International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 3 | ISSN : 2456-3307
DOI : 10.32628/CSEIT1953125

Computer Vision Based Face Identification

Surat Singh[1], Mrs. Priyanka[2]
[1]M. Tech Scholar, Computer Science and Engineering, I.G.U Rewari, SCET Mahendergarh, Haryana, India
[2]Assistant Professor, Computer Science and Engineering, I.G.U Rewari, SCET Mahendergarh, Haryana, India

ABSTRACT

The appearance of the Computer Vision Based Face has emerged as an interesting solution to addressing many of the identification needs and verification of patent applications. Computer Vision Based Identification is identified by the face and define by name. This is reviewing all aspects such as visibility, facial expressions and face identification. To see any face, we must have a recording image. We keep a picture of a picture with a single label with respect. We record that it usually contains a word and image. The main objective of this page shows and simplifies the simplest way to use state-of-the-art technology and be aware of the real time for many people using the Principal Component Analysis (PCA) to use it in multiple fields.

Keywords : Computer Vision Based Identification, Face Identification, Principal Component Analysis

I. INTRODUCTION

Computer Vision Based Face Identification is important part of the capability of human perception. The computational model not only contribute to theoretical insights but also to many practical applications like automated crowd, access control, design of human computer interface, content based image database management, criminal identification and so on. The identification of the Computer Vision Based Based Identification has been identified as an attractive solution to address the many demands of the time of identification and verification of evidence. Face-to-face recognition of a computer operating system is composed of complex systems that use mathematical calculations and mathematics, these find a picture in the raster mode (digital format) and edit and compare pixel with pixel using different methods to achieve quick and reliable results, the results of these results depend on the machine's use to analyze this for power for the use of computational systems that these methods, functions and policies require, these are the most popular methods used to solve the problem today. For example, an algorithm can analyze the status of the size, size, and / or eye shape, nose, cheekbones, and column. These features are used to search for other images with similar features. Some algorithms are preparing photo gallery and pressing data on the face, only to save the data in a helpful image on the face. The probe image is compared to the surface data. One of the first successful programs is based on similar matching techniques used for a set of sensitive face features, giving a type of depressed face type. Recognition algorithms can be divided into two main approaches, geometric, which looks at distinguishing features, or photometric, which is a statistical approach that distill an image into values and comparing the values with templates to eliminate variances. Popular recognition algorithms include Principal Component Analysis with eigenface, Linear Discriminate Analysis, Elastic Bunch Graph Matching fisher face, the Hidden Markov model, and the neuronal motivated dynamic link matching.

II. IMPLEMENTATION

Key Performance Analysis (PCA): - Computer Vision based Face Identification observation based on PCA (Primary Role Analysis) makes a lot of comparison with similarities between face detected and retained saved photographs for this purpose. and lighting conditions, this training makes this seem strong and very accurate Principal Component Analysis (PCA):- The face recognition algorithms based in PCA (Principal Component Analysis) do multiple comparisons and matches between a face detected and the trained images stored in binary database for this reason And for improve the accurate of recognition you should add several images of the same person in different angles, positions and luminance conditions, this training do this prototype solid and very accurate

The default parameters (scale_factor=1.1, min_neighbors=3, flags=0) are tuned for accurate yet slow object detection.

Improve the performance for slower:- All the process of image processing requires a lot of capacity to use information, in this case the CPI internals performed by this research model is extremely difficult to move through the CPO, the easiest way to improve the functionality of this Demo is changing the borders using Demo Find the Haar Cascade method , these allow you to reduce the number of estimates, criticism components and the actual image of the actual image taken by the Webcam better with performance performance. We work in: - observing the limits of these boundaries will affect the effectiveness of recognition algorithms.

Fast speed: -A quick function in real video videos settings: scale_factor = 1.2, min_neighbors = 2, flags = CV_HAAR_DO_CANNY_PRUNING, min_size = <at least (for example, ~ 1/4 to 1/16 video location video conferencing) .You can also change the Minsize parameter at a larger price.

