MAIN PROJECT REPORT -INTELLIGENT SURVEILLANCE SYSTEM

PROJECT GUIDE:

Ms. Aparna Devi P.S

SUBMITTED BY:

Anjali Sam(6)
Anjana Krishnakumar V(8)
Chandana Narayanan(22)
Nayana Dilip(40)
Sudheep P(65)

Chapter 1

INTRODUCTION

Real-time object detection and tracking is a vast, vibrant yet complex area of computer vision. Due to its increased utilization in surveillance, tracking systems are used in security and many other applications. Automatic visual object counting and video surveillance have important applications for home and business environments. Moving object detection and tracking is often the first step in applications such as video surveillance.

Normal surveillance systems do not have a method to count and classify objects such as humans, vehicles etc. In this project, a moving human and vehicle detection & tracking system is proposed to maintain an automatic database of information. Counting and classification are the two main features of the project.

The project maintains a database with the count of vehicles and humans captured from the surveillance camera. Identification of crimes and security threats can be done easily by searching through the database. It reduces the wastage of time involved in scanning through the surveillance videos in search of information. Details are well documented since it is entered into a database.

It can be used in shopping malls to maintain an automatic database of the number of vehicles or people visiting the mall. It can also be used in high security areas like airports, defence areas, military areas etc. and can be used later in case of security threats.

It also can be extended in future to detect the number plates of vehicles, process the image and store the vehicle number in the database. Further it can be implemented in Linux based SBCs for commercial purpose.

Chapter 2
LITERATURE REVIEW

2.1 COMPARISON WITH EXISTING PROJECTS

2.1.1 To Detect and Track Moving Object for Surveillance System

This system finds its application in CCTV network to detect and track abnormal behaviour of some people or vehicles, which can be used in home and business surveillance system to detect and track moving objects. The method they use is colour background modelling with sensitivity parameter (delta) to remove noises and to detect and track moving objects very easily.

Here, extraction of movable region from sequential images is carried out by using RGB background modelling method. But the main drawback is that it is very sensitive to even small changes in light scattering or reflection. To detect image more accurately from the crowd of pixel morphological operations like erosion and dilation operations are applied on the binarized images.

Blob labelling is used to group moving object which requires less computational cost is also easily implemented. Now to track moving object, procedure consists of predicting the position of each group, recognizing the similarity of each group in the sequential frame and identify appearing and disappearing of new group.

Limitations and method of improvement:

Here, RGB background modelling method is used, which is highly prone to errors as it is very sensitive to variations in light. Intelligent surveillance system proposes to use grayscale modelling which is much more efficient.

2.1.2 Moving Vehicle Detection for Measuring Traffic Count Using OpenCV

Vehicle detection, tracking, classification and counting is very important for military, civilian and government applications, such as highway monitoring, traffic planning, toll collection and traffic flow.

The system uses a single camera mounted on a pole or other tall structure, looking down on the traffic scene. Here, the moving-object recognition scheme uses an adaptive background subtraction technique to separate vehicles from the background. The background is modelled as a slow time-varying image sequence, which allows it to adapt to changes in lighting and weather conditions. As a pattern passes over the digital detector, the change is recognized and a vehicle is counted. The length of time that this change takes place can be translated into speed estimates.

This system consists of three stages.

1) System Initialisation

System gets initialised and set up in this stage. Camera records continuous stream of data and sends to the system for analysis.

2) Background Subtraction

In this stage, a set of frames are taken into focus and on successive analysis and operations background subtraction takes place.

3) Vehicle Detection

In this stage, using the subtracted background image all the moving vehicles/objects can be tracked and counted.

Limitations and method of improvement:

Instead of using Background reduction method, we use Haar classifier algorithm for image extraction. This stands out because the former has disadvantages such as errors due to scattering or reflection in case of indoor scenes, and errors due to rain or wind in case of outdoor scenes, which are ruled out in the latter case.

2.2 THEORY

2.2.1 Background reduction (Foreground detection)

Background subtraction, also known as Foreground Detection, is a technique in the field of image processing wherein an image's foreground is extracted for further processing. Generally an image's regions of interest are objects (humans, cars, text etc.) in its foreground. After the stage of image pre-processing, object localisation is required which makes use of this technique. Background subtraction is a widely used approach for detecting moving objects in videos from static cameras. The rationale in the approach is that of detecting the moving objects from the difference between the current frame and a reference frame, often called "background image", or "background model".

