ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 3, pp: (195-202), Month: July - September 2016, Available at: www.researchpublish.com

HOG Based Face Detection in Live Video Streaming

Detty Susan George

M.G University, Mount Zion College of Engineering, Pathanamthitta, India

***Abstract*: In this work it describes a complete framework for detecting objects in images and videos. Here the proposed approach is based upon ideas in computer vision, image processing and machine learning. These ideas provide an easy to use, simple and fast method for object detection. The main objective of this project is developing robust images feature sets for the task of object detection. And well normalized grids of gradient orientation histograms are used for proposed feature sets. These features helps to provide invariance in object location, changes in shape and good resistance to illumination changes and shadowing, background clutter and camera view point. For automatic/semi-automatic face detection here face recognition algorithms and motion detection algorithm are integrated. Here face detection technology is used to sort faces by their similarity to a chosen face or live webcam reducing user workload to search faces that belongs to the same person. Even many progress made in recent years, face recognition is a challenging topic in computer vision research.**

***Keywords:* HOG, LBP, face recognition, feature sets.**

I. INTRODUCTION

In computer vision face detection algorithms are widely used as they provide reliable and fast results depending on the application domain. To detect frontal and profile pose of people face in live video a multi view approach is presented here by using Histogram of Oriented Gradients, i.e. HOG, features. To detect faces a K-mean clustering technique is used in a cascade of HOG feature classifiers. Our system can be quickly trained before detection is possible. Here performance of the system is considerably increased in terms of lower false detection rate and lower computational cost when combined with motion constraint given by moving objects in video sequences. The moving object within a tracking framework is integrated with detected HOG features. This helps in reliable face tracking which results in several tested surveillance video sequences.

In many video applications it requires objects to be recognized, detected and tracked within a scene in order to identify scene activity and human behaviour. Most video surveillance applications are used to detect human activities captured by static cameras. Even cameras remain fixed, many issues occur such as in outdoor scenes varying lighting conditions may be displayed (e.g. sunny/cloudy illumination, shadows), crowded public areas (e.g. subways, malls) and a low resolution images can be obtained and sometimes highly compressed. Hence it is hard to detect and track objects in such complex environment. In this paper the aim is to track faces from a live video environment. Here we mainly use web camera for face recognition. For improved performance and increased accuracy the approach is designed such that it integrates face recognition algorithm and motion tracking method. In this proposed work it combines HOG (Histograms of Gradients) and LBP (Local Binary Pattern) for feature extraction and classification. This system has many applications in law enforcement, security, attendance control, visitors counting, traffic monitor and others.

II. EXISTING SYSTEM

In the existing system, it detects pedestrians using feature detection algorithm for human identification. Calculations and computations for this typical process is very complex. The existing system has to use complex and expensive computational devices to perform these calculations in a time critical manner. For the detection process here it uses a method of histogram of gradient for pattern matching to find the pedestrian detection. The system is trained with lot of pedestrian images. So that the HOG pattern matching process can compare the detected HOG patterns with the HOG patterns stored in the dictionary to find out the pedestrians.

The pedestrian detection approach implemented in this work has two phases training or learning phase and detection phase.

A. Training Phase:

The training phase includes image pre-processing, feature extraction and HOG pattern generation. Firstly image is pre-processed where the given image is uploaded and then the corresponding frame of the image undergoes a number of pre-processing steps. This includes elimination of the noise for better detection results. Then study the image based on its resolution, RGB configuration, size , type etc. Output of this analysis will be a image matrix that contains information about basic histogram. Secondly the feature extraction is performed which involves a filter based representation to decompose images into information at multiple scales to extract texture of human .Feature extraction include image binarization which is used to remove colour variations such as clothing in human images. This will result an image with only two colours, black and white. After completing binarization process edges are detected from binary image. The goal of edge detection is to produce a line from a scene and to extract important features from edges of an image. Canny edge detection algorithm is used for edge detection. In HOG pattern generation feature vectors are generated. HOG is calculated as follows:

- Convert an input image to gray scale.
- Calculate edge degree and strength per pixel.
- Splits an image into cell areas.
- Builds a histogram from edge degree and strength.
- Normalize the histogram by neighbour cells.

