PROJECT ON ENTROPY BASED PIXEL CLUSTER FOR FACE TRACKING

A PROJECT REPORT

SUBMITTED BY

Koushik Saha

REGISTRATION NO. **:151170510021 OF 2015-2016**

UNIVERSITY ROLL NO.:**11701015021**

Diganta Datta

Kaushik Roy

UNDER THE SUPERVISION OF

Asst. Prof. MANAS GHOSH

Department of

Computer applications

From

RCC INSTITUTE OF INFORMATION TECHNOLOGY

Affiliated to Maulana Abul Kalam Azad University Of Technology

RCC INSTITUTE OF INFORMATION TECHNOLOGY

CANAL SOUTH ROAD,BELIAGHATA,KOLKATA-700-015

May, 2018

CERTIFICATE

The report of the project “Multiple Face Tracking” using python, submitted by Koushik Saha (Roll No: 11701015021)has been prepared under my/our supervision for the fulfillment of the requirements for MCA degree in Maulana Abul Kalam Azad University Of Technology. The report is hereby forwarded.

_____

MANAS GHOSH

Dept of Computer Application

RCC INSTITUTE OF INFOR MATION TECHNOLOGY

KOLKATA

(Internal Supervisor)

Countersigned by

_____

Assistant Prof. Arup Kumar Bhaumik

Head of Department

RCC INSTITUTE OF INFOR MATION TECHNOLOGY

KOLKATA

Department of Computer Application

ACKNOWLEDGEMENT

I express my sincere gratitude to **Asst. Prof.Manas Ghosh** sirof Department of Computer Application, RCCIIT, our project guide and extending his valuable times for me to take up this problem as a project

I am also indebted to our Head of the Department (MCA), **Asst. Prof.**Arup Kumar Bhaumiksir for his unconditional help and inspiration.

Date : _____ _____

REGISTRATION NO :151170510021 OF 2015-2016

UNIVERSITY ROLL NO :11701015021

MCA 6th semester, RCCIIT

RCC INSTITUTE OF INFORMATION TECHNOLOGY

KOLKATA-700 015

CERTIFICATE OF ACCEPTANCE

The report of the project Multiple Face Tracking (using python) submitted by Koushik Saha (Roll:11701015021) is hereby recommended to be accepted for the partial fulfillment of the requirements for MCA degree in Maulana Abul Kalam Azad University Of Technology.

Name of the Examiners(s)	Signature with date
_____	_____
_____	_____

CONTENTS

Topics	Page No.
1. Purpose	6
2. Introduction	7
3. What is Image Processing	8
4. Purpose of Image Processing	**9**
5. Face Recognition	10
6. Data Acquisition	11
7. Database	12
8. Development Environment	12
9. Python openCV	12
10.Application	13
11. Image Entropy	14
12. Video Tracking	15
13. Objective	15
14. Algorithms	16
15. System Diagram	17
16. Experiment Result	18
17. Further Research	22
18. Conclusion	22

PURPOSE:

The purpose of this document is to present a detailed description of the requirements for the "**Multiple Face Tracking** " application. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is primarily intended for a reference for developing the first version of the system for the development team.

INTRODUCTION:

One of the most interesting areas of human computer interaction in face detection and tracking. Distinguishing facial features are comparatively low and it is most interesting task to observe these. Detection and tracking face object from video is a challenging task. In video the problem associated with unrestricted nature of faces and environment in which they occurred. As there are no restriction on the size,location occluded and jumbled face and poses of faces in video, difficulties may arise.It may be difficult to detect the face and tracking face from video sequence if face camera and background are in motion. The aforementioned methods mostly depends on specify visual and motion attributes. So the algorithm face difficulties in object detection and tracking. In addition, an efficient algorithm is needed to save computation cost in order to process the large amount. Face detection is the primary step of face recognition system. In biometric are vivid research has been made but most of this algorithm works on still images only. As such it cannot be applied directly video sequence. Lots of statistical-based methods, natural network-based methods etc. Frontal faces and non-frontal faces with the online adapted face models are now detected but template matching procedure. Less edge application is performed when slightly outside of facial features are introduced and higher edge focus is appeared in the neighbourhood of facial features. Pixel counting method is introduced to detect and track facial features. Temporal correspondence between frames is exploited through tracking. Head tracking facial feature tracking and combination of both head and facial tracking are the three categorized of tracking. Real time is to be considering as the foremost feature for video processing. Boundary matching and connection are the major step in tracking. The first one construct to determine the face boundary and the second one is evolved to extract the region between two key points.