Thumbnail or Resize:- Get the "icon" or change the original photo capture to reduce FrameGrabber processing time modify size sizes in small size (initially 320x240).

.**Parameters:**

haarObj: Haar classifier focuses on internal measurement measurement: Factory: A feature where the search window is combined between the following calls, for example, 1.1 means adding a window by 10%

minNeighbors: Minimum number (minus 1) of neighbor rectangles that makes up an object. All the groups of a smaller number of rectangles than min_neighbors-1 are rejected. If min_neighbors is 0, the function does not any grouping at all and returns all the detected candidate rectangles, which may be useful if the user wants to apply a customized grouping procedure

flag: Mode of operation. Currently the only flag that may be specified is CV_HAAR_DO_CANNY_PRUNING. If it is set, the function uses Canny edge detector to reject some image regions that contain too few or too much edges and thus cannot contain the searched object. The particular threshold values are tuned for face detection and in this case the pruning speeds up the processing.

minSize: Minimum window size. By default, it is set to the size of samples the classifier has been trained on (~20x20 for face detection)

Fisher recogniser:

Fisher recognition takes two changes as an Eigen manufacturer. The first is the number of sections containing discriminatory discrimination and the Fisherfaces policy. It's easy to keep all the parts, that is, the amount of installation. If you leave this at a default price (0), put it at least 0 or more of your training numbers. This is automatically set to the appropriate number (input entries - 1). he second variable is an unknown limit, if the maximum space is above this value, the prediction method () returns the number 1 anonymous This method works and the

screen is set to the default value of 3500, change this to limit your search results directly.If you change the value in the constructor the recogniser will need retraining.

LBPHFaceRecognizer :-

LBPHFaceRecognizer uses local banking patterns (LBP) to create a vector feature for a vector support machine or another machine learning algorithm. LBP unifies traditionally divergent statistical and structural models of texture analysis. LBP is very robust in real-world applications due to the manner in which it is deals with monotonic gray-scale changes caused by variations in illumination. See this ScholarPedia article for more information.

Mean Subtraction

This data is simple and makes our calculation of our covariance matrix easier now that does not remove the general description from each of our prices as that covariance requires at least the amount of data. In fact the removal of each line name from that element in that list.It is in fact the subtraction of the mean of each row from each element in that row.
(Alternatively the mean of each column from each element in the column however this would adjust the way we calculate the covariance matrix)

Covariance Matrix

The basic Covariance equation for two dimensional data is:

$$cov(x,y) = \frac{\sum_{i-1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)}$$

Which is similar to the formula for variance however, the change of x is in respect to the change in y rather than solely the change of x in respect to x. In this equation x represents the pixel value and $\bar{x}$ is the mean of all x values, and n the total number of values. The covariance matrix that is formed of the image data represents how much the dimensions vary from the mean with respect to each other. The definition of a covariance matrix is:

Eigenvectors and Eigenvalues

Eigenvalues produce many yamatrices but they are as a special solution. Eigenvalues are available for the matrix percentage of covariance with vector in both camps (ie Eigenvector). This creates a covariance matrix similar to the transformation matrix.

Feature Vectors

Now a usually the results of Eigenvalues and Eigenvectors are not as clean as in the example above. In most cases the results provided are scaled to a length of 1.

Transposition

The last section on PCA takes the vector signature matrix installation and returns it to the left of the converted data set (the set of data is from Stage 1 where the data is removed from the data).

Storing of the Training Data

The data source area is within the TrainedFaces folder of the application. One XAML file containing the person's name tags and file name for the training photo.

This structure can be easily converted to work with additional data or other structure. The following tasks must be adjusted to enable additional information and additional add-ons where they need it.

Each image is saved using a random number so that unique file identifiers can be generated. This prevents images being overwritten and easily allows several images for one individual to be aquired and stored with no problems.

