PROFILE IDENTIFICATION WITH ENGROSSMENT SPAN DETECTION

Anthony M.
Department of Computer Science
Jeppiaar Engineering College
Chennai, India

Arun Kumar P.
Department of Computer Science
Jeppiaar Engineering College
Chennai, India

Deepak Sharran V.
Department of Computer Science
Jeppiaar Engineering College
Chennai, India

M.Sasi Kumar, ME.,
Department of Computer Science
Jeppiaar Engineering College
Chennai, India

ABSTRACT

Organisations provide job and security to its employees and also provide supporting financial services . Employees spent 80% of their time in day at their offices working for the organisation and improving it by their hard work . When it comes to an employee's career progression there is no solid data on their performance , naturally their time and hard work goes unnoticed most of the time .An employee's career progression for now depends mostly on their superior staff and their personal evaluation on that particular employee, which causes stress and pressure on some levels on the employee. To have solid data on a particular employee's performance is to determine where the employee stands in the career ladder of progression . This can be done with an intermediate-level Python project , to record the performance of an employee , by using facial recognition, retinal displacement, drowsiness detection , off screen time in a span of time . An alert message would pop up if the A.I determines that the employee is not concentrated on the task at hand . It will be impossible for a human being to be dedicated 100% the whole time .

I INTRODUCTION

Profile identification , based on Machine Learning, aims to recognise a personnel functioning under an Organisation and identifying their personal information such as name, organisational id through distinguishing their facial features . Engrossment span detection , by constant observation of the personnel and identifying wavering actions from the given task through a video device such as a webcam. Such as retinal displacement from screen for a longer period , drowsiness , moving out of screen for a longer period .The project focuses on providing an interface for the employer to monitor and keep track of the progress of their employees.The employer thus can know the status of their employees on specific projects . The employee's performance is now directly noted and makes their hard work pay off.The employees need not worry about the human relation factor for their promotions,bonuses. To analyze the performance of an employee by calculating the factors of disturbances and off-screen time To make things easier for the employer to track the stats and progress of an employee without taking much time and hussle .

II LITERATURE REVIEW

[1] Gang Pan, Lin Sun, Zhaohui Wu, and Yueming Wang. Monocular camera-based face liveness detection by combining eyeblink and scene context. Telecommunication Systems, 2011. 3

This paper presents a face liveness detection system against spoofing with photographs, videos, and 3D models of a valid user in a face recognition system. Anti-spoofing clues inside and outside a face are both exploited in our system. The inside-face clues of spontaneous eye blinks are employed for anti-spoofing of photographs and 3D models. The outside-face clues of scene context are used for anti-spoofing of video replays. The system does not need user collaborations, i.e. it runs in a non-intrusive manner. In our system, the eyeblink detection is formulated as an inference problem of an undirected conditional graphical framework which models contextual dependencies in blink image sequences. The scene context clue is found by comparing the difference of regions of interest between the reference scene image and the input one, which is based on the similarity computed by local binary pattern descriptors on a series of fiducial points extracted in scale space. Extensive experiments are carried out to show the effectiveness of our system.

[2] Keyurkumar Patel, Hu Han, and Anil K Jain. Secure face unlock: Spoof detection on smartphones. TIFS, 2016. 3

With the wide deployment of the face recognition systems in applications from deduplication to mobile device unlocking, security against the face spoofing attacks requires increased attention; such attacks can be easily launched via printed photos, video replays, and 3D masks of a face. We address the problem of face spoof detection against the print (photo) and replay (photo or video) attacks based on the analysis of image distortion (e.g., surface reflection, moiré pattern, color distortion, and shape deformation) in spoof face images (or video frames). The application domain of interest is smartphone unlock, given that the growing number of smartphones have the face unlock and mobile payment capabilities. We build an unconstrained smartphone spoof attack database (MSU USSA) containing more than 1000 subjects.

[3] Allan Pinto, Helio Pedrini, William Robson Schwartz, and Anderson Rocha. Face spoofing detection through visual codebooks of spectral temporal cubes. TIP, 2015. 8

Despite important recent advances, the vulnerability of biometric systems to spoofing attacks is still an open problem. Spoof attacks occur when impostor users present synthetic biometric samples of a valid user to the biometric system seeking to deceive it. Considering the case of face biometrics, a spoofing attack consists in presenting a fake sample (e.g., photograph, digital video, or even a 3D mask) to the acquisition sensor with the facial information of a valid user. In this paper, we introduce a low cost and software-based method for detecting spoofing attempts in face recognition systems. Our hypothesis is that during acquisition, there will be inevitable artifacts left behind in the recaptured biometric samples allowing us to create a discriminative signature of the video .

