Character Identification in Video Using Face Name Matching Method

[1]Kunal Khubchandani, [2]Surbhi Kharat, [3]Nisha Kharat , [4]prof. N. G. Bhojne

[1,2,3,4]Department of Computer Engineering Sinhgad College of Engineering, Pune , India

Email: [1]kunal.kk19@gmail.com, [2]kharat212@gmail.com, [3]kharatnisha19@gmail.com

Abstract— **With the advances in technology the hunger for the smart systems is seeing a tremendous rise in the last five years. Face detection and recognition is one of the method of adding to the range of such smart systems. Face recognition is the integral part of various surveillance systems in our modern world. Different techniques have been introduced and various theories have been proposed for the same. There are various techniques of face recognition in images but few are introduced in the field of face recognition through videos. In this paper we propose a system which focuses on recognition of faces in videos. This system includes following main steps input video, face detection, and face recognition and labelling. We will be using AdaBoost along with haarcascade for fast face detection. We have used OpenCV library which has a collection of various Image Processing functions. The system thus proposed reduces the cost and time of developing a face recognition unit. The proposed system can be used in various applications ranging from entertainment to security.**

Keywords— **Face detection, Eigen object recogniser, face recognition**

I. Introduction

Face detection has been widely used in many industrial areas such as digital camera, visual surveillance system etc. The role of frontal face is very important, because it provides vital information about the persons identity. For such purposes, we can use existing face detector as a frontal face detector. The aim of this paper is on annotating characters in videos which is character identification in videos. The objective is to identify the faces of the characters in the videos and label them with the corresponding names.

Our main focus is on face recognition and tagging a face with a name. Here a video will be played and a face detection algorithm will be used to detect faces from the video. When a face is detected a facial pattern will be created by extracting features and then stored in database for further use. Then a face recognition process starts, if a face has been detected for the first time then the system allows user to manually tag the user with a name. And if a match is found for the detected face then the system automatically tags the face with the name from the database.

Character identification is an extremely challenging task in computer vision. The reason is four-fold:

1) Face identification in videos poses more challenges that are, Low resolution, large motion, complex background and other conditions make the results of face detection and recognition unreliable[1].

2) The same character may be seen with different expressions throughout the video. There can be huge pose, wearing, clothing, expression and illumination variation, even makeup and hairstyle changes[2][3]. Moreover, there may be variations in character's age stage, e.g., from youth to the old age.

3) The number of faces in video that can be determined is of little importance. But it is quite challenging to detect each face in the video. More the number of faces more is the time required to detect each one. So there is a need for proper algorithm.

4) Also the angle of faces in the videos makes a lot of difference. The side faces can induce errors in the detection process. This can hamper the efficiency of the paper to a greater extent. As much needed faces can get skipped due to improper angle. So the proper frontal face must be detected.

II. System Layout

The system includes face detection, face recognition and name tagging as the main components.

