MAJOR PROJECT REPORT

ON

AUTOMATED LONG-TERM OBJECT TRACKING WITH ONLINE LEARNING

A Dissertation Submitted for the Partial Fulfillment of the Degree

MASTER OF TECHNOLOGY

IN

SIGNAL PROCESSING AND DIGITAL DESIGN

BY

ABHILASHA YADAV

2K15/SPD/03

Under the Guidance of

Sh. RAJESH ROHILLA

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY, DELHI-110042

(Formerly Delhi College of Engineering)

(SESSION 2015-2017)

DELHI TECHNOLOGICAL UNIVERSITY

Established by Govt. of Delhi vide Act 6 of 2009
(Formerly Delhi College of Engineering)

CERTIFICATE

This is to certify that the thesis entitled "**Automated Long-term Object Tracking with Online Learning**" being submitted by **Abhilasha Yadav**, 2K15/SPD/03 for partial fulfillment of the degree "Master of Technology" in "Signal Processing and Digital Design" from Delhi Technological University is based on work carried out by Abhilasha Yadav under my guidance and supervision. The matter contained in this thesis has not been submitted elsewhere for award of any other degree to the best of my knowledge and belief.

Sh. Rajesh Rohilla
(Associate Professor)
Department of ECE
Delhi Technological University
Delhi-110042

ACKNOWLEDGEMENT

I would like to express my heartily gratitude and thanks to my project guide **Sh. Rajesh Rohilla**, Associate Professor in Department of Electronics and Communication Engineering, Delhi Technological University, for continuous inspiration, encouragement and guidance in every stage of preparation of this thesis work.

I am also extremely thankful to **Dr. S. Indu**, Head of the Department of Electronics and Communication Engineering, Delhi Technological University, for the support provided by her during the entire duration of degree course and especially in this thesis.

I would also like to thank my batch mates and friends for their support throughout the entire duration of the degree.

Abhilasha Yadav
2K15/SPD/03
Department of Electronics and Communication Engineering
Delhi Technological University

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This thesis work looks into automatic detection as well as long-term tracking of any unknown object in a video sequence. Every object is described by position and area covered in any particular frame. Bounding box defines the object of interest in the first frame. For automatic detection of object, Gaussian-Mixture-Model for background subtraction is used, this makes the system suitable for use in automatic surveillance and monitoring. In consecutive frames, objective is to find objects position and area or to point out objects absence when not present. This tracking approach fragments the long-term tracking task into simpler subtasks of tracking-learning-detection. The tracker tracks object in every consecutive frame. Detector localizes every appearance that is observed so far and makes tracker error free by correcting when necessary. Neither tracking nor detection can single-handedly give solution to the long-term tracking problem. Learning removes the detector's errors and also updates the detector to overcome future errors. This work studies way to find detector's error along with learning from it. The novel online learning approach (P-N learning) removes errors by pair of 'experts': a) P-expert finds out the missed detections b) N-experts finds out false alarms. The process of learning is semi-supervised learning with a set of labelled data and we need to label the unlabelled one. The TLD framework along with P-N learning is described. This method is different in the way that here the classifier is trained online, hence this method is suitable for tracking any unknown object.

The outcome is real time tracking which enhances with time. This framework is advertised under Predator i.e., a smart camera that learns with time.

CHAPTER 1

INTRODUCTION

The purpose of object tracking is identification of target objects from frame to frame. Object tracking is appropriate for various tasks, some of them are: security and surveillance, human identification, human computer interaction, autonomous navigation, monitoring of traffic, augmented reality, automated Google cars, games (Kinect) and many more.

1.1 CHALLENGES

Object tracking is a complex process and a challenging problem, incorporated with various difficulties that are generated because of abrupt motion of object as well as camera, change in appearance patterns of both the object as well as the background, complex object shapes, illumination changes, background clutter and occlusions.

The main challenges faced by long-term tracking is the detecting of object as it reappears in camera field of view and the problem is further exaggerated by the reality that object changes its appearance making the first frame appearance immaterial.

1.2 GOAL

Taking an unconstrained video stream into consideration in which objects moves inside and outside the view of camera and may change in its appearance significantly or may get partial or full occlusion. Object is described by position and area covered in any particular frame. Bounding box defines the object of interest in the first frame. Our objective is to automatically find objects position and area (objects bounding box) or to point out object is not visible when object is absent in consecutive frames. When the video sequence is processed at the frame rate and this process will run for indefinitely longer time, we refer this task as *long-term tracking* problem.

1.3 MOTIVATION

This thesis research is mainly inspired by real-time, automatic and interactive applications. ***"A real-time system is one in which the correctness of the system depends not only on the logical result of computation, but also on the time at which the results are generated."***

- Human-computer interaction (HCI): It provides interface of computer with users through gesture recognition based on hard-coded rules. Long-term trackers provide an advantage that one can imagine of being a personalized controller using gestures or using objects selected at the runtime.
- Digital surveillance systems: The systems continuously generate a large amount of video data. To analyse this huge data with human inspection is a tiresome work. Hence there is a requirement of automatic detection of threats and keep track on these objects when the reappear.
- Human-robot interaction (HRI): It allows to build systems that can work outside from laboratory, in these systems there long-term interaction of users with environment and robots are designed to help individuals.
- Automated Google cars: These are self-driving cars on public roads. Long-term tracking is required here because many times vehicles move inside and outside the field of view of camera and we need to detect he reappeared vehicle.
- Surveillance: To monitor the presence, absence and tracking of a particular object or individual in a video sequence long-term tracker is required. This adds more security in the surveillance process.
- Object-centric stabilisation: In hand-held camera where the user selects any arbitrary object automatic adjustment of camera settings, long-term tracker will enable user to restart the stabilisation when object reappears in field of view. Utility of this application is when observing distant object through digital zoom known as the auto focus problem.

Hence long-term tracking methods can be applied to solve these problems it has a wide range of applications and is of great interest.

1.4 CONTRIBUTION

The first contribution of this thesis work is automatic detection of objects. In the surveillance systems for the monitoring of traffic and airports CCTV cameras are widely used. In order to monitor the objects of interest from the surrounding, automatic detection of the foreground objects must be there, also these objects must be re-tracked when the again come into the camera view. To make the detection process of the objects automatic, this thesis work used

Mixture of Gaussians to model the background and detect the moving objects captured from stationary camera.

The long-term tracking is approached from either of two perspectives tracking or detection. Tracking method approximates the motion of object they require the initialization of object and they result in smooth routes. It generates the track of the object for all instances in the video by locating the position of object. But it fails when the object moves out of camera view i.e. when it disappears and results in building up of error during the runtime (drift).

Detecting methods approximates object location in each frame separately. It treats every frame to be independent. It does not result in drifting and does not fail when the object moves out of camera view. But they need offline training phase and method is not applicable to unknown objects. Hence neither tracking nor detection can single-handedly give solution to the long-term tracking problem. This leads to the design to a new framework in which both these methods operate concurrently using the benefits of one another. Runtime detection process is improved by providing labelled training data by the tracker and thus the detector is able to reset the tracker and minimize the tracking breakdown.

Object detectors were trained assuming the fact that all training examples are well labelled (supervised learning). However, when wish to train the detector with help of single labelled example and the complete video stream. Semi-supervised learning exploits both labelled and unlabeled data.

Second contribution of this novel approach (TLD) is that it fragments the long term tracking task into simpler subtasks of tracking-learning-detection which all operate simultaneously. Tracker tracks object in every consecutive frame. Detector localizes every appearance that is observed so far and makes tracker error free by correcting when necessary. Learning removes the detector's errors and also updates the detector to overcome future errors.

Different information sources are used to get robust learning. For example, in a particular single patch where the objects location is denoted, this patch gives the definition of the appearance of object as well as the neighbouring patches will give the definition how the background appears, so while tracking any patch, we consider both the object and background appearance where we effectively exploit information during the learning process.

Another contribution of this project is the new online learning paradigm known as P-N learning. It is a semi-supervised learning for detection of objects from the video. The aim of this unit is to enhance the efficiency of object detector by online processing of video sequence. This novel machine learning approach (PN learning) removes errors with the help of pair of 'experts': a) P-expert which finds out the missed detections b) N-experts which finds out false alarms. These experts make the errors themselves. However, their independence in making errors enables mutual cancellation of their errors which leads to sane and sensible learning.

Figure-1.1 Long-term tracking. Bounding box defines object position, proposed system does tracking-learning-detection and red dot is indication of absence of object.

CHAPTER 2

RELATED WORK

The video stream is when run at the frame rate and the process runs for indefinitely longer time, we define this work as *long-term tracking* problem.

Tracking tracks object in every consecutive frame. Detection localizes every appearance that is observed so far and makes tracker error free by correcting when necessary. Machine learning in many times employed with both the approaches where the tracker uses machine learning to adapt with the changed in the object appearance. Detector also use machine learning techniques for building better models that covers many appearance of object. Overview of the approaches is mentioned next.

2.1 OBJECT TRACKING

Object tracking is identification of target objects from frame to frame and estimation of its motion. Object is represented by its shape and appearance. To represent any object we choose suitable representation technique:

- *Points*: Representation of object by point or the centroid, is suitable for objects with small regions in any image.
- *Primitive geometric shapes*: usually geometric shapes like rectangle or ellipse represent the object and is suited for the representation of rigid objects.
- *Object silhouette or contour*: boundary of any object is called contour, it is suitable for the tracking of objects with complex non rigid objects.
- *Articulated shapes:* composes of the body parts held together by joints for example the human body. Kinematic motion models use this object representation.
- *Silhouette*: outer boundary of object .It is the contour around the object.

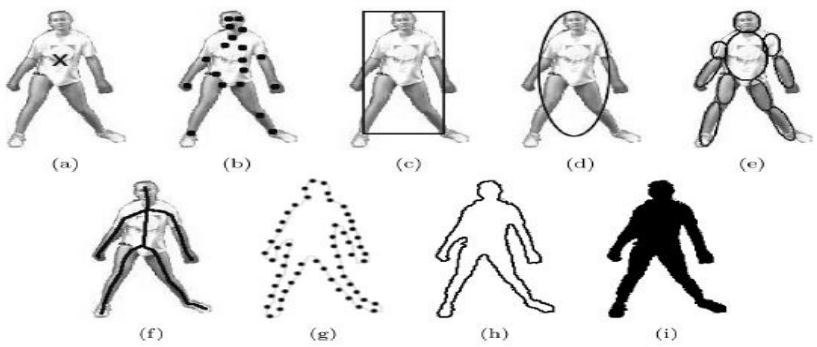


Figure- 2.1 Various representation methods for an object

Tracker aims to generate the track of an object as it moves with time by locating its position in frames of the video.