[4] William Robson Schwartz, Anderson Rocha, and Helio Pedrini. Face spoofing detection through partial least squares and low-level descriptors. In IJCB, 2011. 3

Personal identity verification based on biometrics has received increasing attention since it allows reliable authentication through intrinsic characteristics, such as face, voice, iris, fingerprint, and gait. Particularly, face recognition techniques have been used in a number of applications, such as security surveillance, access control, crime solving, law enforcement, among others. To strengthen the results of verification, biometric systems must be robust against spoofing attempts with photographs or videos, which are two common ways of bypassing a face recognition system. In this paper, we describe an anti-spoofing solution based on a set of low-level feature descriptors capable of distinguishing between 'live' and 'spoof images and videos

[5] Abhinav Gupta. Revisiting unreasonable effectiveness of data in
deep learning era. In ICCV, 2017. 6

The success of deep learning in vision can be attributed to: (a) models with high capacity; (b) increased computational power; and (c) availability of large-scale labeled data. Since 2012, there have been significant advances in representation capabilities of the models and computational capabilities of GPUs. But the size of the biggest dataset has surprisingly remained constant. What will happen if we increase the dataset size by 10× or 100×? This paper takes a step towards clearing the clouds of mystery surrounding the relationship between `enormous data' and visual deep learning. By exploiting the JFT-300M dataset data was used for representation learning. Our paper delivers some surprising (and some expected) findings.we show that representation learning (or pre-training) still holds a lot of promise. One can improve performance on many vision tasks by just training a better base model. Finally, as expected, we present new state-of-the-art results for different vision tasks including image classification, object detection, semantic segmentation and human pose estimation. Our sincere hope is that this inspires the vision community to not undervalue the data and develop collective efforts in building larger datasets.

[6] Reference Face Graph for Face Recognition- Le An , Bir Bhanu

Face recognition has been studied extensively; however, real-world face recognition still remains a challenging task. The demand for unconstrained practical face recognition is rising with the explosion of online multimedia such as social networks, and video surveillance footage where face analysis is of significant importance. In this paper, we approach face recognition in the context of graph theory. We recognize an unknown face using an external reference face graph (RFG). An RFG is faces in the constructed RFG. Centrality measures are utilized to identify distinctive faces in the reference face graph.The proposed RFG-based face recognition algorithm is robust to the changes in pose and it is also alignment free. The RFG recognition is used in conjunction with DCT locality sensitive hashing for efficient retrieval to ensure scalability.Experiments are conducted on several publicly available databases and the results show that the proposed approach outperforms the state-of-the-art methods without any preprocessing necessities such as face alignment. Due to the richness in the reference set construction, the proposed method can also handle illumination and expression variation

III IMPLEMENTATION

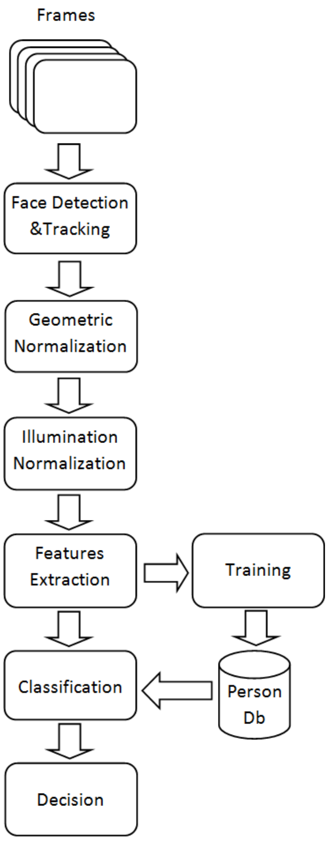


Fig.1 Architecture Diagram

In order to create this system first we will have to make the datasets. When the image quality becomes favourable different procedures will take place in the face recognition system the tasks are performed using the python queries "python encode_faces.py". The input will be taken from the dataset which will be received in the "encodings.py". There will be precision formatting in the system wherein face embedding for each face will occur. Secondly a file"recognize_faces_images.py" will contain all the required methods and the techniques for the process of identification of the face of the person from the given image of the dataset. The given file will be executed by the python command "pythonrecognize_faces_image.py-encodings". We can resize or turn the image for proximity with the goal for getting the desired output. The present classifier along with OpenCV libraries will enhance the outcome or results in the face recognition system.

PROPOSED SYSTEM

Our proposed system is made up of these following modules:

- Choosing the Input
- Extracting frames from videos
- Pre-processing steps
  - Installing and importing libraries
  - Reading image and processing
- Face detection
- Masking the Face
  - Shape of the Face
  - Fisherface Module
- Eye Detection
  - Retinal location
  - Retinal Displacement
- Alert message if conditions met

3.1 : CHOOSING THE INPUT

We must use the input feed from a webcam . as its a video we must extract the various frames from the videos and do the further processes.