B. Detection Phase:

In detection phase all the above steps are repeated for the given input image and it compares the detected HOG patterns with the HOG patterns stored in the dictionary to find out the pedestrians.

III. PROBLEM DEFINITION

In the first stage it focused on the issue of pedestrian detection in images. In practice, it involves the construction of human detectors, where the detectors search given mages for human and localize them. A human detector can be considered as a combination of two key factors: a feature extraction algorithm that transformers image regions to feature vectors, and a detector that uses the computed features to make human/non-human decisions. This targets general purpose human detectors that do not make strong contextual assumptions. This work provides improved detection rate with low computational cost and also allows real time multi-scale detection.

So towards the next stage it proposes a method for recognising human faces from live video streaming. For this it integrates face detection algorithm with motion detection one. Here it uses both HOG and LBP (Local binary pattern) for face detection.

IV. PROPOSED SYSTEM

Our aim is to track a human face from a live video stream. Here we use a web cam which is embedded into the program. When the object moves at that time itself we have to track the object and have to find the edge. While taking a picture it contains only one frame but in case of a video the frames get changed every per second. The program we write should be capable of taking all these frames within the given time gap. Here we analyse a collection of frames that consecutively changes from one frame to other as it is a lively environment. These frames are further processed for the recognition.

There are many face detection algorithms but here we go for detecting faces when objects move. At the same time it also detects edge of the moving object.

This work has described a complete framework for the problem of detecting objects in images and videos. The proposed approach builds upon ideas in image processing, computer vision and machine learning to provide a general, easy to use and fast method for pedestrian detection. Our main contribution is the development of robust images feature sets for object detection tasks. And also proposed feature set based on well normalized grids of gradient orientation histograms. These features provide some invariance to shifts in object location and changes in shape and good resistance to changes in illumination and shadowing, background clutter and camera view point. The overall conclusions are that capturing fine detail with unsmoothed gradients and fine orientation voting, moderately coarse spatial binning; strong normalized and overlapping blocks are needed for good. The descriptors do not involve any arbitrary thresholding of edges and they are relatively fast to compute. A straight forward idea for automatic/semi-automatic face detection is to integrate face recognition algorithms and motion detection algorithm. Here we used face detection technology to sort faces by their similarity to a chosen face or live webcam, reducing user workload to searching faces that belongs to the same person.

However, despite progress made in recent years, face recognition continues to be a challenging topic in computer vision research. Most algorithms perform well under a controlled environment, while in the scenario of family photo management, the performance of face recognition algorithms becomes unacceptable due to difficult lighting/illumination conditions and large head pose variations. These works are proposed as a simple pre-processing step in the whole system without adopting sophisticated techniques. Here applied a modified k means clustering approach for cleaning up the noisy web facial images. The new algorithm works by normalizing each face into a 150 x 120 pixel image, by transforming it based on five image landmarks: the position of eyes, the nose and the two corners of the mouth. It then divides each image into overlapping patches of 25 x 25 pixels and describes each patch using a mathematical object known as a vector which captures its basic features. To balance the cluster sizes, the bisection scheme is also employed. Specifically, in each iteration step, we partition the largest cluster into two parts by cutting its largest MST edge to ensure the size of the smaller cluster in the cutting result is larger than a predefined threshold value.

A. Algorithms and Techniques used:

- Canny Edge detection algorithm
- Pseudo-Zernike face recognition algorithm (local binary pattern)
- Histogram of Gradients (HoG) pattern matching algorithm

Canny Edge Detector:

1) Smooth image with a Gaussian optimizes the trade-off between noise filtering and edge localization
2) Compute the Gradient magnitude using approximations of partial derivatives 2x2 filters
3) Thin edges by applying non-maxima suppression to the gradient magnitude
4) Detect edges by double thresholding

Gradient:

- At each point convolve with magnitude and orientation of the Gradient are computed as
- Avoid floating point arithmetic for fast computation

Non-Maxima Suppression:

- Thin edges by keeping large values of Gradient

Thresholding

Reduce number of false edges by applying a threshold T

- all values below T are changed to 0
- selecting a good values for T is difficult
- some false edges will remain if T is too low
- some edges will disappear if T is too high
- some edges will disappear due to softening of the edge contrast by shadows