Detecting face in the frame and then its tracking through integration in whole video sequence is possible. Loss of information cannot be avoided as detected and tracking are not similar and information from one to another is use at one time. On the whole instead of detection each frames temporary relationship between the frames can be generated in order to detect multiple human faces. It can be done in two phases, namely detection and prediction and then tracking. In every two second time interval pixel values of two contiguous still frames are compared to detect the location of the face object

and track the same. For detection of movement two frame are required. Reference frame is a first frame and input frame is second frame. The differences in pixel values are determined through the comparison of two frames.

This parer presents a novel face detection and face tracking strategy from video sequence. Video frame does not show any expression of prior localization of a face nor do make any postulation about the pose. A rectangular window is drawn by calculating top-left, top-right, bottom-left, bottom-right point face-image contour of a video frame. For removing noise some pre-processing arithmetical tasks is required for a video frame. In respect of contour boundary images production by edges is required. After that, scalar and vector distance between four corner points of two consecutive frames are computed to track the face location. Corner points displacement means face position and location changes into the next frame.

What is Image Processing?

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually **Image Processing** system includes - images as two dimensional signals while applying already set signal processing methods to them.
It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too.
Image processing basically includes the following three steps.

- Importing the image with optical scanner or by digital photography.
- Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.
- Output is the last stage in which result can be altered image or report that is based on image analysis.

Purpose of Image Processing:

The purpose of image processing is divided into 5 groups. They are:

1. Visualization - Observe the objects that are not visible.
2. Image sharpening and restoration - To create a better image.
3. Image retrieval - Seek for the image of interest.
4. Measurement of pattern – Measures various objects in an image.
5. Image Recognition – Distinguish the objects in an image.

Types

The two types of **methods used for Image Processing** are **Analog and Digital** Image Processing. Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. The image processing is not just confined to area that has to be studied but on knowledge of analyst. Association is another important tool in image processing through visual techniques. So analysts apply a combination of personal knowledge and collateral data to image processing.

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre-processing, enhancement and display, information extraction.

Face recognition:

In a general sense face recognition starts with face acquisition from a video stream or still image followed by feature extraction then some classification operation as in the figure provided by Rania. The features can be actual pixel values or handcrafted features such as dense-SIFT, Histogram of Oriented Gradients (HOG) features e.t.c and classifiers can be k-nearest neighbor, support vector machine (svm) classifiers e.t.c.

The problem with handcrafted features is that a lot of knowledge concerning the problem at hand is required. Deep learning on the other hand doesn't require handcrafting features because both features and classifier functions are learn from training data, hence making deep learning a very powerful tool for solving complex mapping problems.

Face recognition compared to image processing is far more complex and there are several techniques in use today as listed below. Note: some methods are very old but the intuition behind them might come in handy.

- ***EigenFaces:*** Eigenfaces comes from eigen vectors in mathematics. It's merely an application of Principle Component Analysis (PCA) on a collection of faces to reduce the dimensionality of face representations. EigenFaces can be linearly combined to approximate any given face.

- ***Graph matching:*** In Elastic Bunch Graph Matching a set of Gabor filters is used to construct an image graph of a face and face recognition is done by a straight forward matching of the graphs.

- ***Template matching:*** Face recognition using genetic algorithm based template matching can work remarkably well when implemented properly as earlier template based approaches were not practical due to severe variations between faces of the same person.