The training form allows data to be added to the training program, it has noticed that this process can

be reduced. While the quickest method can load and write all data open and closed respectively this is not installed. If that is the case where mental management is considered it should be considered carefully so that the number of training models does not cause memory problems.

Improving Face Recognizer Accuracy

The methods to improve the accuracy of the FaceRecognizer have been made more stringent. The threshold used to control unknown faces either in the constructor or in the case of the EigenFaceRecognizer from the calculated distance can be adjusted to allow better accuracy. For the EigenFaceRecognizer this can be done on the form by changing the value within the 'Unknown Threhold' textbox. A default of 2000 is used but be increasing this to 5000 for example will mean it will be less likely to allow a false match. To high however and you may never achieve a match the eigenDistance is displayed in the right hand panel with the face to allow calibration.

The FisherFaceRecognizer and LBPHFaceRecognizer have the threshold set in the constructor. Additional controls can be added to the form to allow post training calibration. This is discussed with the 'How EMGU the FaceRecognizer Works' section.

Histogram equalization is also used to improve accuracy, this produces a more uniform image that is more resilient to changes in lighting. Alternative methods could also be taken to produce unique training sets. You could just take eyes and mouth features concatenate the data and use this however further experimentation would be required. In version 2.4.9 the face data is centralized to its detected position in order to remove background noise that could affect results in previous versions.

Detecting Unknown Faces

As discussed in previous comments reached by editing the 2.3.0 to allow access to the Eigen variable. EMGU v2.4.2 makes this possible by installing Recognition Result class to keep three changes. Searched face label, Eigen range, and Index made available. V2.4.2 allows you to set up an Eigen_threshold so that if the Eigen height exceeds this there is a negative effect it will be restored. This contains the same targeted bug with this code version. Classfier_Train.cs category has an Eigen bag in itself only when Eigen Face Recognizer is used. Fisher Face Recognizer and LBPH Face Recognizer allow the border to be installed correctly.

The main form includes a threshold box for calibrating the threshold value. This value changes depending on the size of your training data. To aid in calibration the Eigen distance is printed next to the name when recognized. The recognise method examines to see if the Eigen distance is greater than the set threshold. If not a "Unknwn" label is returned instead of the last person's name within the database.

Improving the detection performance

Instead of focusing on the development of slower research researchers this program was designed to increase performance in modern machines. It is often sought, although it is not possible, to process the real time. Real-time processing is closely related to the accuracy of the image processing algorithm. Quick algorithms are the result of small data processing and their ability to determine the true from false data. In acquisition photographs 30 each second-class is regarded as a standard standard, moving faster than our eyes can face and thus giving the lead organization. In real-world actions this can reduce the computer's accuracy.Today's instant cameras can find 640 x 480 images in 300 Fps, placing a standard web camera to humiliate. These cameras on the upper end use some grabbers specific to finding the image. It is unlikely that the average user will

encounter such a mass of perhaps 60 Fps, but the most important way these cameras do is to process the image during real working time. The frame grabbers have FPGA characters (Field-programmable Gate Array) attached to the card. This deals with the production of photographs but can also make processers such as histogram equalization and availability of the same time. In order to understand that this is achieved it is important to identify which FPGA chip and its buildings. FPGA has a simple advanced processor process that can be created to do a specific job if you have a smart phone and think it uses it (ARM Processor FPGA High). On the computer you can configure one drive name, another browser drive and other games to run. Obviously the difficulty and performance that prevents this. The FPGA processor can be created to have a more relevant architecture. So when you make the availability of the buffer you can also make a recognition of it.

While FPGA use is beyond the scope of this article a parallel architecture for image-processing can be produced. Many users of visual studio will have come across threading application before. This is where the processing of data is spread across the cores of your computer.

III. CONCLUSION

Over the years this process has been done by people. This process provides a direct image of the criminal but it is difficult to identify crime details and requires a large amount of personal responsibility.