3.2 : EXTRACTING FRAMES FROM VIDEOS

Actually videos are made up of series of still images. Getting all these still images is known as frame extraction. Getting all these images all at the same time is more complex and time consuming. So we code a program to extract the frames each and every image from the given input video. After extraction all these frames or still images will be stored in a separate folder named data. We can use any frame or image to detect the faces. We first get these frames from the video because we must make the video solid and clear by reducing the laid off frames and to speed up the processing of the video. We use certain python functions for getting frames of the video. Those functions are Video Capture (), read () and imwrite ().The video capture function consists of one argument that is the path of the video file. The function read () calls the object and reads the data of the object. The imwrite () function is made up of two arguments: the filename and the data we save to the file. We input the video, read the video and convert it into frames and stores the frames of video into a single folder named data and use the frames according to our usage that is to detect the lane lines found in the source image or video frame.

3.3 : PRE-PROCESSING STEPS

After we get the frame of the video or image we do some pre-processing steps so that we can get a clear cut view of the lane environment and detection of lane lines could be made easier. For doing this project of lane line detection we make use of three important libraries of python. First of all, for installing these libraries we should have the latest version of pip installed in our system. Those three important libraries are as follows:
I) OpenCV II) MatplotLib III) Numpy

The next step is reading the image and doing a simple processing step**.** We read the image or the frame extracted from the video using the cv2.imread () function. This method gets the image from the specific file where the image for face detection is present in our system. If the file is not available it returns none. In general it may consist of a second argument in which way the image must be seen or read. Those can be 1, 0 or -1(color, grey or including alpha channel). Normally the image will be in BGR format and now the simple processing step is we change the format into RGB format. We do this because we use the opencv function and pillow function of Matplot library which reads images in RGB format. Then we use plt.imshow () function and plt.show () function to print the image.

3.4 : FACE DETECTION

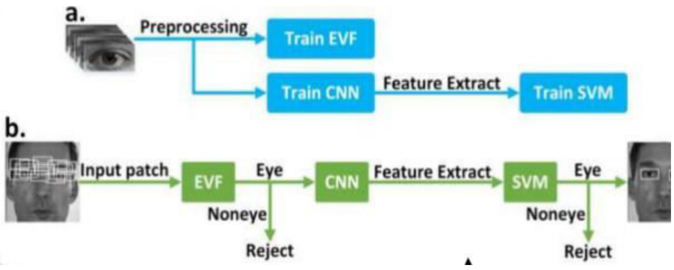
In order to create this system first we will have to make the datasets. When the image quality becomes favourable different procedures will take place in the face recognition system the tasks are performed using the python queries "python encode_faces.py". The input will be taken from the dataset which will be received in the "encodings.py". There will be precision formatting in the system wherein face embedding for each face will occur. Secondly a file"recognize_faces_images.py" will contain all the required methods and the techniques for the process of identification of the face of the person from the given image of the dataset. The given file will be executed by the python command "pythonrecognize_faces_image.py-encodings". We can resize or turn the image for proximity with the goal for getting the desired output. The present classifier along with OpenCV libraries will enhance the outcome or results in the face recognition system.

3.5 : MASKING THE FACE

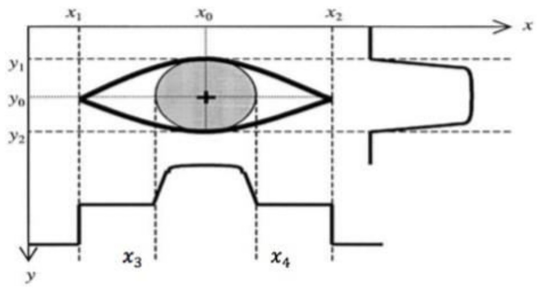
FisherFaces face recognizer algorithm extracts principal components that differentiate one person from the others. Helps to correct the error of misidentification due to different lightings.First input an image .Image will be extracted using fisherface method .Then image is trained in the database.Fisherface is one of the popular algorithms used in face recognition, and is widely believed to be superior to other techniques, such as eigenface because of the effort to maximize the separation between classes in the training process. The purpose of this research is to establish a program of face recognition application using fisherface method by utilizing GUI applications and databases that are used in the form of a Papuan facial image. Image recognition using fisherface method is based on the reduction of face space dimension using Principal Component Analysis (PCA) method, then apply Fisher's Linear Discriminant (FDL) method or also known as Linear Discriminant Analysis (LDA) method to obtain features of image characteristic. The algorithm used in the process for image recognition is fisherfaces algorithm while for identification or matching face image using minimum euclidean. The method used in this study is literature study that is studying and reviewing various books or literature related to mathematical concepts that underlies the formation of fisherface algorithms to recognize the image of a person's face which is then applied in programming language.