Disadvantages:

a. Background subtraction is generally based on a static background hypothesis which is often not applicable in real environments. With indoor scenes, reflections or animated images on screen leads to background changes.
b. In the same way, due to wind, rain or illumination changes brought by weather, static backgrounds methods have difficulties with outdoor scenes.

2.2.2 Haar classifier algorithm (Viola Jones object detection)

In this algorithm, a cascade function is trained from a set of positive and negative images to extract features. For the training, a 24x24 image is considered. Viola-Jones have empirically found that a detector with a base resolution of 24x24 pixels gives satisfactory results. A window of target size is moved over the input image. For each subsection of the image, the Haar like character is calculated. The difference can then be used to separate non-objects from objects (vehicles), using Haar like features.

2.2.2.1 Viola Jones object detection algorithm

An object detection system is designed by giving an input of objects and non-objects and training a classifier that identifies an object. When the training is done, this classifier can be used to extract features and store it in a file. This can be used to detect any objects in the input images by comparing it with the available features stored in the file. In this case, the object is vehicle or a human.

Haar features are used to detect the presence of a feature in a given image. Example for Haar features are shown in figure 2.1. The black region has a value of +1 and the white region has a value of -1. Each feature results in a single value which is calculated by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle.

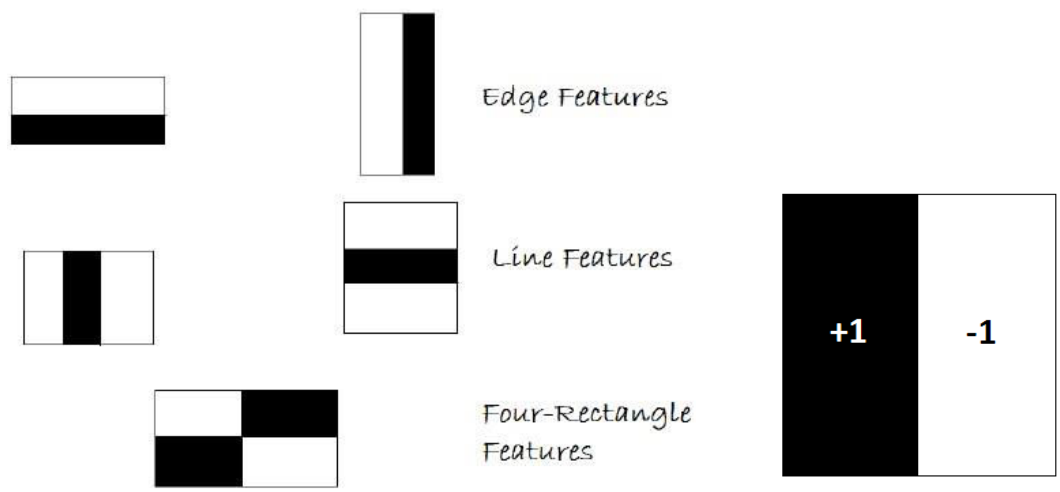

Fig 2.1: Haar features

Fig 2.2 Example of haar features-1

Consider examples in figure 2.2 and 2.3 used to distinguish the features. The feature applied to identify the bridge of the nose cannot be used to identify the dark region on the eyes and the bright region on the cheeks. These features will give a positive value only if it is placed in the right position.

Fig 2.3 Example of haar features-2

However, it is difficult to calculate the sum of all pixels in an image to find its area. Viola Jones devised the concept of integral image to overcome this.

2.2.2.2 Integral image

Figure 2.4 explains the concept of integral image. In an integral image, the value at pixel (x,y) is the sum of pixels above and to the left of (x,y).

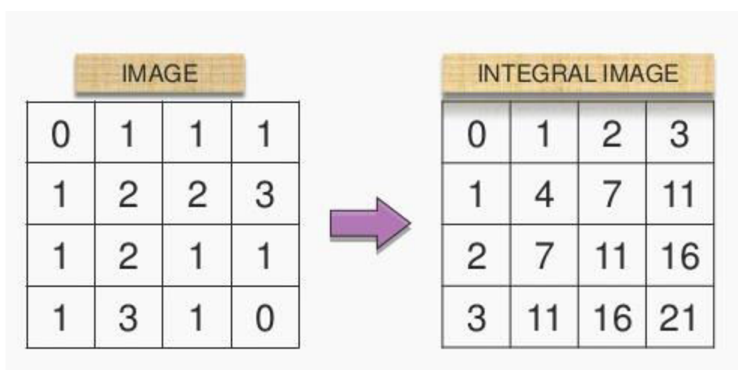

Fig 2.4 Conversion of image to integral image

Hence, the area of the whole image can be calculated using the four corner values of pixels of the image. The key advantage of Haar like features over most other features is its calculation speed, due to the use of integral images.