Template-match tracking is an approach used for to estimate object in between the consecutive frames. Objects description lies in object appearance and location.

Target template gives information about appearance of image patch also its coloured histogram, along with this it also gives motion which is denoted by transformation which minimizes the mismatch between candidate and target template. Two types of template tracking are there: 1) static template (here target template remains unchanged) and 2) adaptive template (target extraction from previous frames). Template are limited to fewer modelling capabilities as they have single look of object.

In order to model more appearances and its variations **generative models** are used where environment modelling is also done. Environment supports object movement and is correlated with region of interest, also it is considered to be negative class, and tracker discriminates against it. Another approach is use of **discriminative models** which builds up classifier that represent decision plane of boundaries in between the object and background. The static-discriminative-tracker trains the classifier before the process of tracking. **Adaptive-discriminative-tracker** builds classifier during the process of tracking. The most essential component of this tracking method is update. The positive and negative samples are used for the updating of classifier in each and every frame. This method handle short-term occlusions, changes in appearance and cluttered background and drastically suffer from drift when object moves outside the view of camera for longer duration. To handle this problem a pair of independent classifiers are used.

2.2 OBJECT DETECTION

Object detection is the process in which localization of the object is done for a given input image. The description of an “object” varies from image to image. An object be a single instance or an entire class of objects. Object detection methods are dependent on the features of local image or a sliding window.

The feature-based approach usually follows the order of:

1) Detection of feature

2) Recognition of feature and

3) Model fitting.

Planarity, or a full 3D model is typically exploited. These algorithms reach a level of maturity and operate in real time even on low-power devices and also enables the detection of a large number of objects in the input image. The main strength, as well as the limitation of this approach is the detection of features of the input image and the requirement of knowing the geometry of the object in advance.

The approach based on sliding window scans the input image by a window of various sizes and for every window decides whether the underlying patch contains the object of interest or not. For example, in a QVGA frame, there are around 50,000 patches that are scanned in each frame. In order to achieve real-time performance, detectors based on sliding window adopted the cascaded architecture.

By exploiting the fact that the background is more frequent than the object, a classifier is separated into a number of stages, each of which enables early rejection of background patches, thus reducing the number of stages that have to be evaluated on an average. The training of such detectors requires a large number of training examples and very intensive computation in the training stage to represent the boundary deciding the object and the background accurately. An alternative approach is based upon modelling the object as a collection of templates. In this case, learning involves just the addition of one more template.

2.3 MACHINE LEARNING

Traditionally, the object detectors are trained taking the assumption that all of the training examples are labelled. This assumption is very strong in our case since we want to train the detector from a single labelled example and a video stream. This problem can be formulated as a semi-supervised learning that exploits both the labelled and the unlabelled data. These methods typically assume independent and identically distributed data with certain properties, such as that the unlabelled examples form “natural” clusters in the feature space. Many algorithms based on similar assumptions are proposed, including the Expectation-Maximization (EM), Self-learning, and Co-training.

Expectation-Maximization is an old and basic method to find the estimates of model parameters in a given unlabelled data. EM is an iterative process which alternates between estimation of soft labels of the unlabelled data and training the classifier, in case of binary classification. EM technique was earlier successfully applied to document classification and learning of object categories. In semi-supervised learning terminology, the EM algorithm relies on the “low-density separation” assumption, which means that the classes are well separated. EM is sometimes interpreted as a “soft” version of self-learning.

Self-learning starts by training an initial classifier from a labelled training set; the classifier is then evaluated on the unlabelled data.

Co-training where one classifiers learns from the other.

There are many approaches derived through the combination of tracking-learning –detection for example:

- Sometimes offline training of detector is done in order to get correct trajectory output of tracker and in case the trajectory is not correct an image search is performed on whole image to search the target.
- In some methods the detector is integrated with particle filter, particle filter does the tracking.

These methods are required to have offline training and the detector remains same throughout the runtime. To make the process more generalized online training approach is used where the real-time processing can be done.

CHAPTER 3

AUTOMATIC DETECTION USING GMM

In surveillance systems for monitoring of traffic and airports CCTV cameras are used widely. For intelligent monitoring of objects of interest from the surrounding, there must be automatic detection of foreground objects. To make the detection process of object automatic, mixture of Gaussian's is used to model the background and detect moving objects in static cameras.

3.1 STEPS FOR AUTOMATIC DETECTION

Detection is the prior step to before carrying out other sophisticated tasks such as classification or tracking. GMM focuses on detection of objects.

- The process starts with obtaining the initial frame in which segmentation of moving objects is done from background. In order to initialize GMM certain number of frames are used.
- The process of foreground segmentation is not so perfect because it contains undesirable noise. Morphological opening is used to remove noise and also filling the gaps in the object detected as the objects moves all together. Opening is dilation done on erosion of set A by a kernel that is structuring element B. Together with closing, it is workhorse for removal of noise in applications of computer vision and image processing. Opening operation removes small objects from foreground(the foreground is considered to be bright pixels) in image, and placing them in the background ,closing removes small holes that are present in foreground by making change in small islands of the background to foreground.
- Next step is drawing the contour boundaries around the objects detected. After drawing the boundaries we need to draw bounding box of minimum area that encloses the boundary.
- Smaller contours with area less than a threshold are automatically discarded.
- Now after getting bounding box get the top left and right bottom x and y coordinates that used to initialise the object of interest. It works as labelled frame.

- Display the result obtained and also get the frame number where the object was first successfully detected.

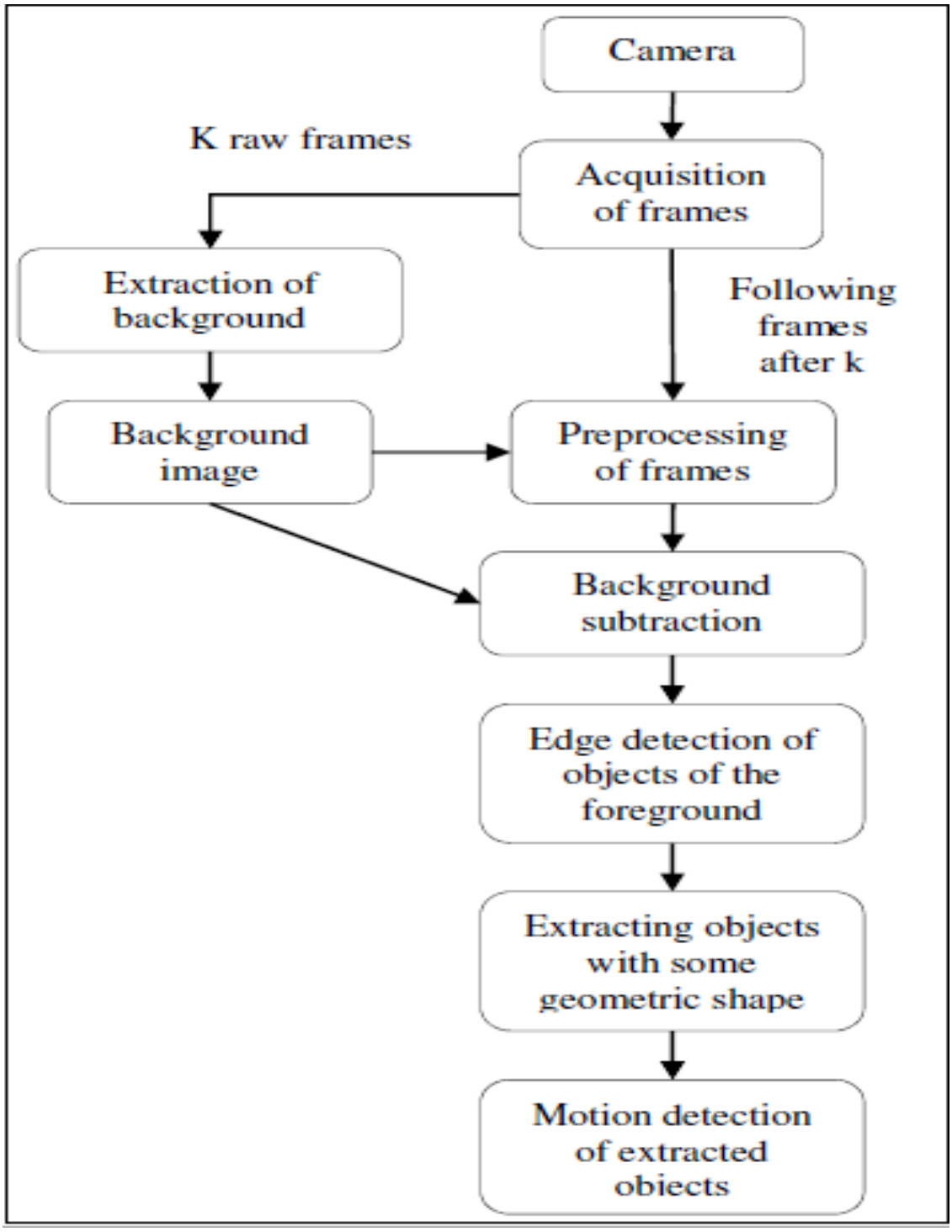


Figure-3.1 Flow Chart of automatic detection

3.2 GMM USED FOR BACKGROUND SUBTRACTION:

Background is modelled with the help of mixture of distinct Gaussians that correspond to various background objects. For the detection of foreground every pixel in the image is compared with each Gaussian and classification is done according to the respective Gaussian. In Gaussian mixture model, for learning gradual changes with time, in each and every pixel of the image is taken as a Gaussian distribution. The model parameters are mean μ(x,y) and covariance Σ (x,y) which are learned from the consecutive video frames. All the pixels are evaluated for their probability is they are included in the foreground or the background.

$$P(X_t) = \sum_{i=1}^{K} W_i, \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

• X_t is current pixel of t[th] frame.

• *K* is the count of distributions used in the mixture.

• W_i, t is the weight of the k^{th} distribution of t[th] frame.

• $\mu_{i,t}$ is the mean of the k^{th} distribution of t[th] frame.

• $\Sigma_{i,t}$ is the standard deviation.