3.6 : EYE DETECTION

The proposed training and eye detection processes , in the training stage, three models (EVF, CNN and SVM) are trained. The training samples used for EVF are scaled to the resolutionof 48 ×24 pixels and samples for CNN are scaled to the size of 32 ×32 pixels, respectively.In the testing stage, the boosted cascade face detection is applied for initial face region location.The wrong detecting face regions are then manually corrected. The detected face image is normalized toan image of size 200 ×200 pixels. To improve calculation efficiency, the eye search region is limited to the top half of the detected face region, as human eyes tend to exist in this region. The input patch size(detected windows) is first set to 48 ×24 pixels and the patch passes the trained EVF. If the EVF judge sit as an eye image, the patch passes to the next stage for further eye classification. If the judgment has no eye image, it is filtered out. The same pre-treatments of image size are carried out for these candidates.



After the eye detection the Retina's location will be calculated using .



after plotting the points the retina can be found by using the formula

$$x_0 = \frac{x_3 + x_4}{2}, \quad y_0 = \frac{y_1 + y_2}{2}.$$

3.7 :ALERT MESSAGE IF THE CONDITIONS ARE MET

There are conditions like off screen time , drowsiness , looking away from the screen for a longer time . These conditions are the statements set to activate an alert message to the user , to alert them of their procrastination towards the task at hand .
a subject paying attention is defined according to the two

following rules:

1 The subject is paying attention if they are looking towards the front of the Monitor .

2 The subject who has their eyes closed in one frame is paying attention if the subject is paying attention in the previous frame and the following

frame according to the previous rule.

An alert message pops up when there is a contradiction to these rules as such looking away for a longer time , and being idle a longer time , etc. Performance of that person is analysed by calculating the number of times the alert message popped up during a specific time period and converting them into the percentage

Performance =
(no.of Alert messages/ total time duration) x 100

Thus the number of alert messages the user receives during his duration of task with its time iss converted into percentage and thus the performance of that user is well noted , this data can be used for the user's career progression and help ward off unnecessary human conduct on his promotion as it is happening now .

The problems below faced by the employee are minimized

1.Often times these workers will not take the credit they deserve for the great work they do. Once these workers start taking credit where it is due, management will start to see their potential.

2. Employees sometimes do not have clarity on their personal brand . i.e the difference one makes in their organisation .

3.If someone is perceived as grabbing power and not crediting colleagues for work well done, they may suffer the consequences of not getting the recognition they desire.

4. Not being able to make the case that they are a valuable asset to the organisation and ask for a promotion without a doubt.

5. Not liked by co-workers or the manager of an employee's group , also relatable to back stabbing of colleagues.

IV. CONCLUSION

In this paper we present an effective algorithm for eyes detection in face images. It consists of two steps: first the region candidate to contain one eye is detected in the whole image matching the edge directions with an edge template of the iris. Then, the search of the second eye is applied in the opposite regions whose distance and orientations are compatible with the range of possible eyes positions. Our system does not impose any constraint on the background and does not require any preprocessing step for face segmentation. Tests have been carried out on 1423 images of 6 people; with different eyes color, some of them wearing glasses. High detection rates have been obtained. The results are surprisingly good also when the eyes are not completely open. Future work will be addressed to add the eyes recognition ability to our system: indeed in our experiments we supposed that the face images contain always eyes, even if not necessarily completely open.

References

[1] Driver drowsiness detection system ,Belal al shaqaqi, abdullah salem baquhaizel., *IEEE* ,2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA),Electronic ISBN:978-1-4673-5540-7.

[2] **1.**Z. Huang, et al., "A benchmark and comparative study of video-based face recognition on COX face database", *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5967-5981, Dec. 2015.

k[3] L. Best-Rowden, B. Klare, J. Klontz and A. K. Jain, "Video-to-video face matching: Establishing a baseline for unconstrained face recognition", *Proc. IEEE Int. Conf. Biometrics Theory Appl. Syst.*, pp. 1-8, 2013.

[4] Y. Sun, X. Wang and X. Tang, "Deeply learned face representations are sparse selective and robust", *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2892-2900, 2015.

[5] J. Liu, Y. Deng and C. Huang, "Targeting ultimate accuracy: Face recognition via deep embedding", 2015.

[6] M. Turk and A. Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1):71-86, 1991.

[7]H. Wang, S.Z. Li and Y. Wang. Face recognition under varying lighting conditions using self quotient image. In IEEE International Conference on Automatic Face and Gesture Recognition (AFGR), pages 819-824. 2004.

[8] H. A. Rowley, S. Baluja, T. Kanade Neural Network-Based Face Detection IEEE Transaction on Pattern Analysis and Machine Intelligence Vol. 20,No. 1, January 1998, pp 23-38

[9] Y. Li, Xiang-lin Qi, Yun-jiu Wang, Eye detection by using fuzzy template matching and feature-parameter based judgement Pattern Recognition Letters 22 (2001) 1111-1124