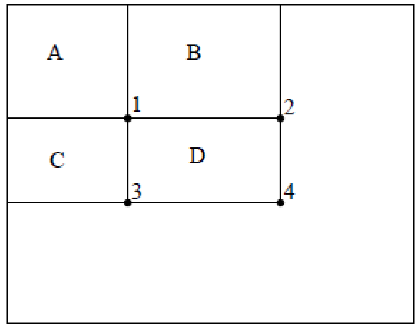

Fig 2.5 Calculation in integral image

$$\textit{Sum of all pixels in } D = 1+4-(2+3) = A+(A+B+C+D)-(A+B+A+C) \qquad (2.1)$$

Viola Jones algorithm uses a 24x24 window as the base window size to evaluate the features of an image. Even in the smallest window of size 24x24, if all possible parameters of the basic haar features like position, scale, type etc are considered, there will be around 160,000+ different features in a window. From this, the features which are redundant need to be eliminated and only the most required ones should be selected. This is done by using Adaboost algorithm. Adaboost is a machine learning algorithm which helps in finding only the best features among these 160,000+ features.

Selection of relevant features using Adaboost algorithm:

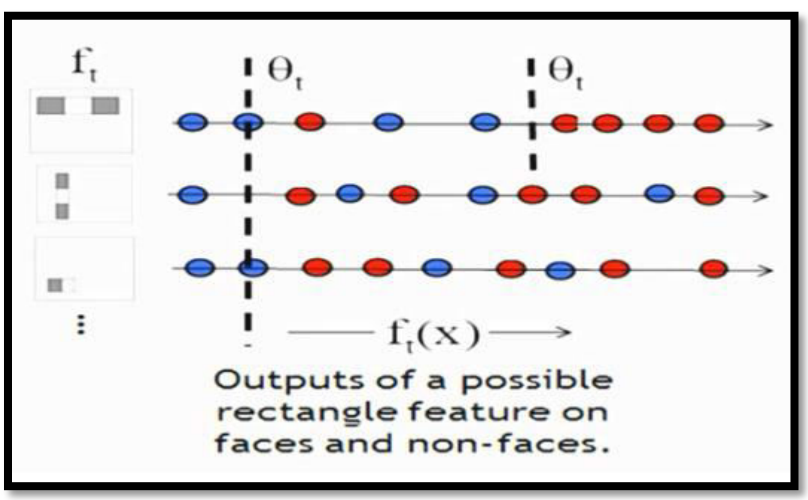

Fig 2.6 Selection of relevant features using Adaboost algorithm

Adaboost finds the single rectangular feature and threshold that best separates the positive (objects) and negative (non-objects) in terms of weighted error. The steps are:

- Consider sample images (x_1,y_1), (x_2,y_2),..............,(x_n,y_n) where y_i=0,1 for positive and negative samples respectively.
- Initialise weights $w_{1,i}$ = 1/2m ,1/2l for y_i=0,1 respectively, where m and l are the number of positives and negatives respectively.
- For t =1,2,..........,T:
    - Normalise weights
    - For each feature j, train a classifier h_j, which is restricted to using a single feature
    - Choose classifier, h_t , with lowest error e_t
    - Update the weights

$$w_{t+1,I} = w_{t,i}\,\beta_t^{\,1-ei} \tag{2.2}$$

where e_i=0 if x_i is classified correctly else 1 and $\beta_t=e_t/(1-e_t)$

- The final strong classifier is

$$C(x)=\begin{cases}1 & \sum_{t=1}^{T}\alpha_t h_t(x)\geq\frac{1}{2}\sum_{t=1}^{T}\alpha_t\\ 0 & \text{otherwise}\end{cases} \tag{2.3}$$

where α_t=log $1/\beta_t$

After these features are found, a weighted combination of all these features is used in evaluating and deciding whether a given window has an object or not. The features will be selected only if they can at least perform better than random guessing (detects more than half the cases).