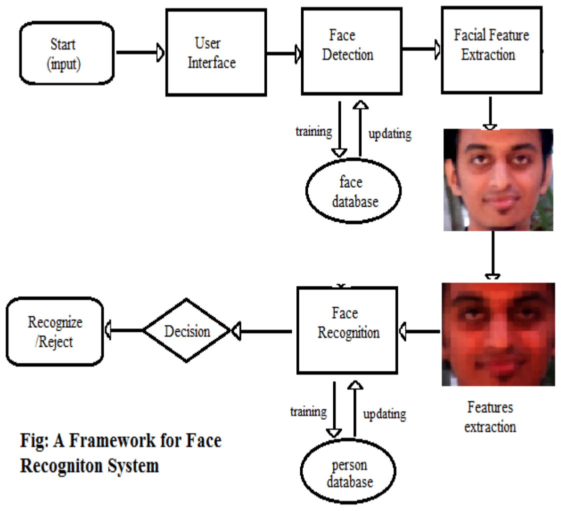


Fig. 1 Framework for face recognition system

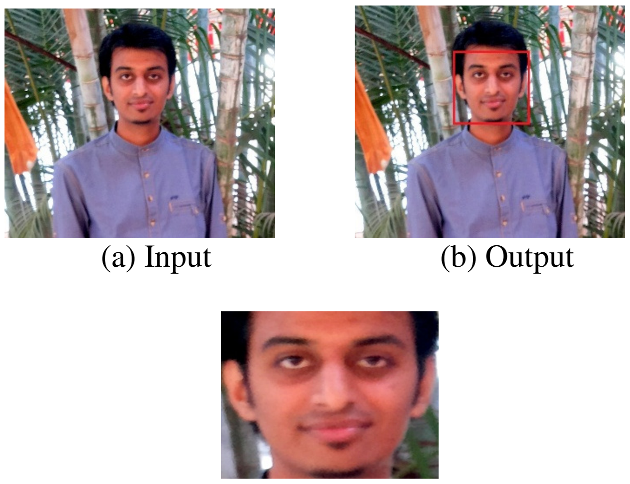
Overall system is designed to give better performance. The main components of this system are face detection and face recognition process which will include the face name tagging as well.

The face detector[4], where the input image will have the face identified and will then be cropped so as to include only that face and as little of the background as possible. This cropped image from the face detector will be passed to the recogniser where it will either be added to the image database used to train the recogniser or used to test the recogniser based on the images already added. The recogniser will have a threshold applied to it so as to allow for the rejection of an image whose closest match in the system does not meet the threshold. Once an image is recognised it is added to the database, it is tagged with the respective name and displayed to the user.

III. Face Detection

A. AdaBoost Algorithm with haar cascade classifier

Here we are using the AdaBoost algorithm and cascade technique as a basis for our face detection [4]. We will be using AdaBoost along with haarcascade for fast face detection. It uses haar basis functions and operates on the gray level images. We consider haar-like features i.e. the black region and the white region is separated using rectangles and in each rectangle area is calculated by multiplying the weights of those regions and again summing values we get haar feature value that will be computed for face detection. Here the decision depends on the parity of a threshold difference of the sums computed over the rectangular regions. AdaBoost helps in classification of frontal and non-frontal faces.


(a) Input (b) Output

(c) Cropped Image

Fig. 2 Face detection process

Mainly the AdaBoost trains the classifier using a set of examples from the training data set. There are classifiers which hold features that can act as threshold. Instead of using a single classifier with many features in it for computing, we build a cascade of classifiers and apply them in stages. Face detection is done by moving the cascade detector across the image. An image is either classified as a non-face or a decision is deferred and the image is passed to the second classifier. If the image passes for all the classifiers then it is considered as a detected image. This happens as a continuous process for various image. In this way, by using the classifiers frontal faces are classified and saved in the database. In this way face detection process is completed.

IV. Face Recognition

The main objective of the recogniser object is to compare an input image against those in the database to determine which it is most similar to one in database. We will be using emguCV library for face recognition. Once the input images have been segmented, they are passed to the constructor of the Eigen Recogniser object. Along with this array of segmented images, an array of strings is also passed to the array which contains the true names of each respective image. Once these arrays have been stored within the object, the learn () function is called. This array of images and strings serves as the training data for the recognizer.

The basis of the face recognition component is the Eigen faces technique[7]. This technique makes use of Principle Component Analysis to extract the features of the face that are deemed most important. We use the Eigen Object recogniser class from the library for recognizing the faces.

Here the cropped images are passed to the recogniser where the training process takes place. This process involves the creation of a subsequent projection of each training image onto the database. The values obtained for this are then stored within the newly trained recogniser.

The recognition process therefore involves the projection of the test image onto the database and then the identification of the nearest neighbour. Depending on the distance to this neighbour, a positive recognition is either returned or not.

After the face is recognised, that image will get tagged with its respective name from the database. If any face is detected for the first time i.e.it is not present in the database then manual tagging is allowed. That face will get tagged and that name along with the face will get stored in the database.

