wewillimporttherequiredlibrariesandmodules.Anin-depthdiscussion of why we are importing them will follow when their needsarise.

LITERATURE REVIEW

A face recognition system using Eigenface system was proposed by Dhavalsinh to cover the attendance of the scholars, where the face acts as the main indicator. Eigenface is a set of eigenvectors used in face recognition and discovery. it's used to determine the variation among multiple faces by performing a statistical analysis on the facial images. Sirovich and Kirby designed the Eigenfaces approach to do facial recognition and the same was used by Matthew Turk and Alex Pentland for face bracket. Kranthikiran and Pulicherla (8) made use of Eigenfaces and Star Element Analysis (PCA) to perform face discovery for lot surveillance.

Nonstop face biometric recognition has been used by Fayyoumi and Zarrad in developing a prototype for conducting online examinations. The prototype has been estimated by carrying feedbacks from different experts through a check using a five- point Likert scale. The proposed system contains a question bank to help the preceptors in generating different tests aimlessly. Kamencay etal. suggested a face recognition system using Convolutional Neural Network (CNN). The authors used OLR dataset comprising 400 different realities (40 orders/ 10 images for every order) to carry out the trials and validate their results. The discovery delicacy of the suggested system has been compared with the three popular image recognition approaches like PCA, Original Double Patterns Histograms (LBPH) and KNN. In comparison with these styles, the proposed CNN-grounded system performs better by achieving an identification delicacy of about98.3.

Traoré etal. used a multimodal biometric frame to authenticate the actors in online examinations. The frame consists of three modalities similar as mouse dynamics, keystroke dynamics, and face biometrics to check the authenticity. This frame has been included as a module in ExamShield which is an online test monitoring tool. Nonstop face biometric recognition has been used by Fayyoumi and Zarrad in developing a prototype for conducting online examinations. The prototype has been estimated by carrying feedbacks from different experts through a check using a five- point Likert scale. The proposed system contains a question bank to help the preceptors in generating different tests aimlessly.

Zhu etal. addressed colorful challenges in face discovery systems by developing a new approach videlicet ContextualMulti-Scale Region- grounded Complication Neural Network (CMS-RCNN) which consists of two factors 1) region offer element and) the region-of- interest (RoI) discovery element. The proposed system deals with bitsy face regions by groupingmulti-scale information in both the factors and also allows unequivocal body contextual logic.

Zehenguo Yuan addresses the facial occlusion and bettered the discovery delicacy by developing a visual attention guidance model that guides in pressing the visible area in an clotted face. This model avoids setting the fresh parameters by using an activation chart that predicts the position and scale of the face

The below styles are the bones that are presently available for face recognition. But there's a need for a briskly and dependable system that could be used for covering the scholars during their examinations. In this design, an approach analogous to Eigenface is used for rooting facial features through facial vectors and the datasets are trained using an Support Vector Machine (SVM) model which is one of the popular machine learning algorithms. This ensures that the face recognition can be briskly and can be fluently used.

Cascade-based:

These methods extend the Viola and Jones detector cascade. For example, [41] proposed to train a detector cascade for each view of the face and combined their results at the test time. Recently, [22] combined this method with integral channel features [3] and soft-cascade [1], and showed that by using 22 cascades, it is possible to obtain state-of-the-art performance for multi-view face detection. This approach, however, requires face orientation annotations. Moreover its complexity in training and testing increases linearly with the number of models. To address the computational complexity issue, Viola and Jones [39] proposed to first estimate the face pose using a tree classifier and then run the cascade of corresponding face pose to verify the detection. While improving the detection speed, this method degrades the accuracy because mistakes of the initial tree classifier are irreversible. This method is further improved by [13, 12] where, instead of one detector cascade, several detectors are used after the initial classifier. Finally, [35] and [28] combined detector cascade with multiclass boosting and proposed a method for multiclass/multi-view object detection.

- **DPM-based:**

These methods are based on the deformable part models technique [5] where a face is defined as a collection of its parts. The parts are defined via unsupervised or supervised training, and a classifier, latent SVM, is trained to find those parts and their geometric relationship. These detectors are robust to partial occlusion because they can detect faces even when some of the parts are not present. These methods are, however, computationally intensive because 1) they require solving a latent SVM for each candidate location and 2) multiple DPMs have to be trained and combined to achieve the state-of-the-art performance [22, 25]. Moreover, in some cases DPM-based models require annotation of facial landmarks for training, e.g [25].