$\eta(X_t, \mu_{i,t}, \Sigma_{i,t})$ is pdf which is Gaussian:

$$\eta(Xt, \mu, \Sigma) = \sqrt{\frac{1}{2\pi\Sigma}} \exp(\frac{-1}{2}(X_t - \mu)\Sigma'(X_t - \mu))$$

The RGB is uncorrelated and hence difference in the value of intensity possess uniform value of standard deviation. The covariance matrix is : $\Sigma_{i,t} = \sigma_{i,t}^2 I$

If the Gaussian is greater than a pre-decided threshold it is classified to be the part of background model, otherwise it will be classified to be foreground.

When the pixel matches with any of the Gaussians then value of w, μ and σ are updated.

$$W_{i,t+1} = (1 - \alpha)W_{i,t} + \alpha$$

$$\mu_{i,t+1} = (1 - \rho)\mu_{i,t} + \rho . X_{t+1}$$

$$\sigma_{i,t+1}^2 = (1-\rho)\sigma_{i,t}^2 + \rho(X_{t+1} - \mu_{i,t+1})\,(X_{t+1} - \mu_{i,t+1})'$$

$$where, \rho = \alpha \times \eta\left(X_{t+1}, \mu_i, \Sigma_{i,}\right)$$

if Gaussian do not match then only w is updated.

$$W_{i,t+1} = (1-\alpha)W_{i,t}$$

In GMM, the pixels of the current frame are checked against background model by making its comparison with every Gaussian used in the model unless and until a matching Gaussian is found. In case a match is found, the mean value and variance value of the matched Gaussian will be updated, else a new Gaussian with the mean value equal to the current pixel colour and some initial variance will be introduced into mixture model. Each pixel is then classified depending on whether the matched distribution is representing the background. Background subtraction improves the tracking abilities and makes the system more accurate and error free.

CHAPTER 4

TRAINING-LEARNING-DETECTION SETUP AND P-N LEARNIING

4.1 TLD FRAMEWORK

To address to the problem of long-term tracking of any unknown object in video sequence TLD setup is used. The flow diagram is shown in figure.

Tracking algorithm approximates the motion of object they require the initialization of object and they result in smooth routes. But it fails when the object moves out of camera view i.e. when it disappears and results in building up of error during the runtime (drift).

Detection algorithms treat every frame of video stream to be independent and do the scanning operation on the full image to find the location of object and thus localizes the appearance. It does not result in drifting and does not fail when the object moves outside view of camera. But they need offline training phase and method is not applicable to unknown objects. They result in two types of errors: false positives and also false negatives.

Since neither tracking nor detection can single-handedly give solution to the long-term tracking problem. Learning method do following works:

- Learning monitors the execution of both tracker and the detector.
- Learning approximates the errors of detector.
- Learning induces training examples in order to overcome the above errors in coming future.

Learning element is based on assumption that during the process both tracker and detector may fail. With the integration of learning in the algorithm, the detector can now generalise to more appearances of the objects and also can easily distinguish against the background. Main objective of learning is that when we are given with a single patch from the video, we need to concurrently learn the object classifier as well as make the correct labelling of patch as "object" or "background".

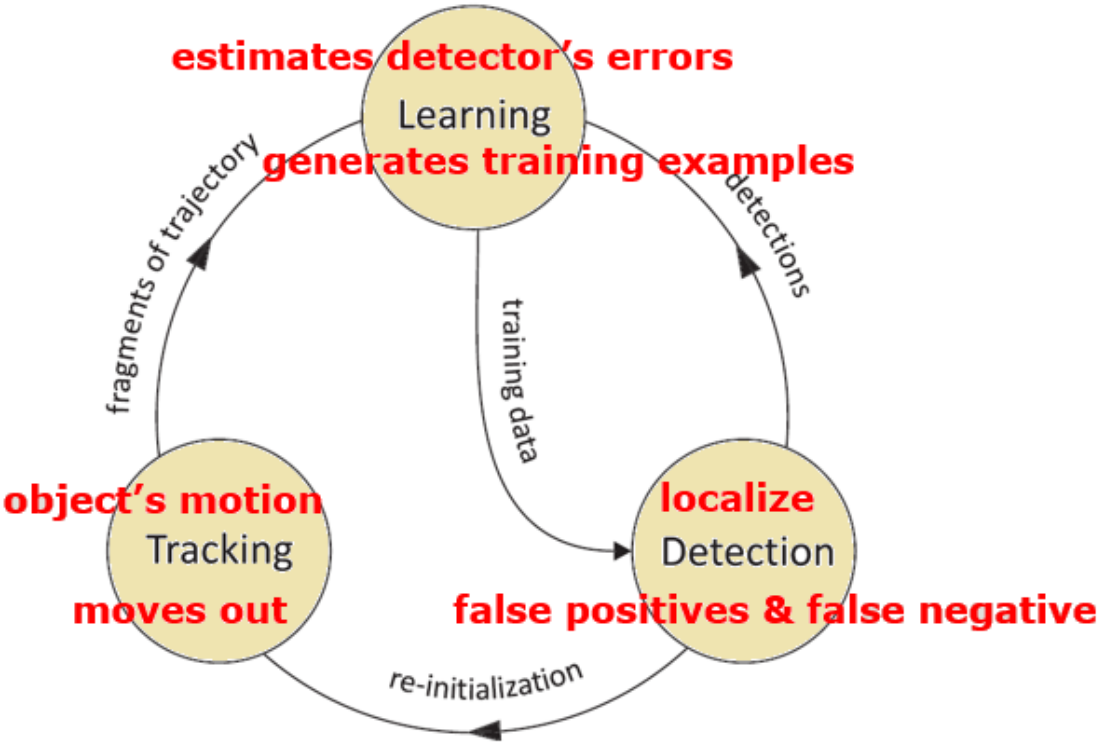


Figure-4.1 TLD Framework

4.2 P-N LEARNING

Learning component of the TLD setup is explored out in this section. The new online learning paradigm known as P-N learning, is a semi-supervised learning for detection of objects from the video. Main objective of learning is that when we are given with a single patch from the video, we need to concurrently learn the object classifier as well as make the correct labelling of patch as "object" or "background".The aim of the module is to enhance the efficiency of object detector by doing online process of the video sequence. This novel machine learning approach (PN learning) removes detector errors with the help of pair of 'experts': a) P-expert removes missed detections i.e. it identifies the false negatives b) N-experts removes false alarms i.e. it identifies the false positives. These experts make the errors themselves. However, their independence in making errors enables mutual cancellation of their errors which leads to sane and sensible learning.

4.2.1 BLOCK DIAGRAM :

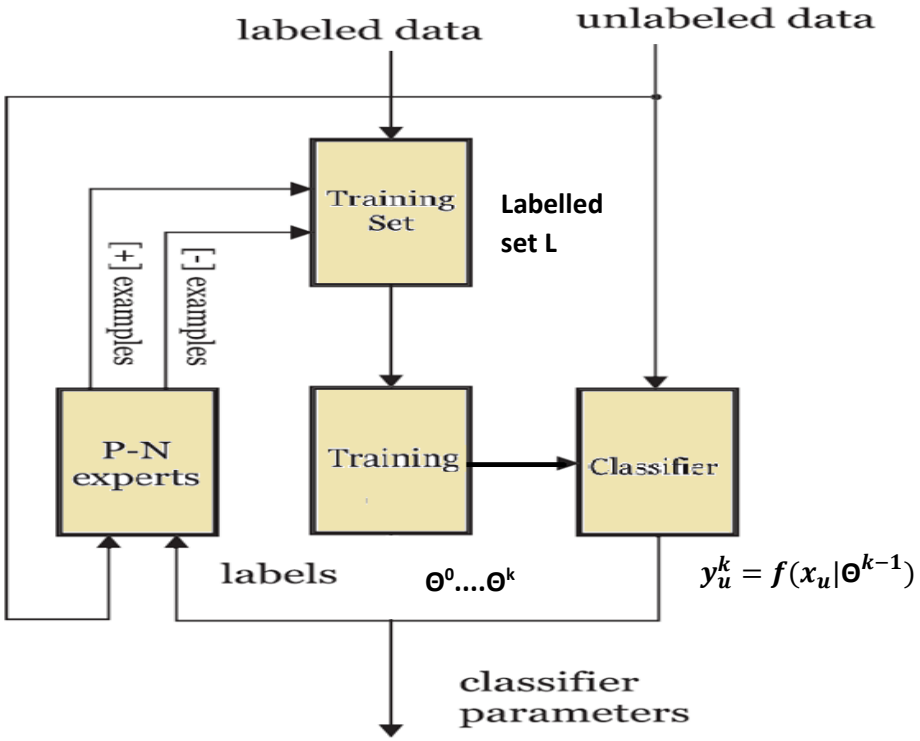


Figure- 4.2 The block diagram of P-N learning algorithm on detector classifier.

4.2.2 ALGORITHM

Train a classifier using all labelled data available.

Iterate

{

(1) Classify unlabelled data

(2) Discover structure in the data (e.g. track the patch)

(3) Apply P-constraints to generate positive data (false negatives)

(4) Apply N-constraints to generate negative data (false positive)

(5) Update classifier

}

4.2.3 P-N LEARNING A SEMI-SUPERVISED LEARING METHOD SETUP:

Let us assume,

x: an example taken from a feature space X (unlabeled set)

y: a label taken from a space of labels Y = {-1,1} (a set of labels)

L = {(x,y)} : a labeled set

Input: a labeled set L_l and an unlabeled set X_u, where l << u

The function of P-N learning is to learn a classifier

f: X→Y from labeled set L_l and *bootstrap its performance by the unlabeled set X_u

Classifier f is a function from a family F, family F is regarded to be fixed in training which are corresponding to calculation of the parameters Θ.

Blocks of P-N learning are:

- *Classifier* which will learn.
- *Training set*- which consist of the labelled training examples from the detector.
- *Supervised training*- that is used to train the classifier from the set of training set.

- *PN EXPERTS* - It develops positive as well as negative examples during the training.

Initialization of the training process begins with placing labelled set L into training set. Supervised learning is then used to train the classifier and calculate initial parameter Θ^0. Continuous bootstrapping is done in the learning process. In the k^{th} iteration, previously trained classifier in last iterations id used to classify the unlabelled set, $\boldsymbol{y_u^k = f(x_u|\Theta^{k-1})}$ for all values of x_U belonging to X_U.

The analysis of the classification is done by P-N expert rules, which finds out examples whose classification is not done correctly. Incorrectly classified examples are added to the training set after changing the labels to correct ones. This iterative process finishes up by retraining the classifier and calculation of parameter Θ^k.