Training of the system using Adaboost Algorithm:

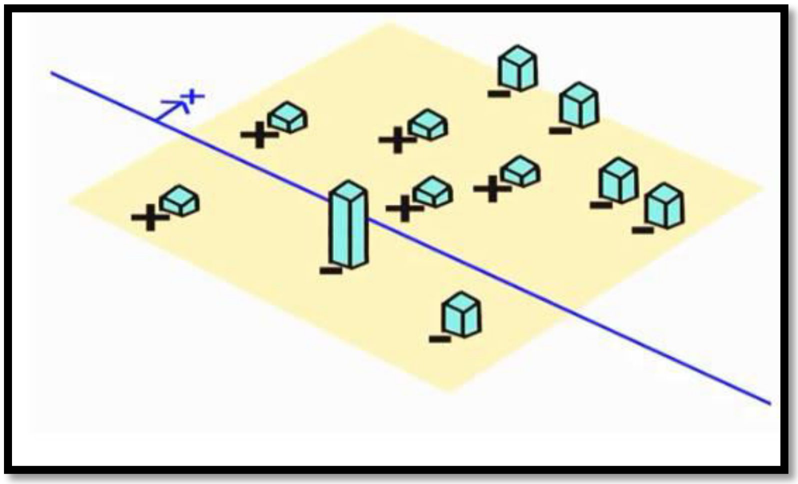

Fig 2.7 Training using Adaboost algorithm

- Adaboost starts with a uniform distribution of "weights" over training examples.
- Select the classifier with the lowest weighted error ("weak" classifier)
- Increase the weights of the training examples that were misclassified
- Repeat
- At the end, make a linear combination of the weak classifiers obtained at all iterations.

These selected features are also called weak classifiers. Adaboost constructs a strong classifier as a linear combination of these weak classifiers.

$$F(x) = \alpha 1 f1(x) + \alpha 2 f2(x) + \ldots\ldots\ldots\ldots \qquad (2.4)$$

Here, $F(x)$ is the strong classifier and $f(x)$ is the weak classifier. $f(x)$ takes values 1 or 0 depending on whether that particular feature is present or not in our image.

If the linear combination exceeds a particular predetermined threshold value, $C(x)$ or $F(x)=1$ which indicates object else it indicates non-object. The equation represented below shows the case for $\boldsymbol{C(x)}$ or $\boldsymbol{F(x) = 1}$ condition, where $\boldsymbol{h(x)}$ is the same as $\boldsymbol{f(x)}$.

$$C(x)=\begin{cases}1 & \sum_{t=1}^{T}\alpha_t h_t(x)\geq\frac{1}{2}\sum_{t=1}^{T}\alpha_t \\ 0 & \text{otherwise}\end{cases} \tag{2.4}$$

The basic principle of Viola Jones object detection algorithm is to scan the Haar features many times through the same image - each time with a new size. The 24X24 window image is divided into sub-windows for the same. Even if an image contains one or more objects, it is obvious that an excessive large amount of the evaluated sub-windows would still be negatives (non-objects). Hence the algorithm should focus on discarding the non-objects more quickly and spend more time on probable object regions. Hence a single strong classifier formed out of the linear combination of all the best features is not good for evaluation if the computational cost is considered.

Therefore, a cascade classifier is used which is composed of stages, each containing a strong classifier. All the features are grouped into stages where each stage has a certain number of features. A hierarchy is assigned to the classifiers for example 10 features in the 1st classifier, 20 in the second, 40 in the next etc. Each stage is used to determine whether a given sub-window is an object or not. It is immediately discarded as a non-object if it fails in any of the stages.

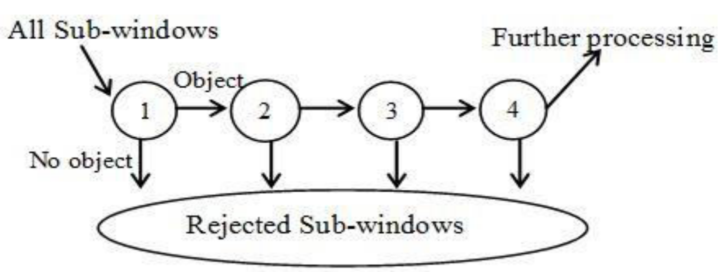

Fig 2.8 Cascade Classifier

Apply the cascade of classifiers to all the sub-windows on the image. If the sub-window passes through all the classifiers, it is detected as an object.