Double Thresholding:

- Apply two thresholds in the suppressed image
- $T_2 = 2T_2$
- two images in the output
- the image from T_2contains fewer edges but has gaps in the contours
- the image from T_1 has many false edges
- combine the results from T_1 and T_2
- link the edges of T_2 into contours until we reach a gap
- link the edge from T_2 with edge pixels from a T_1 contour until a T_2 edge is found again

Pseudo-Zernike face recognition algorithm:

- Input: Training Image set.
- Output: Feature extracted from face image and compared with centre pixel and recognition with unknown face image.
- 1. Initialize temp = 0
- 2. FOR each image I in the training image set
- 3. Initialize the pattern histogram, H = 0
- 4. FOR each centre pixel tc €I
- 5. Compute the pattern label of tc, LBP(1)
- 6. Increase the corresponding bin by 1.
- 7. END FOR
- 8. Find the highest LBP feature for each face image and combined into single vector.
- 9. Compare with test face image.

How Local Binary Pattern Works?

- LBP operator works with the eight neighbours of a pixel, using the value of this centre pixel as a threshold.
- If a neighbour pixel has a higher gray value than the centre pixel (or the same gray value) than a one is assigned to that pixel, else it gets a zero.
- The LBP code for the centre pixel is then produced by concatenating the eight ones or zeros to a binary code
- Later the LBP operator was extended to use neighbourhoods of different sizes.
- In this case a circle is made with radius R from the centre pixel. P sampling points on the edge of this circle are taken and compared with the value of the centre pixel.
- To get the values of all sampling points in the neighbourhood for any radius and any number of pixels, (bilinear) interpolation is necessary. For neighbourhoods the notation (P, R) is used.
- If the coordinates of the centre pixel are (xc, yc) then the coordinates of his P neighbours (xp, yp) on the edge of the circle with radius R can be calculated with the sinus and cosines
- If the gray value of the centre pixel is gc and the gray values of his neighbours are gp, with $p = 0, ..., P - 1$, than the texture T in the local neighbourhood of pixel
- Once these values of the points are obtained is it also possible do describe the texture in another way. This is done by subtracting the value of the centre pixel from the values of the points on the circle
- The Local Binary Pattern characterizes the local image texture around (xc, yc).

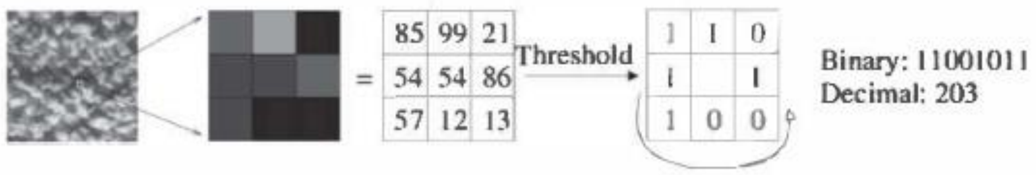

Fig.1: Basic LBP operator

B. Modules:

This project include following modules:

- Feature Extraction
- Motion Detection
- HOG Pattern Generation
- Face Monitor Module

1. Feature Extraction:

In this system the descriptor vectors representing the contents of images are made. Here we use 3 Descriptor vectors which represent the content of the image.

- Gradient Edge detection (Canny Edge Detection Algorithm)
- Harris Laplace corner detection
- Zernike polynomials region detection

Canny Edge Detection Algorithm:

Step 1: The first step is to filter out any noise in the original image before trying to locate and detect any edges.

Step 2: After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image.

Step3: Finding the edge direction is easy once the gradient in the x and y directions are known. However, you will generate an error whenever sum X is equal to zero. So in the code there has to be a restriction set whenever this takes place. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If GY has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees.