Estimation of the errors of the classifier is the most decision making part of learning. Separation of false positives and negatives is the key idea. The unlabelled set is divided into two halves and depending upon the current classification every part is analysed by an independent expert. False negatives are estimated by P-experts and are added to the training set along with positive label $N^+(k)$ which increases the generalization of appearances of object in detector.

N-expert is used to estimate false positives, and will add them to training set along with negative label $N^-(k)$ which increases discriminability.

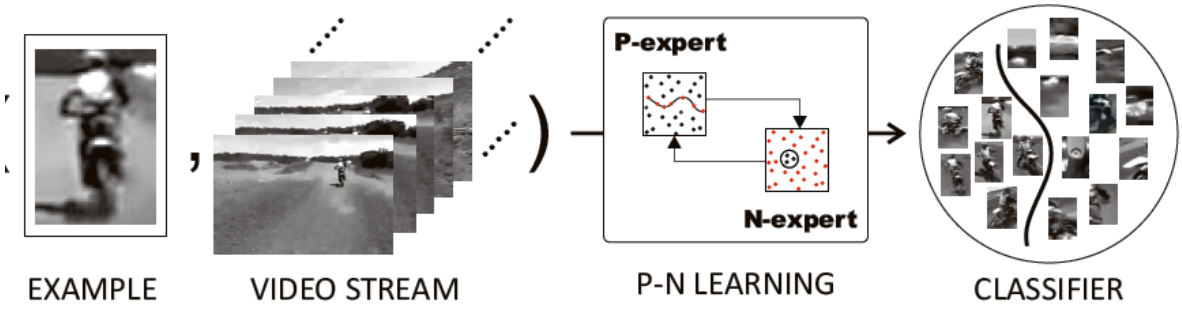


Figure-4.3 Classification of an image patch by P-N expert

SUPERVISED BOOTSTRAPING AND P-N LEARNING:

Bootstrapping is machine learning ensemble algorithm that is designed to enhance the stability of machine learning algorithm used for classification. It reduces the value of variance and thus helps in overcoming over-fitting. P-N learning generalization of bootstrapping, assuming that X_u is labelled we can directly do the recognition of misclassified examples and add correct labels to them such a strategy is called supervised bootstrapping.

In order to assess the similarity match of a tracking-detecting pair, we train up a boosted classifier which has weak learners for the tracking of each target. Classifier is then trained online on one target against all others. Patches which are used as positive training examples are sampled from the bounding box of the associated detection. The negative training set is sampled from nearby targets, augmented by background patches. Classifier updated only on detections that are non-overlapping. After each update step, we keep a constant number of the most discriminative weak learners. The output of the classifier is linearly scaled to the range {−1, 1 }. The weak learners (feature types) are selected by evaluating the classifier for different combinations of colour and appearance features.

4.2.4 IMPACT OF P-N LEARNING ON CLASSIFIER PERFORMANCE

Let us as classifier (eg. Nearest Neighbour) whose performance is measured on X_u which is unlabelled set. Initially outcome of this classifier is at random and then correction is made for those examples that are returned by the P-N experts.

To do analysis assume X_u to be known, this allows the measurement of errors made by classifier as well as the P-N experts.

k indicates iteration of training.

Classifier outputs >> false positives: α(k) false negatives : β(k)

P-expert outputs >> correct: $N_c^+(k)$ false: $N_f^+(k)$

This forces the classifier to change $N^+(k) = N(k) + N_f^+(k)$ negatively classified examples to positive. In next iteration false positive errors of the classifier thus become

$\alpha(k+1) = \alpha(k) - N_c^-(k) + N_f^+(k)$

Above equation shows that false positives α(k) decrease if $N_c^-(k) > N_f^+(k)$ i.e. ,when the number of examples that were correctly relabelled to negative is higher than the number of

Examples that were incorrectly relabelled to positive.

N-expert outputs >> correct: $N_c^-(k)$ false: $N_f^-(k)$

This forces the classifier to change $N^-(k) = N_c^-(k) + N_f^-(k)$ positively classified examples to negative. In next iteration false negative errors of the classifier thus become

$\beta(k+1) = \beta(k) - N_c^+(k) + N_f^-(k)$

Above equation shows that false negatives β(k) decrease if $N_c^+(k) > N_f^-(k)$

4.2.5 QUALITY MEASURES:

P-precision measures authenticity of positive labels, checks its reliability, : $P^+ = N_c^+/(N_c^+ + N_f^+)$

- P-recall measures percentage of recognized false negative errors : $R^+ = N_c^+ / \beta$
- N-precision measures authenticity of negative labels : $P^- = N_c^- /(N_c^- + N_f^-)\backslash$
- N-recall measures percentage of recognized false positive errors: $R^- = N_c^- / \alpha$

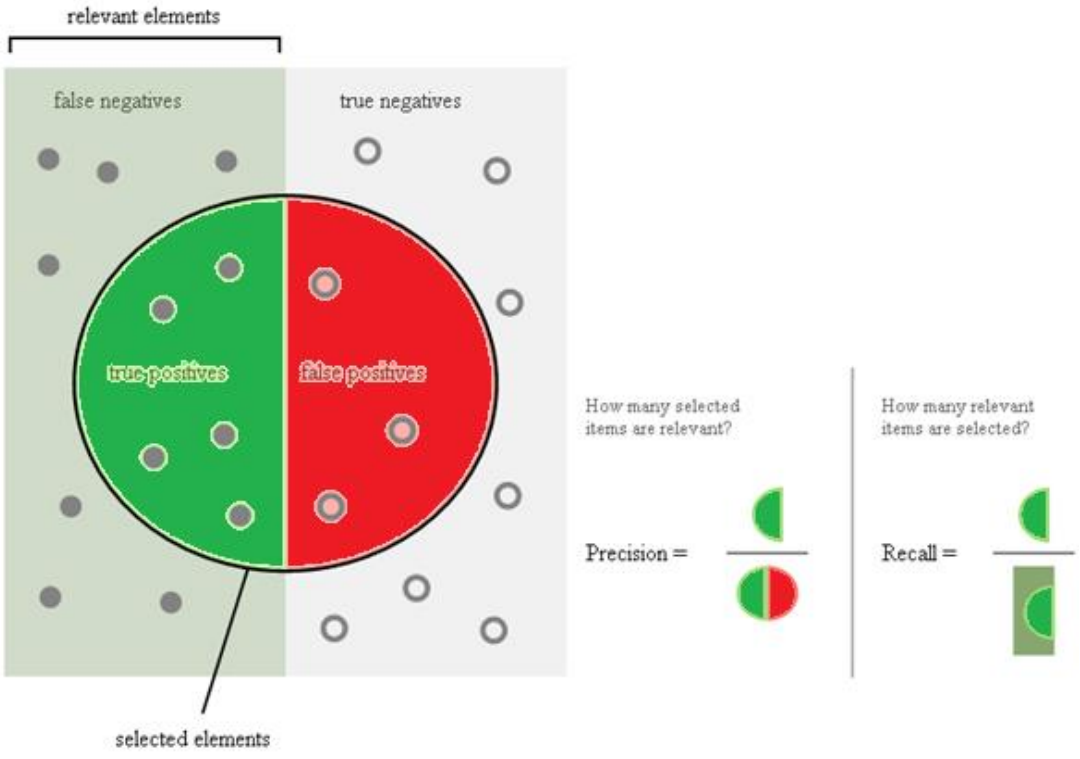


Figure-4.4 Quality measures, Precision and Recall

4.3 WORKING OF ONLINE LEARNING COMPONENT

P-N learning is applied on object detector to train it by using labelled frame automatically obtained from the foreground detection part using GMM. Detector has scanning window that selects patches at various scales and shifts, a binary classifier that classifies the unlabelled data X_u obtained from the video stream and training examples that are obtained from image patches. X_l labelled examples are obtained from labelled frame.

4.3.1 INITIALIZATION

Initialization of PN learning is by supervised training of detector, known as initial detector.

The positive examples for training are synthesized by using the initial bounding box produced by automatic detection by GMM. In the vicinity of initial bounding box 10 bounding boxes are selected using the scanning window. Next step is production of 20 warped versions by performing geometric transformations like shifting, scaling and rotation in-plane and adding them with Gaussian noise at each pixel. This leads to 200 synthetically produced positive patches.

Negative examples are not generated synthetically, they are collected from the surrounding of initial bounding box. Labelled examples thus produced are use to update the object model.

Stages of P-N learning on each frame:

1) Detector is evaluated on current frame.
2) Errors of detector are evaluated.
3) Updating detector with labelled set of examples.

Thus detector obtained after learning is called final detector.

4.3.2 P-EXPERTS :-

P-expert aims at discovering alternative appearances of object and increases the generalization of object detector. It assumes movement of object along the trajectory and utilize temporal information in video. It recollects the position of object in last frame and roughly calculates the location of object using the tracker. If detector labels the current location to be negative (means has made false negative error), the expert initiates positive example. The TLD system may produce discontinuous trajectory as output is a combination of tracker, detector and integrator, which may not be correct always.

Identification of reliable parts in the trajectory is main challenge of P-expert and it uses the reliable parts to generate positive training examples. Object model is used to identify these parts.

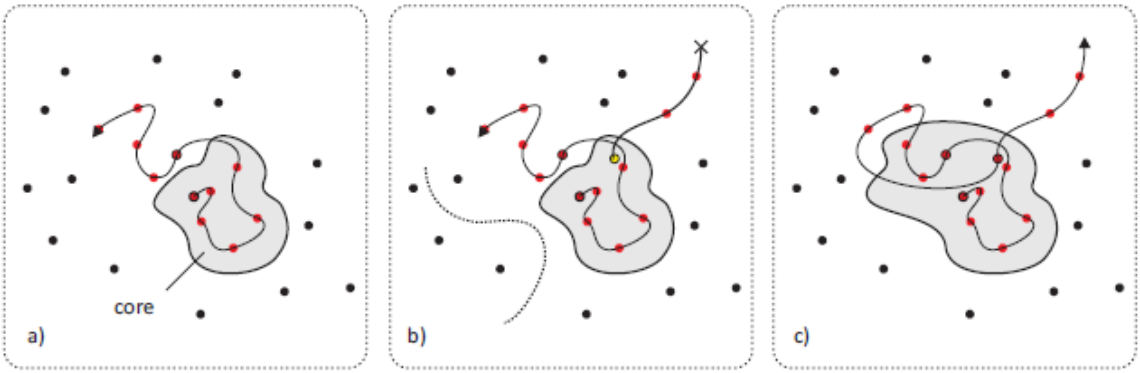


Figure-4.5 P-Experts working to get reliable trajectory

Taking an example where the object model is entitled as coloured points in the feature space. Positive examples are spoken to by red specks associated by a coordinated bend proposing their arrangement, negative examples are dark. Utilizing the similarity S^c greater than a particular threshold, one can characterize a subspace in the feature space. We allude to this subspace to be core of object model. The core is not fixed its size grows as new examples come.