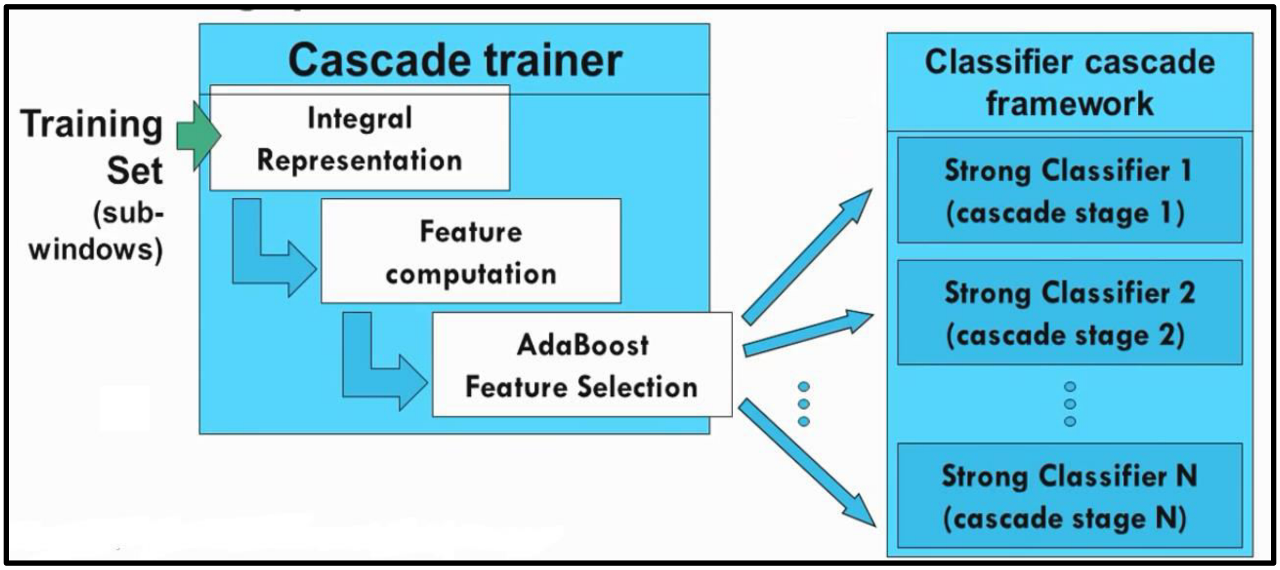

Fig 2.9 Training and detection

2.2.2.3 Object tracking algorithm

Once the object is identified, it needs to be tracked. Mean shift tracking algorithm can be used for the same. The greatest advantage of mean shift over other tracking techniques is its computational efficiency.

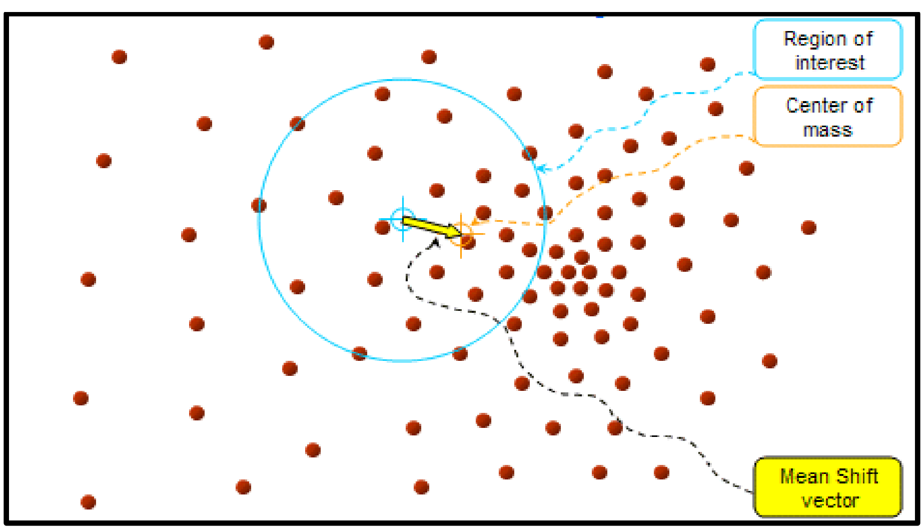

Fig 2.10 Mean shift tracking algorithm

In order to find the densest region in the above case, consider an area bounded by the circle. If the centre of mass of the region doesn't coincide with the centre of the circle, move the circle in such a way that its centre coincides with the centre of mass calculated using moments. Now, find the centre of mass of the new region and check if it coincides with the centre of the circle. Continuing like this in an iterative manner, the densest region will be that region where centre of mass coincides with the centre of the circle. This is the basis of mean shift tracking algorithm.