- **Neural-Network-based:**

There is a long history of using neural networks for the task of face detection [38, 37, 27, 8, 7, 6, 26, 11, 24, 23]. In particular, [38] trained a two-stage system based on convolutional neural networks. The first network locates rough positions of faces and the second network verifies the detection and makes more accurate localization. In [27], the authors trained multiple face detection networks and combined their output to improve the performance. [8] trained a single multi-layer network for face detection. The trained network is able to partially handle different poses and rotation angles. More recently, [23] proposed to train a neural network jointly for face detection and pose estimation. They showed that this joint learning scheme can significantly improve performance of both detection and pose estimation. Our method follows the works in [8, 23] but constructs a deeper CNN for face detection.

The key challenge in multi-view face detection, as pointed out by Viola and Jones [39], is that learning algorithms such as Boosting or SVM and image features such as HOG or Haar wavelets are not strong enough to capture faces of different poses and thus the resulted classifiers are hopelessly inaccurate. However, with recent advances in deep learning and GPU computation, it is possible to utilize the high capacity of deep convolutional neural networks for feature extraction/classification, and train a single model for the task of multi-view face detection. Deep convolutional neural network has recently demonstrated outstanding performance in a variety of vision tasks such as face recognition [34, 30], object classification [19, 31], and object detection [9, 29, 18, 32]. In particular [19] trained an 8-layered network, called AlexNet, and showed that deep convolutional neural networks can significantly outperform other methods for the task of large scale image classification. For the task of object detection, [9] proposed R-CNN method that uses an image segmentation technique, selective search [36], to find candidate image regions and classify those candidates using a version of AlexNet that is finetuned for objects in the PASCAL VOC dataset. More recently, [33] improved R-CNN by 1) augmenting the selective search proposals with candidate regions from multibox approach [4], and 2) replacing 8-layered AlexNet with a much deeper CNN model of GoogLeNet [31]. Despite state-of-theart performance, these methods are computationally suboptimal because they require evaluating a CNN over more than 2, 000 overlapping candidate regions independently. To address this issue, [18] recently proposed to run the CNN model on the full image once and create a feature pyramid. The candidate regions, obtained by selective search, are then mapped into this feature pyramid space. [18] then uses spatial pyramid pooling [20] and SVM on the mapped regions to classify candidate proposals. Beyond region-based methods, deep convolutional neural networks have also been used with sliding window approach, e.g. OverFeat [29] and deformable part models [10] for object detection and [17] for human pose estimation. In general, for object detection these methods still have an inferior performance compared to region-based methods such as R-CNN [9] and [33]. However, in our face detection experiments we found that the region-based methods are often very slow and result in relatively weak performance

We start by fine-tuning AlexNet [19] for face detection. For this we extracted training examples from the AFLW dataset [21], which consists of 21K images with 24K face annotations. To increase the number of positive examples, we randomly sampled sub-windows of the images and used them as positive examples if they had more than a 50% IOU (intersection over union) with the ground truth. For further data augmentation, we also randomly flipped these training examples. This resulted in a total number of 200K positive and and 20 millions negative training examples. These examples were then resized to 227 × 227 and used to finetune a pre-trained AlexNet model [19]. For fine-tuning, we used 50K iterations and batch size of 128 images, where each batch contained 32 positive and 96 negative examples. Using this fine-tuned deep network, it is possible to take either region-based or sliding window approaches to obtain the final face detector. In this work we selected a sliding window approach because it has less complexity and is independent of extra modules such as selective search. Also, as discussed in the experiment section, this approach leads to better results as compared to R-CNN. Our face classifier, similar to AlexNet [19], consists of 8 layers where the first 5 layers are convolutional and the last 3 layers are fully-connected. We first converted the fullyconnected layers into convolutional layers by reshaping layer parameters [14]. This made it possible to efficiently run the CNN on images of any size and obtain a heat-map of the face classifier. An example of a heat-map is shown in Figure 2- right. Each point in the heat-map shows the CNN response, the probability of having a face, for its corresponding 227×227 region in the original image. The detected regions were then processed by non-maximal suppression to accurately localize the faces. Finally, to detect faces of different sizes, we scaled the images up/down and obtained new heat-maps. We tried different scaling schemes and found that rescaling image 3 times per octave gives reasonably good performance. This is interesting as many of the other methods such as [22, 2] requires a significantly larger number of resizing per octave, e.g. 8. Note that, unlike R-CNN [9], which uses SVM classifier to obtain the final score, we removed the SVM module and found that the network output are informative enough for the task of face detection. Face localization can be further improved by using a boundingbox regression module similar to [29, 9]. In our experiment, however, adding this module degraded the performance. Therefore, compared to the other methods such as R-CNN [9], which uses selective search, SVM and boundingbox regression, or DenseNet [10], which is based on the deformable part models, our proposed method (DDFD) is fairly simple. Despite its simplicity, as shown in the experiments section, DDFD can achieve state-of-the-art performance for face detection