- ***Neural Networks:*** Face recognition using Neural networks can range from full-face processing to face-landmark based processing. The full-face recognition approach involves having a large amount of the individual face images from a single person. The trick is to train the network to fire correctly when a particular face is presented using

the usual back-prop algorithm. In the landmark based approach part detector neural nets are trained on face landmarks such as right eye, left eye e.t.c and the final detection or recognition is based partly on the geometric relationship between the landmarks. You can also check face detection using neural nets.

- ***Deep learning:*** The area of deep learning involves learning richer, high-level abstract features from the training set before using a final classifier back-end. The convolutional nets are state-of-the-art in areas such as category-level object recognition systems including face recognition. Recent publications such as FaceNet by Google and DeepFace by Facebook show how deep learning is employed in face recognition today as of 11/25/2015.

- ***3D based***: 3D techniques such those applied in Microsoft's Windows Hello involves modeling a face in 3D from one or more face shots. Ideally it is required to obtain such a representation from one shot as done in DeepFace. The subsequent processing routines can then process the face from a canonical representation hence this helps the face recognizer system handle severe view point variations.

Data acquisition:

The Face images of real human subjects were acquired from any videos and any images.From any video it just extract video frames and detect human faces from it.

We have used the grayscale palette.This is because the other RGB palettes unnecessarily increase the burden of the memory of the system and the computational complexity.This row images were acquired from video frames and then used in our algorithm to detect human faces.

Database:

Post data acquisition the raw images were incorporated into a database JU-FIR-F1:FIR face Database created in the Electrical Instrumentation and Measurement Laboratory,Electrical Engineering Department , Jadavpur University,Kolkata,India. The database consist of 215 images in total,and each and every videos are collected from Internet.

Development Environment:

- Python 3.6+
- OpenCV 3.0
- Numpy

Python OpenCV:

Officially launched in 1999, the OpenCV project was initially an Intel Research initiative to advance CPU-intensive applications, part of a series of projects including real-time ray tracing and 3D display walls.[4] The main contributors to the project included a number of optimization experts in Intel Russia, as well as Intel's Performance Library Team. In the early days of OpenCV, the goals of the project were described[5] as

Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.

Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.

Advance vision-based commercial applications by making portable, performance-optimized code available for free – with a license that did not require code to be open or free itself.

The first alpha version of OpenCV was released to the public at the IEEE Conference on Computer Vision and Pattern Recognition in 2000, and five betas were released between 2001 and 2005. The first 1.0 version was released in 2006. A version 1.1 "pre-release" was released in October 2008.

The second major release of the OpenCV was in October 2009. OpenCV 2 includes major changes to the C++ interface, aiming at easier, more type-safe patterns, new functions, and better implementations for existing ones in terms of performance (especially on multi-core systems). Official releases now occur every six months[6] and development is now done by an independent Russian team supported by commercial corporations.

In August 2012, support for OpenCV was taken over by a non-profit foundation OpenCV.org, which maintains a developer[7] and user site.[8]

On May 2016, Intel signed an agreement to acquire Itseez, the leading developer of OpenCV.

Application:

2D and 3D feature toolkits

Facial recognition system

Gesture recognition

Motion understanding

Object identification

Segmentation and recognition

Motion tracking

Image Entropy:

Description:

Image entropy is a quantity which is used to describe the `business' of an image, i.e. the amount of information which must be coded for by a compression algorithm. Low entropy images, such as those containing a lot of black sky, have very little contrast and large runs of pixels with the same or similar DN values. An image that is perfectly flat will have an entropy of zero. Consequently, they can be compressed to a relatively small size. On the other hand, high entropy images such as an image of heavily cratered areas on the moon have a great deal of contrast from one pixel to the next and consequently cannot be compressed as much as low entropy images.

Calculating Image Entropy:

Named after Boltzmann's H-theorem, Shannon defined the entropy H (Greek capital letter eta) of a discrete random variable X with possible values {x1, ..., xn} and probability mass function P(X) as:

$$\mathrm{H}(X) = \mathbb{E}[\mathrm{I}(X)] = \mathbb{E}[-\ln(\mathrm{P}(X))].$$

Here is the expected value operator, and I is the information content of X. I(X) is itself a random variable.