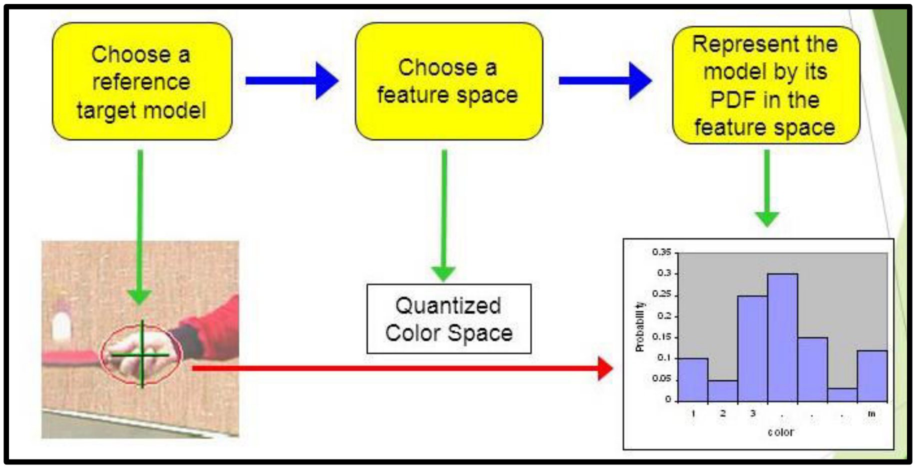

Fig 2.11 Histogram representation

The first step is to choose a reference target model. The second step is to choose a feature space that can be used to identify the feature, such as hue or color. The third step is to represent the model using the feature space as a probability map and calculate the histogram of the region represented by the target model. Figure 2.11 represents the target model, hand inside an ellipse. A quantized color space is chosen to represent the model as a histogram representing the pixel values ranging from 0 to 255 for each colour.

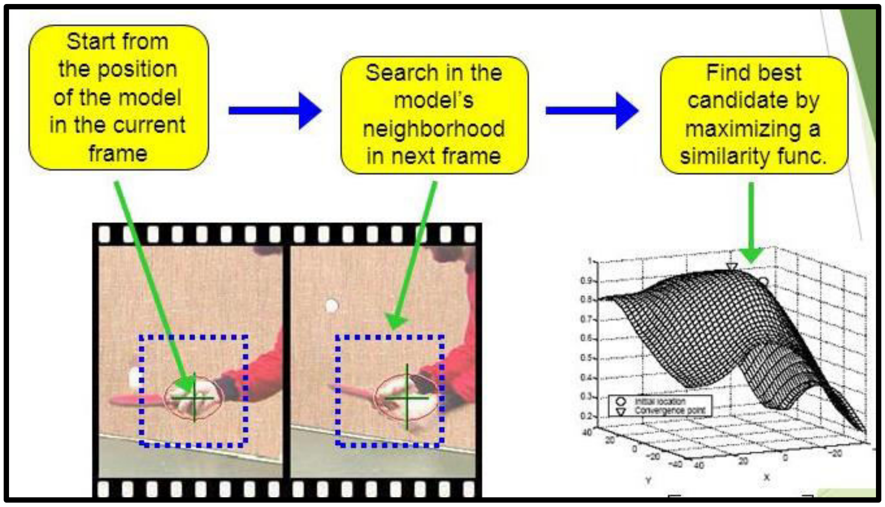

Fig 2.12 Probability map comparison

Now start with the same position in the next frame too. Try to find the probability map of the next frame and apply mean shift tracking to find the position of the object. Hence using mean shift tracking, the reference model moves towards the point where the mean or the centre of mass is located. Hence it converges on to the point where the probability functions have maximum similarity.

This probability map is created using Histogram Back Projection. In the beginning, a window is chosen to track the object. The hue values under the window are extracted and a histogram is formed using these values. This histogram of the first frame is used as a reference and for any new image, the pixel values in it are replaced by the value of reference histogram corresponding to that pixel value. Mean shift is then applied on this back projected image, in order to track the object wherever it moves to.