2.1 Detector Analysis

In this section, we look into the scores of the proposed face detector and observe that there seems to be a correlation between those scores and the distribution of positive examples in the training set. We can later use this hypothesis to obtain better training set or to design better data augmentation procedures and improve performance of DDFD. We begin by running our detector on a variety of faces with different in-plane and out-of-plane rotations, occlusions and lighting conditions (see for example Figure 1, Figure 2- left and Figure 3). First, note that in all cases our detector is able to detect the faces except for the two highly occluded ones in Figure 1. Second, for almost all of the detected faces, the detector's confidence score is pretty high, close to 1. Also as shown in the heat-map of Figure 2-right, the scores are close to zero for all other regions. This shows that DDFD has very strong discriminative power, and its output can be used directly without any post-processing steps such as SVM, which is used in R-CNN [9]. Third, if we compare the detector scores for faces in Figure 2-left, it is clear that the up-right frontal face in the bottom has a very high score of 0.999 while faces with more in-plane rotation have less score. Note that these scores are output of a sigmoid function, i.e. probability (soft-max) layer in the CNN, and thus small changes in them reflects much larger changes in the output of the previous layer. It is interesting to see that the scores decrease as the in-plane rotation increases. We can see the same trend for out-of-plane rotated faces and occluded faces in Figures 1 and 3. We hypothesize that this trend in the scores is not because detecting rotated face are more difficult but it is because of lack of good training examples to represent such faces in the training process. To examine this hypothesis, we looked into the face annotations for AFLW dataset [21]. Figure 4 shows the distribution of the annotated faces with regards to their in-plane, pitch (up and down) and yaw (left to right) rotations. As shown in this figure, the number of faces with more than 30 degrees out-of-plane rotation is significantly lower than the faces with less than 30 degree rotation. Similarly, the number of faces with yaw or pitch less than 50 degree is significantly larger than the other ones. Given this skewed training set, it not surprising that the fine-tuned CNN is more confident about up-right faces. This is because the CNN is trained to minimize the risk of the soft-max loss function

$$R = X\ x_i \in B \log [prob(y_i|x_i)] ,$$

where B is the example batch that is used in an iteration of stochastic gradient descent and y_i is the label of example x_i.

The sampling method for selecting examples in B can significantly hurt performance of the final detector. In an extreme case if B never contains any example of a certain class, the CNN classifier will never learn the attributes of that class. In our implementation $|B| = 128$ and it is collected by randomly sampling the training set. However, since the number of negative examples are 100 times more than the number of positive examples, a uniform sampling will result in only about 2 positive examples per batch. This significantly degrades the chance of the CNN to distinguish faces from nonfaces. To address this issue, we enforced one quarter of each batch to be positive examples, where the positive examples are uniformly sampled from the pool of positive training samples. But, as illustrated in Figure 4, this pool is highly skewed in different aspects, e.g. in-plane and out-of-plane rotations. The CNN is therefore getting exposed with more up-right faces; it is thus not surprising that the fine-tuned CNN is more confident about the up-right faces than the rotated ones. This analysis suggests that the key for improving performance of DDFD is to ensure that all categories of the training examples have similar chances to contribute in optimizing the CNN. This can be accomplished by enforcing population-based sampling strategies such as increasing selection probability for categories with low population