The P-expert works to search reliable parts.

The trajectory is said to be reliable if it enters core and will remain reliable until the time re-initialization or the tracker is failed. Figure b) shows both reliable and unreliable trajectories in the space. And c) shows change in core size after the acceptance of new patches which are positive that are obtained from the reliable track. Reliability is checked for every location, for reliable ones positive examples are created that updates the object model and hence the classifier in the detector stage also gets updated. For every bounding box 10 boxes are chosen on the scanning grid and after shifts, translations and scales 100 synthetic positive examples are created for the ensemble classifier.

4.3.2 N-EXPERTS:-

Negative training examples are generated by this expert. It aims to spot clutter in background and against this clutter the detector should discriminate. Its working is based on a simple assumption that appearance of any object can be at a single location in frame and it utilizes spatial information in video. In the current frame N-expert analyses the responses produced by detector and selects the patch which is most confident. Non Overlapping patches with the most confident one are labelled to be negative detections as they contain background (which is surrounding of the location that is labelled). Most confident patch reinitialize the trackers location. When the patches are far away from the current bounding box (ie, overlap <0.2) the patches are labelled as negative.

CHAPTER 5

IMPLEMENTATION OF PROPOSED WORK

This module explains the implementation of automatic detection of object and TLD building blocks. figure

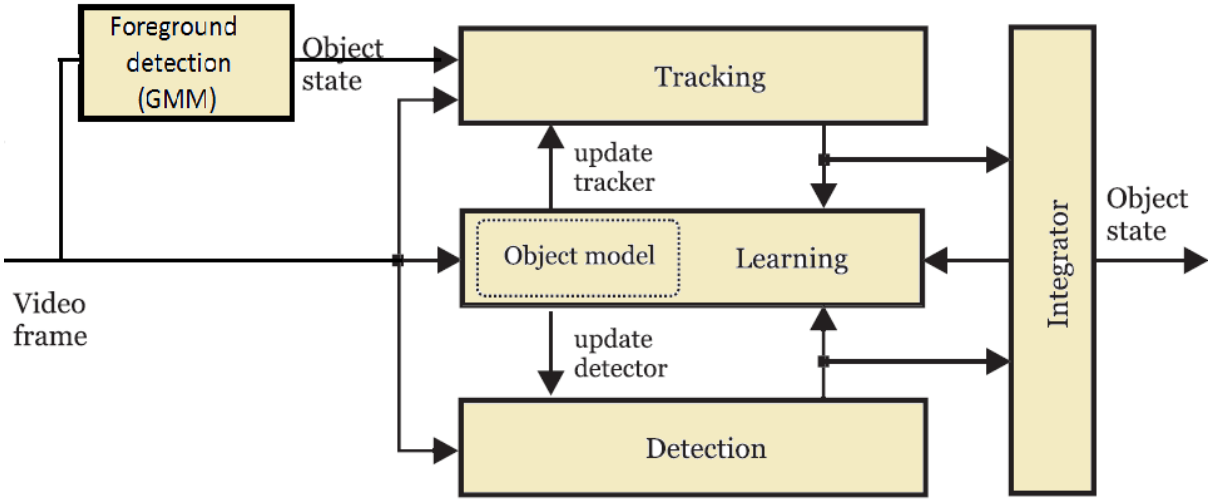


Figure-5.1 The block diagram of proposed work.

5.1 PREREQUISITE KNOWLEDGE

At any point of time, object is represented by its position and area known as the state of object, it could be a bounding box (when our object is present in frame) or a flag (indicating the absence of object) in our case. Aspect ratio of the bounding box is fixed for the object and it is parameterized by its location and scale, rotation is not considered. To measure the spatial similarity between two patches of the video ratio of intersection and union is measured, this is called overlap.

Object's appearance at any instance is represented by image patch *p*. Correlation coefficients are to measure similarity between two patches p_i ,p_j .In the applications of image-processing where the brightness of image patches can vary due to lighting conditions, image is first normalized which is done by subtracting the mean and by dividing it by standard deviation. The correlation of the patches is

$$\frac{1}{n}\sum_{x,y}\frac{1}{\sigma_p^i\sigma_p^j}(p^i(x,y)-\overline{p^i})(p^j(x,y)-\overline{p^j})$$

Where σ_p^i , σ_p^j are standard deviations of patches p^i *and* p^j.

$\overline{p^i}$, $\overline{p^j}$ are means of patches p^i *and* p^j.

Similarity is measured by $S(p^i,p^j) = 0.5(\,\mathrm{NCC}(p^i,p^j) + 1))$

Where NCC is Normalized Correlation Coefficient.

5.2 FOREGROUND DETECTION USING GMM

Background is modelled with the help of mixture of distinct Gaussians that correspond to various background objects. For the detection of foreground every pixel in the image is compared with each Gaussian and classification is done according to the respective Gaussian. In Gaussian mixture model, for learning gradual changes with time, in each and every pixel of the image is taken as a Gaussian distribution.

In GMM, the pixels of the current frame are checked against background model by making its comparison with every Gaussian used in the model unless and until a matching Gaussian is found. In case a match is found, the mean value and variance value of the matched Gaussian will be updated, else a new Gaussian with the mean value equal to the current pixel colour and some initial variance will be introduced into mixture model. Each pixel is then classified depending on whether the matched distribution is representing the background. Background subtraction improves the tracking abilities and makes the system more accurate and error free.

5.3 OBJECT MODEL

Object and its surrounding are represented as a data structure known Object model M.

M = {p_1^+, p_2^+,…,p_m^+, p_1^-,p_2^-,…, p_n^-} where p^+ represents the object patch and p^- represent the background patches. All the patches are ordered according to time when the patch was added to the set of collections, the last patch added to the collection is p_m^+.

In order to estimate the similarity or resemblance of any arbitrary patch *p* and object model M, we define following measures:

- Resemblance of patch with positive nearest neighbor, $S^+(p,M) = \max_{p_i^+ \in M} S(p,p_i^+)$.
- Resemblance of patch with negative nearest neighbor, $S^-(p,M) = \max_{p_i^- \in M} S(p,p_i^-)$.
- Relative similarity . $S^r = \frac{S+}{S+ + S-}$. Its range is form 0 to 1, greater values of the relative similarity show that the patch in more confident and shows the presence of object.

Similarity measure S^r depicts how much a random patch resembles with the appearance model of the object. Nearest neighbor classifier is defined by the relative similarity value.

If $S^r(p, M) > \Theta_{NN}$ the patch is classified to be positive , otherwise it is classified to be negative , where Θ_{NN} is the tuning parameter of NN classifier.

Object model is updated with the new labelled patch if the labelled patch from the NN classifier is different from the P-N experts label which leads to coarser decision boundaries.

5.4 OBJECT DETECTOR

The approach based on sliding-window that scans the input image by a window of various sizes and for every window decides whether the underlying patch contains the object of interest or not. Scanning grid is created at possible scales and shifts with initial bounding box. For example, in a QVGA frame, there are around 50k patches that are scanned in each frame.

Since the possible patches to be classified are too large and we need to design an efficient algorithm hence a structure of three stages is designed that concatenates other classifiers. Stages of object detector are:

- Patch variance filter
- Ensemble classifier
- Nearest-neighbour classifier

Here each stage rejects patches and allows selected patches to pass to next stage, the final template allowed give reliable results for detection. Block diagram of 3 stage object detector is shown in figure.

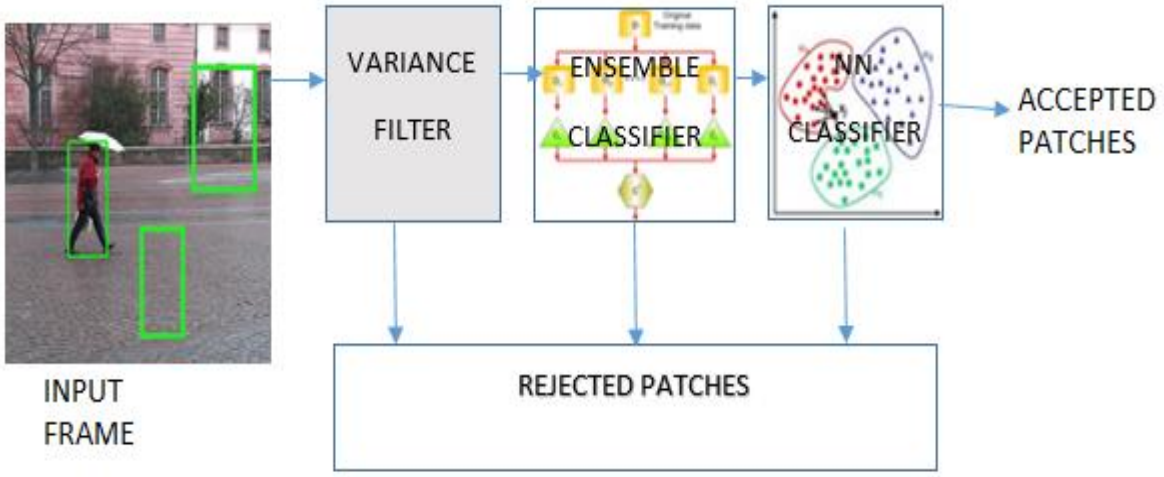


Figure-5.2 Cascaded classifier for object detection

5.3.1 PATCH VARIANCE FILTER:

This filters out the image patches whose gray value variance is less than 50% of the variance of the initial patch selected for tracking. Variance measures the uniformity in the image patch, it rejects the background regions that are uniform (example sky, street, walls) , hence helps in the elimination of more than 50% non-object patches . Variance calculation for patch *p* is done by formula as follows:

$$D(p) = E(p^2) - E^2(p)$$

Where, $D(p)$ - variance of gray image patch

$E(p)$ - mean value of gray image patch

$E(p^2)$ - is the mean value of gray value in the square of image box.