CAMshift is an algorithm where the window size is continuously adapted according to the size of the object that is being tracked. CAMshift needs to be applied in cases when the object moves forward or backward from the camera, hence causing an increase or decrease in size of the object. The window size also needs to adapt to the size of the object, according to whether it moves forward or backward. For the same, the area of the white region in the back projected image is calculated using sum of all pixels and a new window is calculated using this area. The area of the white region in the back projected image varies according to whether the object moves forward or backward.

Chapter 3

SYSTEM OVERVIEW

3.1 BLOCK DIAGRAM

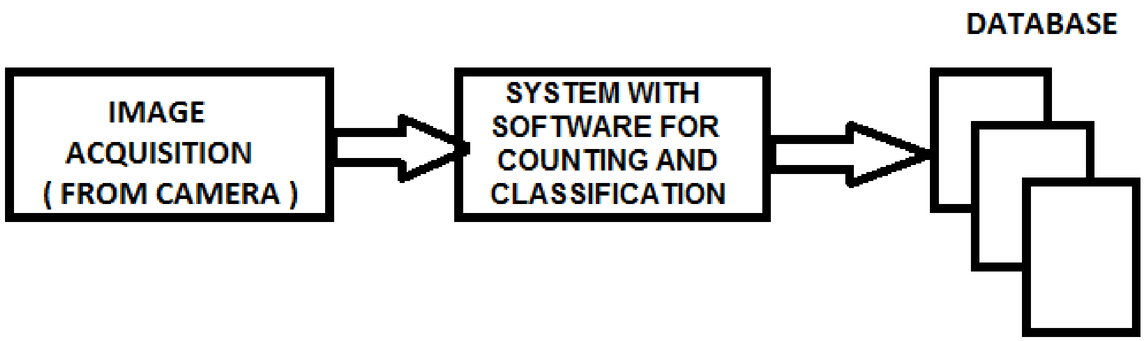

Fig 3.1 Block diagram

3.2 WORKING

The first step is to train the system to identify objects and non-objects. That is, the system should be able to identify cars and human beings in this case. The training can be done using Viola Jones Detection Algorithm by a giving a set of sample objects and non-objects and then extracting the features of objects alone and storing it in separate files in order to distinguish it from non-objects. This is the training phase.

The first step in actual implementation is image acquisition from a video camera. The video from a high resolution HD camera is fed to a system with appropriate software. The video consists of various frames and these frames are extracted and stored in files. Then using Haar Classifier algorithm, human beings and cars are detected and classified. After detection these objects needs to be tracked using mean shift tracking algorithm and histogram back projection technique. Simultaneously the count of each is automatically updated in the database. Figure 3.1 shows the schematic representation of the project.

Chapter 4
SYSTEM SPECIFICATIONS

4.1 HARDWARE USED

1080p HD Camera - High resolution

4.2 SOFTWARES USED

MATLAB

MATLAB is an object-oriented high-level interactive software package for scientific and engineering numerical computations. Its name stands for matrix laboratory. MATLAB enables easy manipulation of matrix and other computations without the need for traditional programming. MATLAB's basic data element is the matrix. It integrates numerical analysis, matrix computation, signal processing, and graphics into an easy-to-use environment where problems and solutions are expressed just as they are written mathematically, without much traditional programming. Using MATLAB, one can analyze data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enable to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java.