The entropy can explicitly be written as

$$\mathrm{H}(X) = \sum_{i=1}^{n} \mathrm{P}(x_i)\,\mathrm{I}(x_i) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i),$$

where b is the base of the logarithm used. Common values of b are 2, Euler's number e, and 10, and the corresponding units of entropy are the bits for b = 2, nats for $b = e$, and bans for b = 10.

In the case of $\mathrm{P}(x_i) = 0$ for some i, the value of the corresponding summand $0 \log_b(0)$ is taken to be 0, which is consistent with the limit:

$$\lim_{p \to 0+} p \log(p) = 0.$$

One may also define the conditional entropy of two events X and Y taking values x_i and y_j respectively, as

$$\mathrm{H}(X|Y) = -\sum_{i,j} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(y_j)}$$

where $p(x_i, y_j)$ is the probability that $X = x_i$ and $Y = y_j$. This quantity should be understood as the amount of randomness in the random variable X given the event Y.

Video tracking:

Video tracking is the process of locating a moving object (or multiple objects) over time using a camera. It has a variety of uses, some of which are: human-computer interaction, security and surveillance, video communication and compression, augmented reality, traffic control, medical imaging and video editing. Video tracking can be a time consuming process due to the amount of data that is contained in video. Adding further to the complexity is the possible need to use object recognition techniques for tracking, a challenging problem in its own right.

Objective:

The objective of video tracking is to associate target objects in consecutive video frames. The association can be especially difficult when the objects are moving fast relative to the frame rate. Another situation that increases the complexity of the problem is when the tracked object changes orientation over time. For these situations video tracking systems usually employ a motion model which describes how the image of the target might change for different possible motions of the object.

Algorithms:

Entropy based pixel cluster and face tracking algotrithm

The first step of a video based face recognition is the detection of face(s) from video frames. In a video scene, human faces can have unlimited number of orientations and positions, even corrupted by background noise and illumination;thereby making the face detection task more Complex and challenging detecting the moving part of a person reveals the outline region of the same which can be relatively easily easily detected on the basis of frame differences if the person is the only moving object.

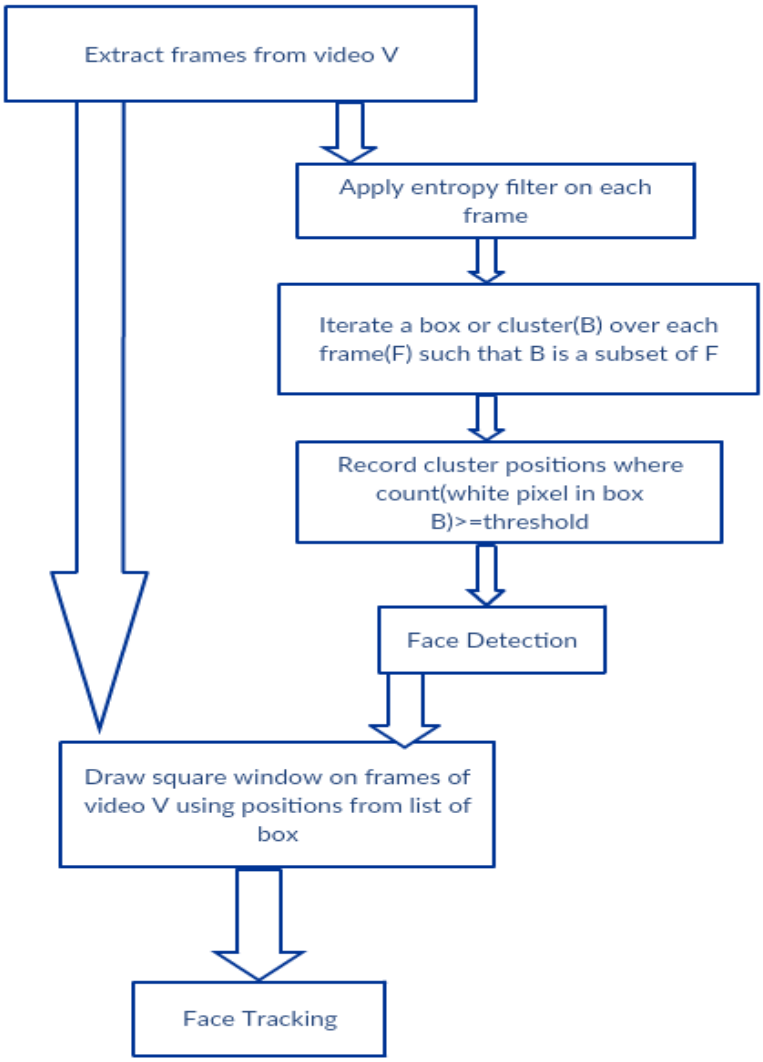
Let video V_n be a set of video frames F_i (where i=1,2,…n) extracted from the video.