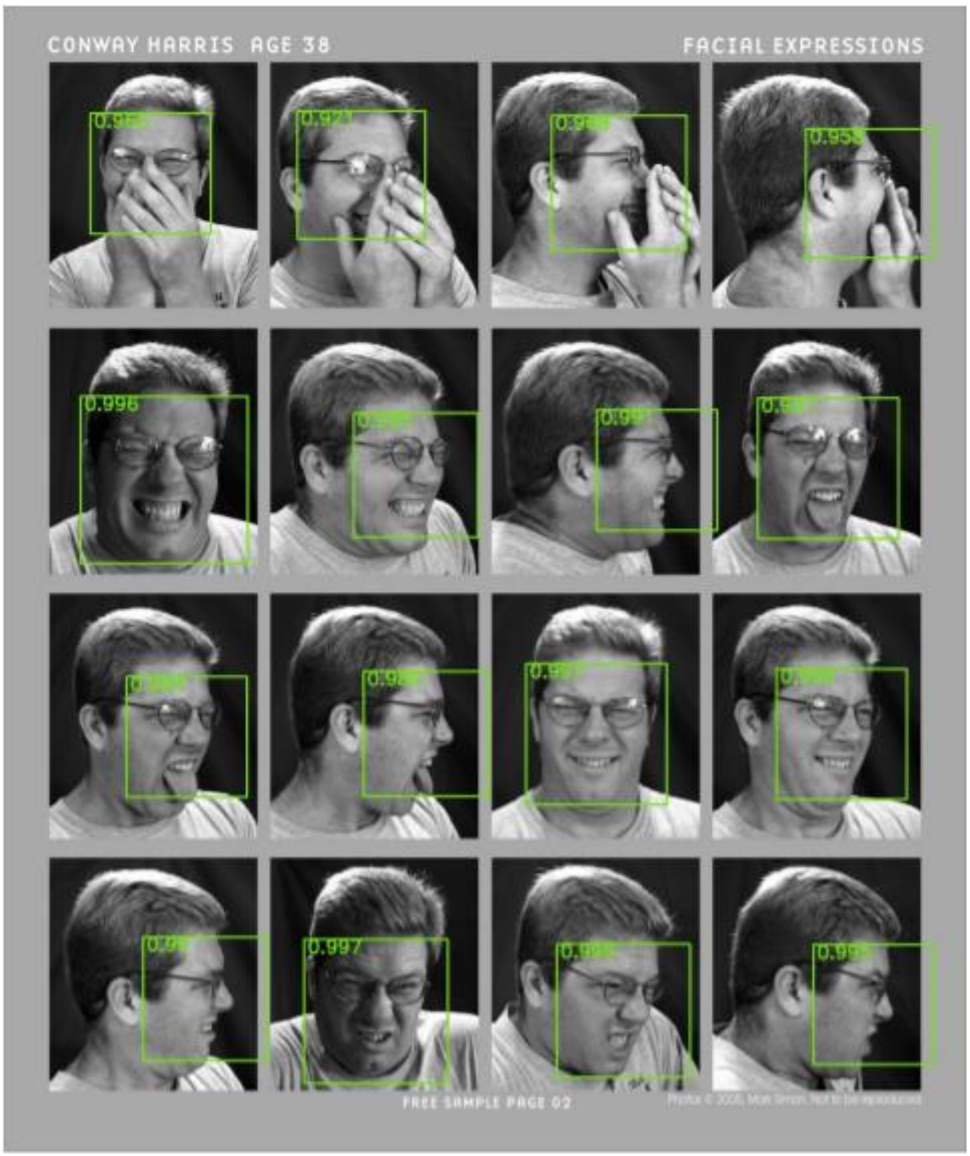
Similarly, the current face detector still fails to detect faces with heavy occlusions. Similar to the issue with rotated faces, we believe that this problem can also be addressed through modification of the training set. In fact, most of the face images in the AFLW dataset [21] are not occluded, which makes it difficult for a CNN to learn that faces can be occluded. This issue can be addressed by using more sophisticated data augmentation techniques such as occluding parts of positive examples. Note that simply covering parts of positive examples with black/white or noise blocks is not useful as the CNN may learn those artificial patterns.