In the process of filtering the threshold chosen by the variance classifier is:

$$var \geq 0.5 * D_1$$

here, *var* – variance of candidate box.

The methods results in faster rejection of non-target areas and makes the computation time faster.

5.3.2 ENSEMBLE CLASSIFIER

When input is not rejected by the patch variance filter it is input to ensemble classifier.

Ensemble classifier is composed of n *base classifiers* where the work of each classifier is to do pixel by pixel comparisons that results in binary code (0 or 1)which is index to posteriors. $P_i(y|x)$, y belongs to {0,1}. These posteriors calculated by each base classifier are averaged and if average value of the posterior is >50%, the patch is classified to be object.

Procedure to do pixel comparison:

1. *Convolve image with Gaussian kernel*: This removes noise from image and smoothens it.
2. *Generation of pixel comparisons*: Independent base classifiers do pixel comparisons. For this first discretization of pixel location in a normalized patch is done and generation of all possible comparisons in vertical as well as horizontal direction. Next step is permutation of comparisons and splitting them into base classifiers that returns 0 or 1 and these results are joined to get binary code. The binary code indexes
3. *Classification* : These posteriors calculated by each base classifier are averaged and if average value of the posterior is >50%, the patch is classified to be object.

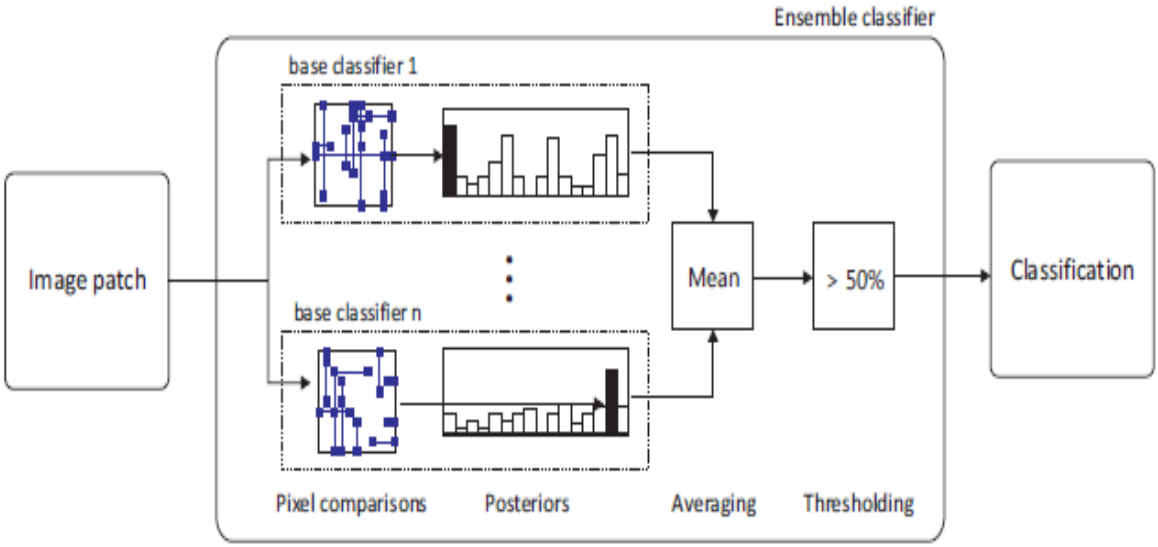


Figure -5.3 Ensemble classifier

5.3.3 NEAREST NEIGHBOUR CLASSIFIER

Following the variance filter and ensemble classifier we are left with few bounding boxes that are still unclassified, almost fifty boxes. Online model is used for classification of these left patches with the help of NN classifier. The patch is classified to be object if S^r relative similarity is greater than a particular threshold of NN classifier Θ_{NN} =0.6 . Colour histograms were used to extend the confidence measure of NN Classifier.

4.4 TRACKER

Estimation of the movement of any object between the consecutive frames is done by the tracking module.

Optical Flow determines motion vectors in every frame of video. By thresholding these motion vectors, binary feature image is created which has blobs of the moving object. This is effected by noise, in order to remove scattered noise median filtering is done and close operation is done to remove the smaller blobs. This median-shift tracker is robust to occlusion, rotation and scale. It gives the estimate speed of object, translation and scale.

This method is based on gradients. Optical flow method recovers the motion in image at every pixel which is translated from one point to another in next frame.

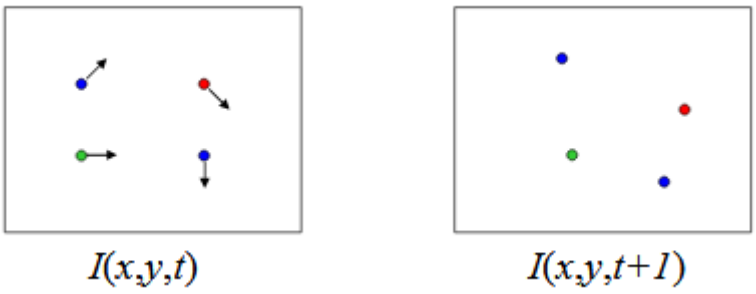


Figure –5.4 Brightness constancy in LK method

Lucas-Kanade assumes following constraints:

- Brightness consistency: image brightness in any small region will remain same only its location changes.

- Spatial coherence: in the neighbourhood of each pixel have similar motion as they belong to same surface.
- Temporal persistence: There is gradual motion of image patch over time.

Constraint equation of the optical flow method is:

$$I_x u + I_y v + I_t = 0$$

Here

I_x , I_y , I_t – spatiotemporal derivatives of brightness in image

u and v are horizontal and vertical optical flow respectively.

The image is divided into smaller parts and each part is assumed to move with constant velocity.

Least-square fit is performed on constraint equation. Fitness is achieved by minimization of following equation:

$$\sum_{x \in \Omega} W^2 [I_x u + I_y v + I_t]^2$$

W gives emphasis on the constraints at the centre of every section. The solution to minimization is:

$$\begin{bmatrix} \sum W^2 I_x^2 & \sum W^2 I_x I_y \\ \sum W^2 I_y I_x & \sum W^2 I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum W^2 I_x I_t \\ \sum W^2 I_y I_t \end{bmatrix}$$

The LK tracker computes I_t with help of difference filter [-1, 1].

And u and v values are calculates as:

Compute I_x and I_y values by the kernel [-1 8 0 -8 1] /12and its transpose.

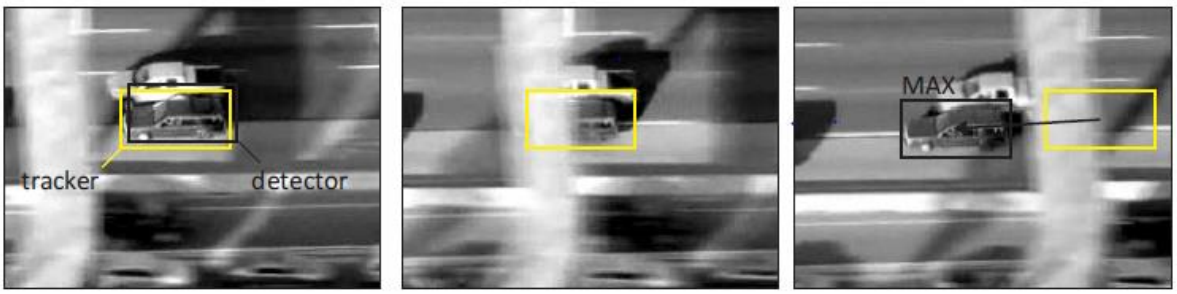
Compute I_t in between two images using [-1, 1] kernel.

Smoothening of the gradient components I_x, I_y, and I_t, by using separable and an isotropic kernel with effective 1-D coefficients as [1 4 6 4 1]/16.

Solve 2 by 2 linear equation. The solution can be non-singular, singular or zero depending on eigen values. The eigen values of the matrix is compared with a threshold for noise and u and v values are calculated.

5.5 INTEGRATOR

The work of integrator is to join the results of tracker and the detector and produce a single output. In case both the tracker and detector produce output which is a bounding box the object is said to be invisible. Detector part do the localization of known templates, tracker does the localization on new templates discovered and thus new data is brought to the detector. The output of integrator is maximal confident box , which is measured by the use of resemblance measure known as S^c(conservative similarity).



(A) (B) (c)

Figure-5.5 Integrator a) successful tracking and detection b)tracker works but detector fails due to occlusion c) the tracker fails as there is sudden change in trajectory but the detector redetects the object when it comes in field of view. The patches are then assigned conservative similarity and tracker is re-initialized by the detector.

CHAPTER 6

EXPERIMENTS AND RESULTS

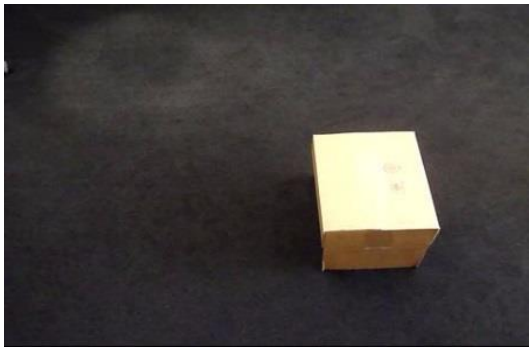
In this section the results of the work done in the thesis are explained. Benchmark results are obtained using automatic detection of object and long-term tracking using TLD framework.

Coding environment - Open Source C++ implementation
Operating system – Ubuntu 64 bit
This project required installation of OpenCV .