Step 4: Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So if the pixels of a 5x5 image are aligned as follows:

x x x x x

x x x x x

x x a x x

x x x x x

x x x x x

Then, it can be seen by looking at pixel "a", there are only four possible directions when describing the surrounding pixels - **0 degrees** (in the horizontal direction), **45 degrees** (along the positive diagonal), **90 degrees** (in the vertical direction), or **135 degrees** (along the negative diagonal). So now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to (e.g. if the orientation angle is found to be 3 degrees, make it zero degrees). Think of this as taking a semicircle and dividing it into 5 regions. Therefore, any edge direction falling within the **yellow range** (0 to 22.5 & 157.5 to 180 degrees) is set to 0 degrees. Any edge direction falling in the **green range** (22.5 to 67.5 degrees) is set to 45 degrees. Any edge direction falling in the **blue range** (67.5 to 112.5 degrees) is set to 90 degrees. And finally, any edge direction falling within the **red range** (112.5 to 157.5 degrees) is set to 135 degrees.

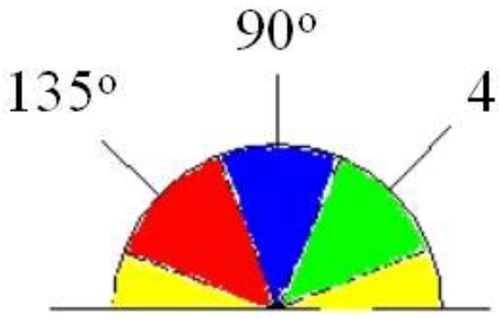

Fig 2: Edge direction

Step 5: After the edge directions are known, non-maximum suppression now has to be applied.

Harris Laplace Corner Detection :

Harris detector localizes corners, based on the fact that gradient values will change in multiple directions around a corner. It uses a scale adapted version of the second moment matrix, known as Harris matrix

$$M = \sigma_D^2 \cdot g(\sigma_I) * \begin{bmatrix} L_x^2(x, \sigma_D) & L_x L_y(x, \sigma_D) \\ L_x L_y(x, \sigma_D) & L_y^2(x, \sigma_D) \end{bmatrix},$$

where σD is the differentiation scale, σI is the integration scale and Lz is the derivative computed in z direction. Differentiation scale σD is used to compute the local derivatives with Gaussian kernels, and a Gaussian window with a size σI is used to smooth and average the neighborhood around the point. The eigenvalues of this matrix represent the gradient changes in two directions.

Zernike Polynomials Region Detection:

In this it normalizes pseudo-Zernike functions to a 2D rectangular patch and convolve the input image using them. Each pseudo-Zernike function is used as a filter and this response constitutes the interestingness measure for detection of point locations.

2. Motion Detection:

Motion detection includes following steps:

Step1: Identifies the objects that have moved between the two frames (using difference and threshold filter). The difference between each corresponding pixel of two frames is calculated; and the pixels with difference greater than the specified threshold are marked in foreground colour (white). All the other pixels are marked in background colour (black). So, the output of Step1 will be a binary-image with only two colours (black and white). The intensity values (brightness level) of the pixels are used to calculate the difference. The intensity value of each pixel in a gray scale image will be in the range 0 to 255. If RGB images are used, then the gray scale intensity value can be calculated as (R + G + B / 3).

Step2: Identifies the significant movements and filters out the noise that is wrongly identified as motions (using erosion-filter). Erosion-filter removes the pixels that are not surrounded by enough amounts of neighbouring pixels.

3. HOG Pattern Generation:

In this module feature vector are generated by using Histograms of oriented Gradients (HOG). HOG are feature descriptors used for the purpose of object detection. This technique counts occurrences of gradient orientation in localized portions of an image. HOG is calculated as follows:

1) Convert an input image to gray scale.

2) Calculate edge degree and strength per pixel.

3) Split an image into cell areas.

4) Build a histogram from edge degree and strength

5) Normalize the histogram by neighbour cells.

4. Face Monitor Module:

Face monitor module communicate with the camera/Web cam connected and captures the image in accordance with user input to system. It uses the dynamic linked library named "avicap32.dll" and uses Windows messaging service to communicate to the driver of image capturing device to view and capture image.