- For each frame F_i we apply entropy based filter.
  - For each pixel p(x,y) of F_i we calculate the 8-pixel neighbour and calculate the entropy(E) of the neighbour matrix $M_{(3,3)}$.
    - If $E>=E_t$ then the pixel p(x,y) along with the neighbour matrix retains its original value(E_t :Global entropy threshold value).
    - Else value is set to 0.
    - Load new frames to a separate list V'_n
- For each frame in V'_n :
  - For each pixel $p_{(x,y)}$ we clone a cluster or box($B_{(h,w)}$) of a fixed dimensions.
  - If white pixel density of $B_{(x,y)}$ is greater than the white cluster threshold value then then store the position of box $B_{(x,y)}$ at position (x,y) in a new list box_list[N] (where N= no of boxes detected in the video).

- Else iterate over to the next block of pixel at (x+step_size,y+step_size). Where step size indicates the number of pixel to jump over during linear iterations of the frame.

- For each frame in video V_n :
  - For each box in box_list[]:
    - Draw a square window using the position stored in box.

System Diagram of the presented method:



Experimental Results:

Test case #1

Input Video

Output:

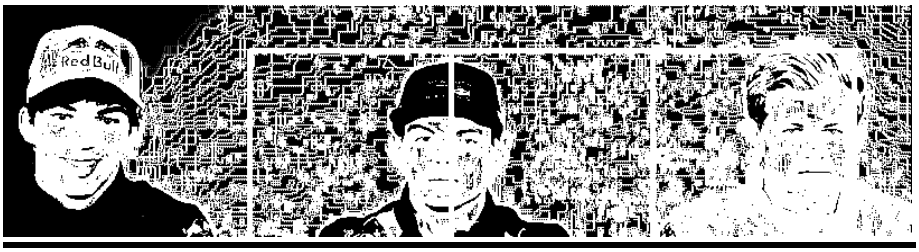

SOUTH COAST MITSUBISHI


SOUTH COAST MITSUBISHI


SOUTH COAST MITSUBISHI

Test Case #2

Output:




Red Bull


Red Bull


Red Bull

Further Research:

After experimentation with various mathematical operations to isolate human faces from a image; we can say that the best method to detect or identify a human facial structure is to use a Neural Network based pattern recognition model dedicated or trained to identify human faces. The limitations of the mathematical operations can be listed as:

- Varying illumination levels
- Extensive noise may be present or introduced during image capture.
- Aspect ratio of human faces may vary
- Largely dependant on various threshold or controlling variables for optimization or tuning.

Conclusion:

This paper discusses about a novel method to address one of the toughest challenges of any face recognition system,which is multiple image recognition during data acquisition phase.This paper elaborately represents,with theoretical analysis and experimental results,a novel multiple face detection methodology which was proposed to detect multiple face from any videos and images.We developed a an entropy based filter which is used to detect face from any videos and images.Eventually, after extensive experimentation ,we found that our proposed algorithm could perform very well showing high accuracy,stability and robustness over its other well-known counterparts.

However we can conclude that the severe disadvantages of representing i.e detection of a face using mathematical expressions for pattern recognition in a cluster of pixels can never produce optimum results in every single case.