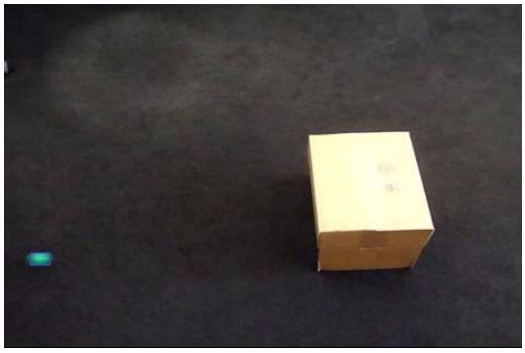
6.1 SCENARIO 1:

Description of test video used:
"singleball.avi" is standard test video from Matlab vision toolbox. This video is taken from stationary camera i.e. Background does not change .Initially no object is present in the camera view. Ball of green colour comes into camera field of view from left side. The ball undergoes occlusion in between due to object placed on the floor, later it reappears after in some time. The ball is moving at a good speed.
The dimensions of video- 480 x 36, frame rate- 30 frames per second, codec- Uncompressed 8-bit RGB

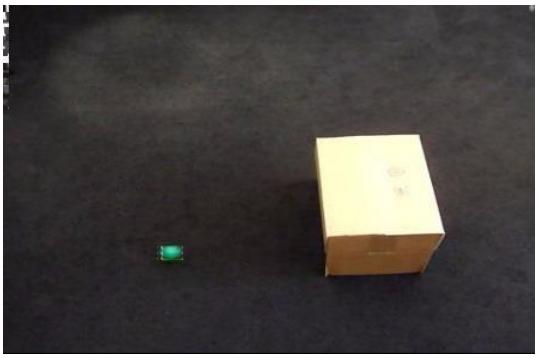
Results:
The proposed method successfully detected the ball in the video when it came into field of view. Result shows that the algorithm successfully works for the rigid objects whose shape does not change with time.
The detection result GMM using is shown in figure. The blue box detection is result of detection. After this TLD framework was used to do further processing. Positive and negative patches were continuously produced and after the occlusion because of the learning component the detector and tracker could track the ball. The red box shows the object.



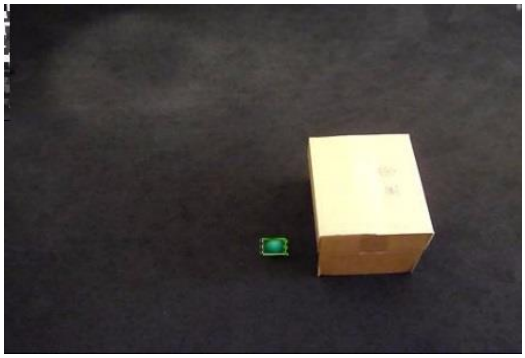
Frame 7



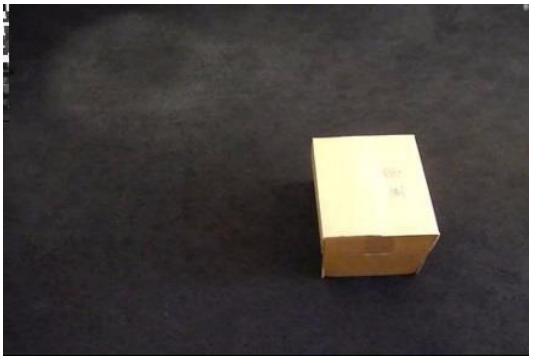
Frame 14



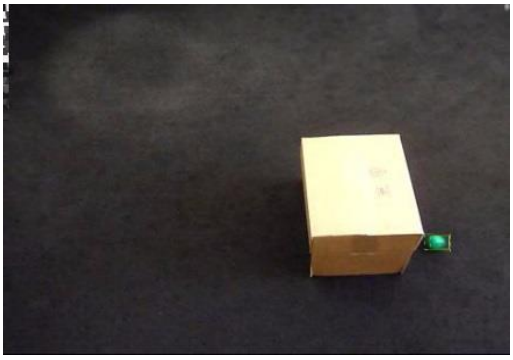
Frame 19



Frame 24



Frame 28



Frame 35

Figure-6.1 Results of tracking in scenario 1.

In the output we can see patches are generated on the left side of window are negative examples and those on right side are positive examples .Colour histograms are used to match and find maximally confident patch.

Green bounding boxes are the result of detector process after cascaded classifier with fern filter. Blue box is mean of the boxes known as cluster means. The red box represent status is ok, shows the surety of object. Yellow box means that status is unsure.

6.2 SCENARIO 2:

Description of test video used: To test the performance of the code. I myself made a video in the DTU college campus. This video is taken from stationary camera (DSLR) i.e. Background does not change much but the video as natural and obvious environmental changes which result in change in illumination. Initially no object is present in the camera view. Initially a person1 comes in camera field of view and walks in the camera field from right hand side. There are various views of person1 due to rotation and drift. Later in video person2 walks and comes in field of view. There is a tree in between which causes full as well as partial occlusion of person1. The person1 reappears and walks.

Length -00:00:32

Frame width, height -1920 X 1080

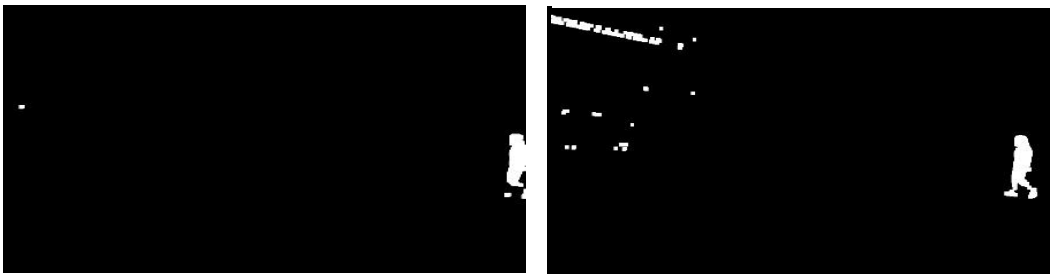
Data rate -25893 kbps, Bit rate – 26149 kbps

Frame rate -50 frames/second

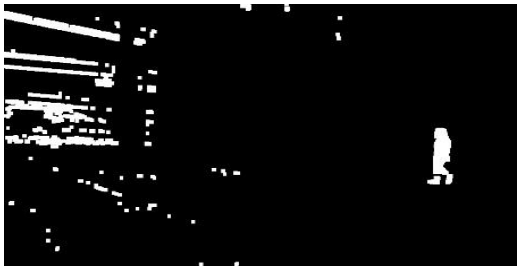
Results:

The proposed method successfully automatically detected the pedestrian coming in video sequence. The algorithm is suitable for tracking pedestrian which is a non-rigid objects whose shape changes as the object proceeds. The detection result GMM using is shown in figure. The blue box detection is result of detection. After this TLD framework was used to do further processing. Positive and negative patches were continuously produced, positive patches of the person can be seen in various views and angles, also negative patches are generated that have background. The red box shows the object.

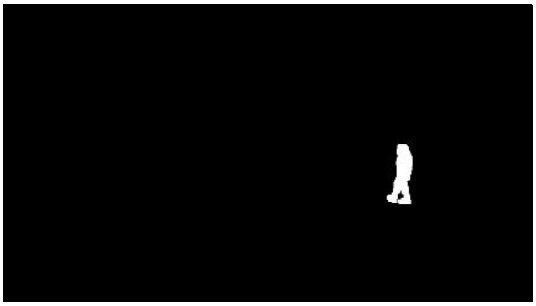
1) Result of GMM to detect foreground, successfully the foreground is extracted. We can see due to change in illumination small patches are also part of foreground.



FRAME 37 FRAME 63

FRAME 110 FRAME 125

FRAME 150 FRAME 197

Figure-6.2 Results of GMM in scenario 2

The background subtraction method can be implemented only in the cases when the video is taken from stationary cameras. Hence generally on platforms, shops, traffic surveillance systems the cameras are stationary and the proposed algorithm will give successful automatic long term tracking results and can track the object when it reappears even after a long time.

2) Outcome of algorithm. We can see that in frame 15 no object is present , person1 is automatically detected in frame 30.


Frame 15 Frame 30


Frame 37 Frame 100


Frame500 Frame 680


Frame 700 Frame 760

Figure- 6.3 Results of tracking in scenario 2

Description of result :

In the output we can see patches are generated on the left side of window are negative examples and those on right side are positive examples .Colour histograms are used to match and find maximally confident patch.

Green bounding boxes are the result of detector process after cascaded classifier with fern filter. Blue box is mean of the boxes known as cluster means. The red box represent status is ok, shows the surety of object. Yellow box means that status is unsure.

6.3 SCENARIO 3:

The test is done on a video from caviar dataset “OneShopOneWait1cor”. Video is taken in shopping centre in Lisbon. The videos are time synchronised. The resolution of video is half-resolution PAL standard - (384 x 288 pixels, 25 frames per second) and is compressed using MPEG2. The MPEG file sizes are mostly between 6 and 12 MB, a few up to 21 MB.

A lady enters far away in the corridor and walks towards the surveillance camera installed. Initial size of the lady is small as it is far but later as she proceeds the size increases also she rotates at several instances hence object is at various orientations.

Results:

1) Automatic detection result through GMM :

Figure-6.4 Automated detection in scenario 3

The lady is successfully detected from top to bottom using proposed method. This is a labelled object that was used to supervise the learning process.

2) The positive and negative patches are continuously add to right and left sides of the running video. This is frame number 570.

Figure- 6.5 Frame with positive and negative patches in right and left side.

3) Positive patches generated by P-expert

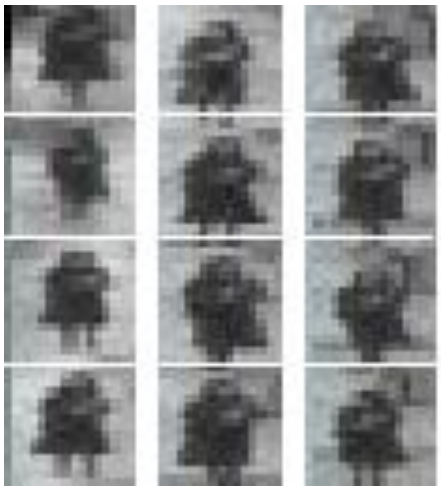


Figure- 6.6 Positive patches in scenario 3

Negative examples generated by N-expert

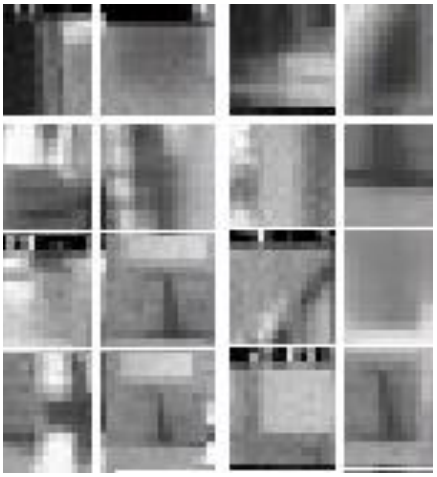


Figure- 6.7 Negative patches in scenario 3

4) Tracking results shows the object being tracked.

Frame 28

Frame43

Frame 300

Frame 500

Frame 600

Frame 660

Figure- 6.8 Results of proposed tracking in scenario 3

The proposed method did successful automatic detection of the lady as she comes in camera field of view using GMM model. This method does not require offline training of object of interest and hence can be applied to detection of any type of object.