The below figure illustrates the complete work flow of the system

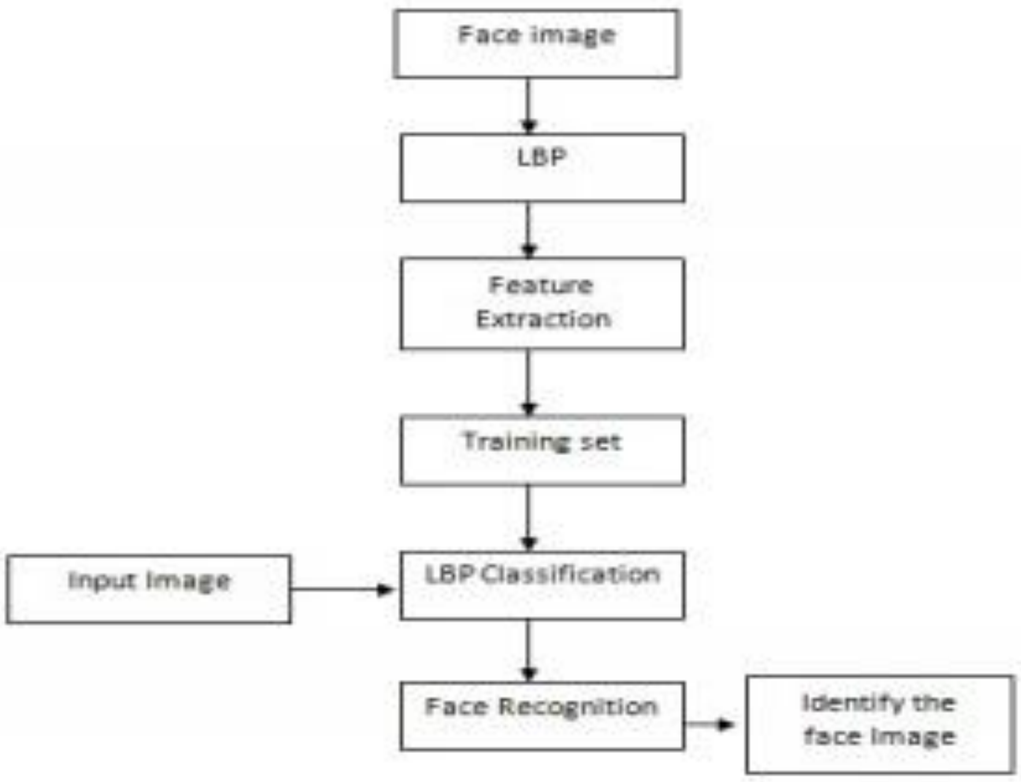

Fig 3: Work flow of the system

V. CONCLUSION

A Histogram of Oriented Gradients (HOG) technique is designed to detect objects. The technique here detects multiple views of people face captured in video sequence. To model facial feature a standard clustering technique is deployed. The work integrates face detection algorithm and motion detection method for recognising faces from a live video. The proposed face detection algorithm uses a multi resolution approach to detect different size faces. Tracks object from a video sequence based on edge detection and feature matching is presented .Canny's operator is used to detect the edges present in respective frames, which helps to estimate the object boundaries. This proposed work results in better performance with low computational cost and takes less time to process.

ACKNOWLEDGEMENT

I would like to extend my thankfulness to the reference authors, as well as reviewer of my paper.

REFERENCES

[1] Adam A., Rivlin E., and Shimshoni I. Robust fragment based tracking using integral histogram. In Computer Visionand Pattern Recognition - CVPR, 2006.

[2] T. Ahonen, A. Hadid, M. Pietikainen and T. M aenpaa. “Face recognition based on the appearance of local regions”,In Proceedings of the 17th International Conference on Pattern Recognition, 2004.

[3] R. Gottumukkal and V.K. Asari, “An Improved Face Recognition Technique Based on Modular PCA Approach ”Pattern Recognition Letters, vol. 25, pp. 429- 436, Mar. 2004.

[4] Laptev I, Improvements of object detection using boosted histograms. In Proceedings of the British Machine Vision Conference, 2006.

[5] W. Zhao and R. Chellappa “Robust face recognition using symmetric shape from- shading” Technical Report, Center for Automation Research, University of Maryland, 1999.

[6] Schwerdt K. and Crowley J.L. Robust Face Tracking using Color. In 4th IEEE International Conference on Automatic Face and Gesture Recognition, 2000.

[7] Jordao L., Perrone M., Costeira J.P., and Santos-Victor J. Active Face and Feature Tracking. In International Conference on Image Analysis and Processing - ICIAP, 1999.