To summarize, the proposed face detector based on deep CNN is able to detect faces from different angles and handle occlusion to some extent. However, since the training set is skewed, the network is more confident about up-right faces and better results can be achieved by using better sampling strategies and more sophisticated data augmentation techniques

We implemented the proposed face detector using the Caffe library [16] and used its pre-trained Alexnet [19] model for fine-tuning. For further details on the training process of our proposed face detector please see section 2. After converting fully-connected layers to convolutional layers [14], it is possible to get the network response (heat-map) for the whole input image in one call to Caffe code. The heat-map shows the scores of the CNN for every 227×227 window with a stride of 32 pixels in the original image. We directly used this response for classifying a window as face or background. To detect faces of smaller or larger than 227×227, we scaled the image up or down respectively.

we scaled the image up or down respectively. We tested our face detection approach on PASCAL Face [42], AFW [25] and FDDB [15] datasets. For selecting and tuning parameters of the proposed face detector we used the PASCAL Face dataset. PASCAL Face dataset consists of 851 images and 1341 annotated faces, where annotated faces can be as small as 35 pixels. AFW dataset is built using Flickr images. It has 205 images with 473 annotated faces, and its images tend to contain cluttered background with large variations in both face viewpoint and appearance (aging, sunglasses, make-ups, skin color, expression etc.). Similarly, FDDB dataset [15] consists of 5171 annotated faces with 2846 images and contains occluded, out-of-focus, and low resolution faces. For evaluation, we used the toolbox provide by [22] with corrected annotations for PASCAL Face and AFW datasets and the original annotations of FDDB dataset.
We started by finding the optimal number of scales for the proposed detector using PASCAL dataset. We upscaled images by factor of 5 to detect faces as small as $227/5 = 45$ pixels. We then down scaled the image with by a factor, fs, and repeated the process until the minimum image dimension is less than 227 pixels. For the choice of fs, we chose $fs \in \{\sqrt{0.5} = 0.7071, \sqrt[3]{0.5} = 0.7937, \sqrt[5]{0.5} = 0.8706, \sqrt[7]{0.5} = 0.9056\}$; Figure 5 shows the

effect of this parameter on the precision and recall of our face detector (DDFD). Decreasing fs allows the detector to scan the image finer and increases the computational time. According to Figure 5, it seems that these choices of fs has little impact on the performance of the detector. Surprisingly, fs = √3 0.5 seems to have slightly better performance although it does



not scan the image as thorough as fs = √5 0.5 or fs = √7 0.5. Based on this experiment we use fs = √3 0.5 for the rest of this paper. Another component of our system is the non-maximum suppression module (NMS). For this we evaluated two different strategies:

• **NMS-max:**

we find the window of the maximum score and remove all of the bounding-boxes with an IOU (intersection over union) larger than an overlap threshold.

• **NMS-avg:**
we first filter out windows with confidence lower than 0.2. We then use groupRectangles function of OpenCV to cluster the detected windows according to an overlap threshold. Within each cluster, we then removed all windows with score less than 90% of the maximum score of that cluster. Next we averaged the locations of the remaining bounding-boxes to get the detection window. Finally, we used the maximum score of the cluster as the score of the proposed detection.

We tested both strategies and Figure 6 shows the performance of each strategy for different overlap thresholds. As shown in this figure, performance of both methods vary significantly with the overlap threshold. An overlap threshold of 0.3 gives the best performance for NMS-max while, for NMS-avg 0.2 performs the best. According to this figure, NMS-avg has better performance compared to NMS-max in terms of average precision.

Finally, we examined the effect of a bounding-box regression module for improving detector localization. The idea is to train regressors to predict the difference between the locations of the predicted bounding-box and the ground truth. At the test time these regressors can be used to estimate the location difference and adjust the predicted bounding-boxes accordingly. This idea has been shown to improve localization performance in several methods including [5, 29, 4]. To train our bounding-box regressors.