After the automatic detection of lady the online learning of object was successful hence at different scales and orientations the lady was tracked. We can see the algorithm generated good number of positive and negative patches.

The P-expert increased the generalizabitlity of the object and the N-expert increase easy discrimination of object of interest with the background with help of negative patches.

In the result we can see different frames with different object views, appearances and scales.

6.4 SCENARIO 4:

The test is done on a video from caviar dataset “OneLeaveShopReenter1cor.mpg”. Video is taken in shopping centre in Lisbon. The videos are time synchronised. The resolution of video is half-resolution PAL standard - (384 x 288 pixels, 25 frames per second) and is compressed using MPEG2. The MPEG file sizes are mostly between 6 and 12 MB, a few up to 21 MB.

Person1 comes out of the store and later in the video he re-enters. The view is of corridor, the person2 comes out of the store and walks in the corridor. While person1 re-enters the store he causes occlusion of person2 and person2 continues to walk in corridor towards the surveillance camera installed.

Results:1)Automatic detection result through GMM :The lady is successfully detected from top to bottom using proposed method. This is a labelled object that was used to supervise the learning process.

Figure-6.9 Automated detection in scenario 4

2) The positive and negative patches are continuously add to right and left sides of the running video. This is frame number 289.

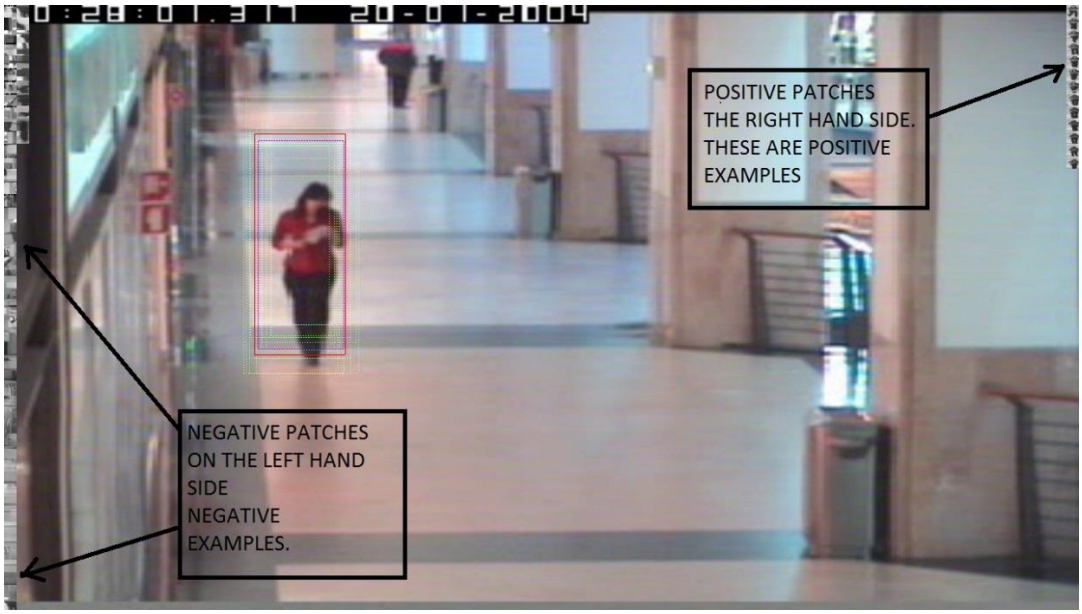


Figure- 6.10 Frame showing positive and negative patches in right and left side.

3)Positive patches generated:-

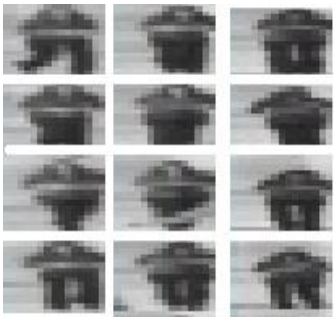


Figure-6.11 Positive patches in scenario 4

Negative patches generated:-

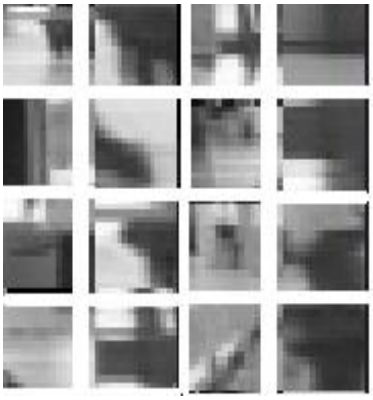


Figure- 6.12 Negative patches in scenario 4

4)Tracking results shows the object being tracked.

Frame 170 Frame 187

Frame 215

Frame 223

Frame223

Frame 227

Frame 237

Frame 254

Figure- 6.13 Results of proposed tracking in scenario 4

The proposed method did successful automatic detection of person2 as she comes in camera field of view using GMM model.

After the automatic detection of person2 the online learning of object was successful hence at different scales and orientations person2 was tracked. In the result frames we can see different frames with different object views, appearances and scales. Even after occlusion of person2 by person1 the framework successfully did long-term tracking when person1 reappeared.

CONCLUSION

In this paper we proposed a new method known as automatic detection using Gaussian Mixture Model that comes under background modelling which was merged with TLD framework for long-term object tracking. The system is suitable for use in automatic surveillance and monitoring. We have tested our proposed method in several videos having stationary background. Each of them had different challenges faced by the tracker like partial occlusion , long-term occlusion , change in orientations , out of plane rotation ,disappearance of object in some frames (when any object goes out of the frame for some time and comes later). Under all such conditions the proposed method outperformed other previously made trackers. This method does not require offline training of object of interest and hence can be applied to detection of any type of object. Online learning using P-N learning successfully trained the classifier with good number of online produced training data.

This work can be extended to multiple automatic object tracking where more than two objects would be tracked efficiently. We can also work for the reduction in the time cost for each frame processing .We can impose learning to the tracker part, so that if detector part fails, it would act as the backup.

REFERENCES

1. Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. "Tracking-Learning-Detection", IEEE Transactions on Pattern Analysis andMachine Intelligence, 2012.

2. H. Wang, Q. Xiao, Q. Ye and X. Wang, "Cross Camera Object Tracking in High Resolution Video Based on TLD Framework," Multimedia Big Data (BigMM), 2015 IEEE International Conference on, Beijing, 2015, pp. 264-267.

3. Jianbo Shi and C. Tomasi, "Good features to track," Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEE
E Computer Society Conference on, Seattle, WA, 1994, pp.593-600.

4. S. Avidan, "Ensemble Tracking," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 2, pp. 261-271, Feb. 2007.

5. Z. Kalal, J. Matas and K. Mikolajczyk, "Online learning of robust object detectors during unstable tracking," Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, Kyoto, 2009, pp. 1417-1424.

6. R. T. Collins, Yanxi Liu and M. Leordeanu, "Online selection of discriminative tracking features," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 10, pp.1631-1643,Oct.2005.

7. B. Babenko, M. H. Yang and S. Belongie, "Visual tracking with online Multiple Instance Learning," Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, Miami, FL, 2009, pp. 983-990.

8. Z. Kalal, K. Mikolajczyk and J. Matas, "Tracking-Learning-Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pp. 1409-1422, July 2012.
40

9. Y. Wu, J. Lim and M. H. Yang, "Online Object Tracking: A Benchmark," Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, Portland, OR, 2013, pp. 2411-2418.

10. E. Park, H. Ju, Y. M. Jeong and S. Y. Min, "Tracking-Learning-Detection Adopted Unsupervised Learning Algorithm," Knowledge and Systems Engineering (KSE), 2015 Seventh International Conference on, Ho Chi Minh City, 2015, pp. 234-237.

11. Q. Yu, T. B. Dinh, and G. Medioni,"Online Tracking and Reacquisition Using Co-trained Generative and Discriminative Trackers" ,2008, pages 678–691, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

12. D. Jepson, D. J. Fleet and T. F. El-Maraghi, "Robust online appearance models for visual tracking," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 10, pp. 1296-1311, Oct. 2003.

13. W. Hailong, W. Guangyu and L. Jianxun, "An improved tracking-learning-detection method," Control Conference (CCC), 2015. 34th Chinese, Hangzhou, 2015, pp. 3858-3863.

14. Z. Kalal, K. Mikolajczyk and J. Matas, "Forward-Backward Error: Automatic Detection of Tracking Failures," Pattern Recognition (ICPR), 2010 20th International Conference on, Istanbul, 2010, pp. 2756-2759.

15. Michael D. Breitenstein. "Robust tracking-bydetection using a detector confidence particle filter", 2009 IEEE 12th International Conference on Computer Vision, 09/2009.

16. S. Chandana, "Real time video surveillance system using motion detection," in *2011 Annual IEEE India Conference*, 2011, pp. 1-6.

17. J. W. Seo and S. D. Kim, "Recursive On-Line and Its Application to Long-Term Background Subtraction," *IEEE Transactions on Multimedia,* vol. 16, pp. 2333-2344, 2014.

18. Kalal Z, Mikolajczyk K, Matas J. Tracking-Learning-Detection [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1409-1422.

19. Cheng Liying, Zhang Dan, Zhao Shuying, Xue Dingyu. An improved visual tracking algorithm based on TLD [J]. Science Technology andEngineering, 2013, 13(9): 2382-2386.

20. Zhou Xin, Qian Qiumeng, Ye Yongqiang, Wang Congqing. Improved TLD visual target tracking algorithm [J]. Journal of Image and Graphics, 2013, 18(9): 1115-1123.

21. Jin Long, Sun Han. An improved TLD visual target tracking method [J]. Computer and Modernization, 2015, 236(4): 42-46.

22. Xiao Guoqing, Ye Qingwei, Zhou Yu, Wang Xiaodong. Long-term video tracking algorithm of optimized TLD based on Mean-Shift [J]. Application Research of Computers, 2015, 32(3): 925-928.

23. Lucas B D, Kanade T. An interative image registration technique with an application to stereo vision[C]// Proceedings of the 7th International Joint Conference on Artificial Intelligence. Menlo Park, California: AAAI Press, 1981: 674-679.

24. Kalal Z, Matas J, Mikolajczyk K. P-N Learning: Bootstrapping binary classifiers by structural constraints [C]// Proceedings of IEEE
Conference on Computer Vision and Pattern Recognition. New York: IEEE Press, 2010: 49-56.2055