Face recognition uses Eigen Faces process which using a computing process for Eigen Faces is explained below

B. Computing The Eigen Faces

Before generating eigenfaces, the normalization of faceimages is done to line up the eyes and mouths, then we resample them at the same pixel resolution. Then the eigen faces are extracted out of the data of images by means of principal component analysis (PCA) in the following manner:

Given M face images with the size of h×w, each image is transformed into a vector of size D (=hw) and placed into the set

$\{\Gamma 1, \Gamma 2, \cdots, \Gamma M\}$.

The facial images are appropriately scaled and aligned, and the background (also possibly non-face areas such as neck and hair) should be removed or constant.

Every face differs from the average by the vector $\Phi i=\Gamma i-\Psi$, where the average face is defined by $\Psi=1M\sum Mi=1\Gamma i$.

The covariance matrix $C\in RD\times D$ is defined as

$C=1M\sum i=1M\Phi i\Phi Ti=AAT$,

Where $A=\{\Phi 1, \Phi 2, \cdots, \Phi M\} \in RD\times M$.

Determining the eigenvectors of C is an intractable task for typical image sizes when $D\gg M$. However, to efficiently compute the eigenvectors of C, one may first compute the eigenvectors of the much-smaller M×M matrix ATA. The eigenvector and eigenvalue matrices of ATA are defined as

$V=\{v1, v2, \cdots vr\}$

And Λ=diag {λ1, λ2, ⋯λr}, λ1≥λ2≥⋯λr>0, where r is the rank of A. Note that eigenvectors corresponding to eigenvalues of zero have been discarded.

The eigenvalue and eigenvector matrices of C are Λ and U=AVΛ−1/2, where U= {ui} is the collection of eigenfaces.

V. Conclusions

In our paper we have proposed a face recognition system which can detect and recognize characters in the videos. We will be using AdaBoost along with haarcascade for fast face detection. We have used OpenCV library which has a collection of various Image Processing functions. The system thus proposed reduces the cost and time of developing a face recognition unit. In future, it has a great scope in various face recognition systems like safe home systems, news broadcasts, smart systems, etc.

References

[1] Jitao Sang, Changshen Xu, Senior Member, IEEE, "Robust Face-Name Graph Matching for Movie Character Identification ", IEEE Transaction on multimedia,vol.10, No.10, 2012.

[2] Jitao Sang, Chao Liang, Changsheng Xu, Jian Cheng, "Robust Movie Character Identification and the Sensitivity Analysis", National Lab of Pattern Recognition, ©2011 IEEE.

[3] R. G. Cinbis, J. Verbeek, and C. Schmid, "Unsupervised Metric Learning for Face Identification in TV Video," in International Conference Computer Vision, 2011

[4] M.Gopi Krishna, A. Srinivasulu," Face Detection System On AdaBoost Algorithm Using Haar Classifiers" International Journal of Modern Engineering Research (IJMER),Vol. 2, Issue. 5.

[5] L. Lin, X. Liu, and S. C. Zhu, "Layered graph matching with composite cluster sampling," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 8, pp. 1426–1442, 2010.

[6] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y. W. Teh, E. G. Learned-Miller, and D. A. Forsyth, "Names and faces in the news," in CVPR, 2004, pp. 848–854

[7] Matthew. A. Turk, Alex P. Pentland "face recognition using eigen faces," deaptment of computer science, Massachustes institute of technology.

[8] M. Everingham, J. Sivic, and A. Zissserman, "Hello! My name is... buffy automatic naming of characters in tv video," in Proceedings of BMVC 2006, pp. 889–908

[9] S. Satoh and T. Kanade, "Name-It: Association of Face and Name in video", in Proceedings of CVPR, 1997, pp.368–373.

[10] Y. Zhang, C. Xu, H. Lu, and Y. Huang, "Character identification in feature length films using global face-name matching," IEEE Trans Multimedia, vol. 11, no. 7, pp. 1276–1288, November 2009.