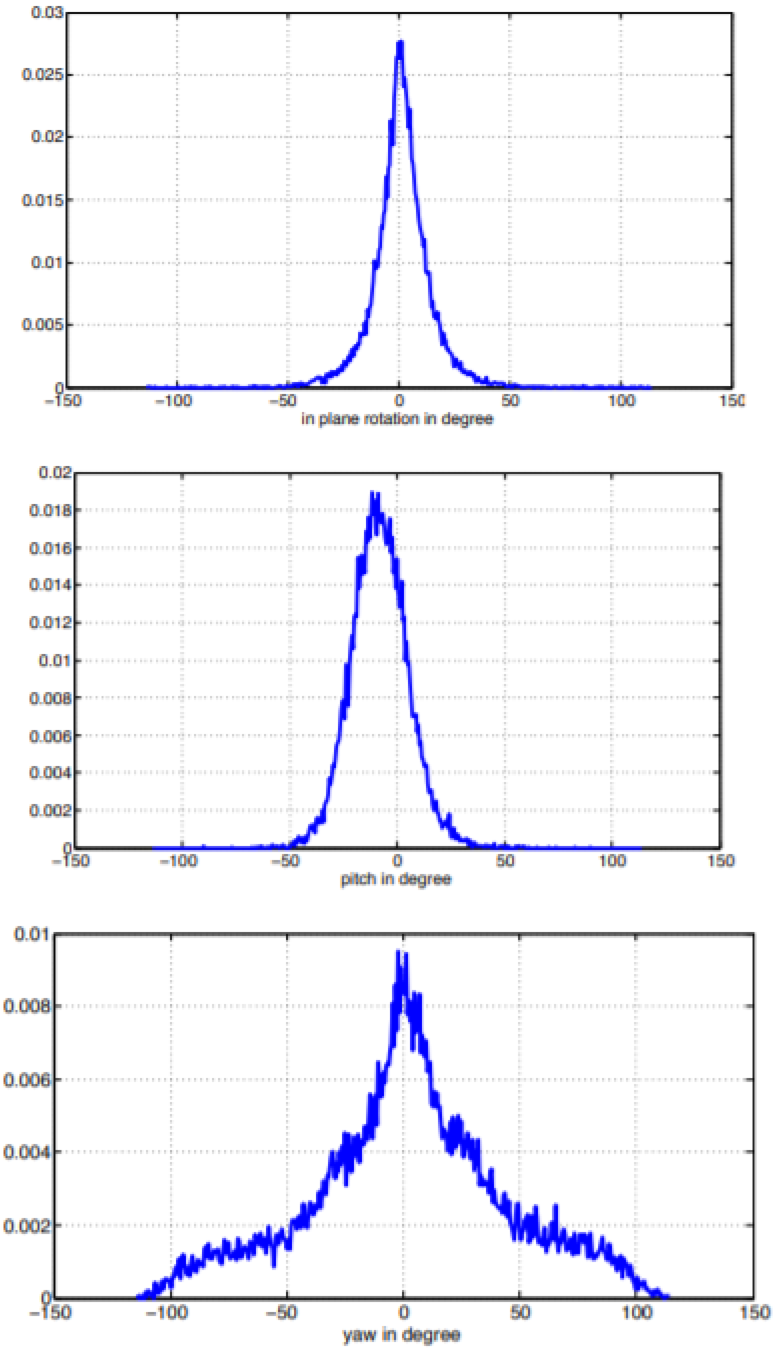


Figure 4: Histogram of faces in AFLW dataset based on their top) in-plane, middle) pitch (up and down) and bottom) yaw(left to right) rotations.

we followed the algorithm of shows the performance of our detector with and without this module. As shown in this figure, surprisingly, adding a bounding-box regressors degrades the performance for both NMS strategies. Our analysis revealed that this is due to the mismatch between the annotations of the training set and the test set. This mismatch is mostly for side-view faces and is illustrated in Figure 8. In addition to degrading performance of bounding-box regression module, this mismatch also leads to false miss-detections in the evaluation process.\

Face Detection and Recognition under Face Occlusion

In addition, in face detection, the situation that the face is obstructed by the obstruction obscuring the true appearance of the detector exists in the market, so there is great research significance for face detection and recognition in the case of face occlusion.

As can be seen from Figure 13, when the detector's eyes are occluded, face detection is completely impossible. When the nose and mouth are blocked, the five-point positioning can be clearly performed, and the portrait and background can be divided more accurately.

The different sample libraries and test libraries of ORL, AR, Yale-B, and CAS-PEAL-R1 are, respectively, cropped to pixels. In addition, the posture of the training library is manually corrected, and the average value is selected for multiple tests. Experiment 1 compares the recognition rates of Algorithms 1, 2, and 3 to verify the importance of multichannel weighted representation. The experimental comparison is shown in Table 11.

Table 11

The experimental comparison.