The main objective of our project is to overcome the system-based challenges of the system based on a machine-based screening system. In this process we store criminal information in the image database and its image or photo. Then make a picture of different pieces of heads, forehead, eyes, nose, lips and mold and keep these clips in the database

IV. REFERENCES

[1]. Issam Dagher, Incremental PCA-LDA algorithm‖, International Journal of Biometrics and Bioinformatics (IJBB), Volume (4): Issue (2).

[2]. J. Shermina,V. Vasudevan,An Efficient Face recognition System Based on Fusion of MPCA and LPP, American Journal of Scientific Research ISSN 1450-223X Issue 11(2010), pp.6-19

[3]. Ishwar S. Jadhav, V. T. Gaikwad, Gajanan U. Patil, Human Identification Using Face and Voice Recognition‖, International Journal of Computer Science and Information Technologies, Vol. 2 (3), 2011.

[4]. Yun-Hee Han,Keun-Chang Kwak, Face Recognition and Representation by Tensor-based MPCA Approach, 2010 The 3rd International Conference on Machine Vision (ICMV 2010)

[5]. Neerja,Ekta Walia, Face Recognition Using Improved Fast PCA Algorithm, Proceedings of IEEE 2008

[6]. S.Sakthivel, Dr.R.Lakshmipathi, Enhancing Face Recognition using Improved Dimensionality Reduction and feature extraction Algorithms_An Evaluation with ORL database, International Journal of Engineering Science and Technology Vol. 2(6), 2010

[7]. Lin Luo, M.N.S. Swamy, Eugene I. Plotkin, —A Modified PCA algorithm for face recognition, Proceedings of IEEE 2003

[8]. A. Hossein Sahoolizadeh, B. Zargham Heidari, and C. Hamid Dehghani, A New Face Recognition Method using PCA, LDA and Neural Network, International Journal of Electrical and Electronics Engineering 2:8 2008

[9]. Feroz Ahmed Siddiky, Mohammed Shamsul Alam,Tanveer Ahsan and Mohammed Saifur Rahim,‖An Efficient Approach to Rotation

Invariant Face detection using PCA,Generalized Regression Neural network and Mahalanobis Distance by reducing Search space, Proceedings Of IEEE 2007 10Vapnik. Statistical Learning Theory. JohnWiley and Sons, New York, 1998.

[10]. E. Osuna, R. Freund, and F. Girosit. Training support vector machines: an application to face detection. Proc. of CVPR, pages 130–136, 1997. 12B. Heisele, T. Serre, and T. Poggio. A component based framework for face detection and identification. IJCV, 74(2):167–181, 2007.

[11]. Q. Tao, D. Chu, and J. Wang. Recursive support vector machines for dimensionality reduction. IEEE Trans. NN, 19(1):189–193, 2008. 14Marian Stewart Bartlett, Javier R. Movellan, Terrence J. Sejonowski, —Face Recognition by Independent Component Analysis, IEEE Transactions on Neural Networks, vol-13, No-6,November 2002, PP 1450- 1464.

[12]. Pong C.Yuen, J.H.Lai, —Face representation using independent component analysis, Pattern Recognition 35 (2002) 1247-1257.

[13]. Tae-Kyun Kim, Hyunwoo Kim, Wonjum Hwang, Josef Kittler, —Independent component analysis in a local facial residue space for face recognition, Pattern Recognition 37 (2004) 1873-1885

[14]. Computer Vision: A Modern Approach" by Forsyth and Ponce (2002).

[15]. Computer Vision: Algorithms and Applications" by Szeliski (2011).

[16]. Digital Image Processing" by Gonzalez and Woods (2001).

[17]. "The Essential Guide to Image Processing", by Bovik (2009).

[18]. "Computer Vision and Applications: A Guide for Students and Practitioners", by Jähne and Haußecker (2000).

[19]. HIPR2 Image Processing Worksheets (simple explanations of many computer vision topics)

Cite this article as :

Sh