4.3 PLATFORMS USED

- Personal Computer
- Windows Operating System

Chapter 5

DESIGN DETAILS

5.1 FLOWCHART

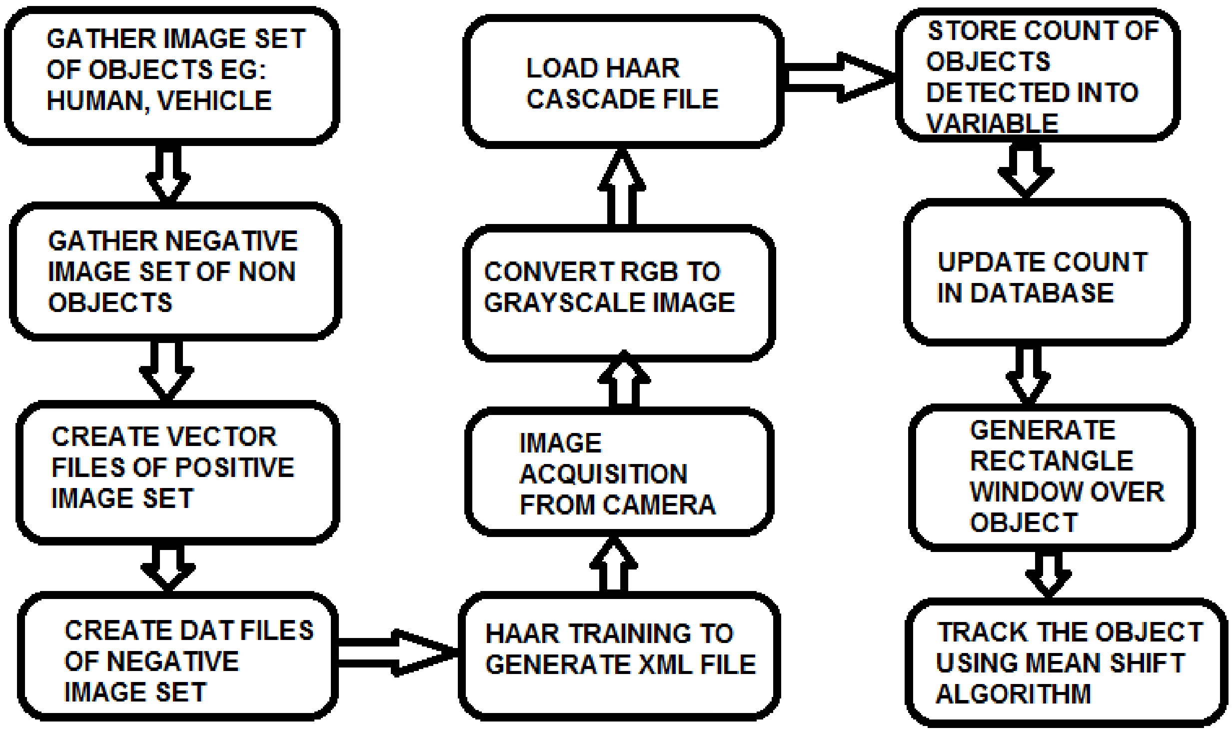

Fig 5.1 Flowchart

5.2 ALGORITHM

- Collect image set of particular object, eg : Human, Vehicle
- Collect negative image set of non-objects
- Create positive vector files using MATLAB for generating variations in image set
- Create collection file of the format .dat of negative images
- Train positive and negative image sets to generate xml file called cascade classifier to detect objects
- Use MATLAB to perform real time detection in video
- Use MATLAB functions for image acquisition from camera into a file

- Convert RGB image to grayscale image
- Load the Haar cascade xml file to detect objects in a new input image
- Store the count of objects detected into a variable
- Update the database with the count of objects detected
- Generate rectangle window on the region of interest of object using MATLAB functions
- Use mean shift algorithm for tracking the object
- Use CAMshift for adjusting the rectangular window size

REFERENCES AND LINKS

[1] Mrinali M. Bhajibhakare, Pradeep K. Deshmukh, “To Detect and Track Moving Object for Surveillance System”, International Journal of Innovative Research in Computer and Communication Engineering, Volume 1, Issue 4, June 2013.

[2] Nilesh J Uke, Ravindra C Thool, “Moving Vehicle Detection for Measuring Traffic Count Using OpenCV”, Sinhgad College of Engineering, Department of Information Technology Pune, India.

[3] Sunil Tiwari N, Ravikumar A.V, “Multiple moving object detection and tracking using Haar features with smart video surveillance system”, International Journal of Engineering Research and Technology, Volume 3, Issue 6, June 2014.

[4] Rajashree Tripathy, R N Daschoudhury, “Real-time face detection and tracking using Haar classifier on SoC”, International Journal of Electronics and Computer Science Engineering, Volume 3, Number 2

[5] Gregory Shaknarovich, Paul A. Viola, Baback Mogaddam, “A unified learning framework for real time face detection and classification”, Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR’02), May 2002.

[6] Paul Viola, Michael J Jones, Daniel Snow, “Detecting pedestrians using patterns of motion and appearance”, Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV’03), April 2003.

[7] http://www.slideshare.net/

[8] www.google.com

[9] www.youtube.com

[10] www.mathworks.com