Compared	Algorithm 1: best single-channel Gabor characterization	Algorithm 2: multichannel Gabor characterization with equal weight	Algorithm 3: weighted multichannel Gabor characterization
ORL	0.86	0.915	0.95
AR	0.77333	0.89167	0.90167
Yale-B	0.6	0.6	0.66667

It is proved that it is more effective to extract the features on the salient face area. The experimental comparison is shown in Table 12.

Table 12

Extract the features on the salient face area.

Compared	Algorithm 4: best single-channel Gabor characterization	Algorithm 5: multichannel Gabor characterization with equal weight	Algorithm 6: weighted multichannel Gabor characterization
ORL	0.95	0.96	0.915
AR	0.90167	0.95333	0.90167
Yale-B	0.66667	0.5	0.5

As shown in Figure 14, no matter whether the eyes, nose, or mouth are blocked, the YouTu method can accurately locate 90 feature points for the detection of facial contours. This method is highly effective for face detection.

As can be seen from Figure 15, the face can be effectively separated from the background only when the nose is blocked, and face recognition cannot be performed when the eyes and mouth are blocked.

Compare the recognition rate of the methods in this chapter, as shown in Table 13.

Table 13

Compare the recognition rate of the methods in this chapter.

Compared	Algorithm 4: best single-channel Gabor characterization	Algorithm 6: multichannel Gabor characterization with equal weight	Algorithm 7: weighted multichannel Gabor characterization based on region selection and FFT
ORL	0.95	0.975	0.975
AR	0.90167	0.90333	0.95333
Yale-B	0.66667	0.5	0.5

Test the sensitivity of the algorithm in this research to noise. Usually, the picture may contain various noises. After adding several common noises to the test picture, compare the method in this chapter with the traditional overall Gabor characterization. The recognition rate is shown in Table 14.

Table 14

Test the sensitivity of the algorithm.

AR-600	Algorithm 5: overall Gabor characterization	Algorithm 7: weighted multichannel Gabor characterization based on region selection and FFT
Salt and pepper noise	0.69	0.95333
Gaussian white noise	0.865	0.7133
Speckle noise	0.7566	0.89833
Poisson noise	0.85833	0.77833

The facial expression changes are extremely diverse. Each person's expression has the uniqueness and uniqueness of his own posture. This complex expression is extremely difficult in face detection and recognition. It is extremely important to find a more effective way to locate various expression changes.

It can be concluded from Figure 16 that the Seetaface method can effectively and accurately distinguish the face from the background and accurately perform the facial features in any expression state of surprise, anger, and crying.

As seen in Figure 17, the YouTu method can also perform face contour segmentation in three different expressions, but it can accurately segment the face and background. However, from a precise point of view, the facial features of the method are slightly deviated, resulting in a slight accuracy decline.

As can be seen from Figure 18, OpenCV can perform face recognition and detection for different expressions, but this method can only be used for face recognition, and accurate five-position positioning cannot be performed in face positioning.

PROPOSED SYSTEM

Educational institutions conduct online examinations for a large number of people and it's delicate to cover the scholars manually. The proposed system focuses on designing a suitable face discovery and recognition model for covering the scholars during online examinations. Then, Eigenface system is used for rooting the facial features through facial vectors and the datasets are trained using an SVM model to ameliorate the discovery delicacy. The inflow illustration for the proposed system is given in Figure 1. At first, the vector values are uprooted from the images in the dataset. The uprooted embeddings are also passed to the SVM classifier to train it for feting the faces from the input images or videotape frames.

FACILITIESREQUIRED

Software requirements:

1. WindowsOS
2. Mysql
3. VisualstudioHardwarerequirements:
1. Processor-DualCore
2. Memory-1GB ram
3. InternetConnection
4. Webcam

CONCLUSION

In this paper, A machine learning based face detection and recognition system using SVM model is proposed to detect the faces of students for monitoring their activities during online examinations. The proposed systemaids in detecting the faces in a faster manner by obtaining feature vectorsfrom the input images. Several algorithms such as LBPH, Fisher faces, SIFT and SURF can also be applied along with this method to build more efficient recognition models that can detect faces in varying illuminations
and light intensities. Still better optimal values can also be obtained by applying different algorithms. Higher accuracy can be obtained using convolutional neural networks.