Subspace Based Adaptation of Detectors for Video

A Thesis submitted in the partial fulfillment for the award of

Degree of Master of Technology

in

Software Engineering

by

Sangeeta Bhan

(2K15/SWE/15)

Under the Guidance of

Dr. Kapil Sharma

DEPARTMENT OF COMPUTER ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

Bawana Road, Delhi

2017

CERTIFICATE

DEPARTMENT OF COMPUTER ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

Bawana Road, Delhi

It is certified that the work contained in this thesis entitled “Subspace Based Adaptation of Detectors for Video", by Sangeeta Bhan (Roll No. 2K15/SWE/15), has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr. Kapil Sharma

Head of Department

Department of Information Technology

Delhi Technological University

Bawana Road, Delhi

Declaration

I hereby declare that the thesis entitled, "**Subspace based Adaptation of Detectors for Video**", is a bona fide work done by me in partial fulfillment of requirements for the award of Master of Technology Degree in software technology at Delhi Technological University (New Delhi) is an authentic work carried out by me. The matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma to the best of my knowledge.

Sangeeta Bhan

Department of Software Engineering

Delhi Technological University,

Bawana Road, Delhi.

Acknowledgment

First and foremost, I would like to express my sincere gratitude towards my thesis supervisor, Dr. Kapil Sharma, for his constant support and encouragement. I am indebted to him for his constant motivation and patient guidance. I would like to thank the faculty and staff of the Department of Computer Science and Engineering, DTU Delhi, for providing the necessary infrastructure and testing facilities and environment which allowed us to work without any obstructions. I would also like to thank to the Almighty God with his blessings I had an opportunity and strength to do this wonderful project and studies, as well as to my parents who always support me and guide me in the right direct direction with their incredible experiences of life.

Sangeeta Bhan
2K15/SWE/15

Abstract

Object detection in videos has always been a challenging problem to work with. Detection of a particular class object plays an important role in many real-world applications. Since the domain of source and target video vary significantly, classifier being trained on source video does not give expected results on the target video. Thus, domain adaptation techniques are used, one of which is Subspace Based Adaptation. In this technique, first, we compute both source and target subspace from the features collected. Since we do not have target data directly, we use different ways to get data from the target video. Compute subspace after collecting the data from both source and target videos. This generated source and target subspaces are described by eigenvectors. These d-dimensional subspaces are independently created by PCA for both source and target video. With the help of these subspaces, a transformation function is generated. Using this function source coordinate system is transformed into the target aligned source coordinate system. Now using this new coordinate system we map the source data to target aligned source subspace. The other thing we use is, learning from online samples. Sliding Window Method is used to categorize online data into TP and FP, this weakly labeled data is used to modify the model iteratively. We run our method of domain adaptation in different ways out of which some perform fairly well.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

In this thesis, we present a novel and efficient method of object detection for a particular class in videos. It operates on both color and grayscale video. Both source and target videos should be captured from stationary cameras.

In detection the object of the particular class, it is typically presumed that source and target data have the same distribution. However, it is not true in real world applications. We are training our simple detector model from the source video with the help of its annotation file. Initially, the problem is to collect the negatively labeled data from video. We have used Hard *Negative Mining* to collect this negatively labeled data. Next, we extract the features of these data using *Histogram of Oriented Gradients*. With the help of *Background Subtraction using MoG*, data samples were collected from the target video for subspace calculation. PCA is used to find the best eigenvectors for both source and target dataset. These eigenvectors represents our subspaces. Once we have calculated the subspaces using PCA, transforms the source subspace coordinate system into the target aligned source subspace coordinate system by aligning the source basis vectors with the target ones. Generate training data by mapping source data into this target aligned source subspace. Now, we train our linear SVM model using this new labeled data. Iteratively use online samples or bounding boxes detected on target video to generate new detector model. We also tested other methods where redundancy in training samples is reduced using tracking method. For that, we collected every object of a particular class with a difference of K frames. But results were not as expected

due to occlusion in the video. We used adaptation technique based on subspace alignment to adapt the feature subspaces between source and target subspace for localized object detection. Our main aim is learning iteratively from target video to train the model with good subspace by using online samples.

1.2 Motivation

Video surveillance cameras are installed and used everywhere around us and have become part of our everyday life. Lots of data is generated by this in today's life and it is very difficult to collect the useful data from this. Many security tasks such as analyzing and monitoring traffic or banks airports, security of objects like important buildings. Crowd surveillance cameras installed in public areas such as shopping malls and public transports. We want smart video data mining like measuring the speed of vehicles or counting people in some crowded areas. These are some of the common applications of stationary video cameras today. Widely used method for such applications is *Automated Object Detection*. An automated object detection and tracking was developed in order to build a reliable visual surveillance system. Object detection is performed by means of a background subtraction technique. It has become an important research area in computer vision field. It is still an open and challenging problem in this area due to image and video complexities. Many advances have also been applied by the computer vision community with promising results in object detection field, detector models are still being trained and tested on images consisting of only one object zoomed and cropped at the center of a relatively uniform background. As a result, in such experimental settings, a problem of object detection is reduced to that of image classification. While domain adaptation is a challenging problem for image classification, it becomes even more challenging for object detection when target domain labels are unavailable and the majority of the frame in the video is occupied by the background class (random sampling of the window will not be sufficient for effective domain adaptation in videos). A lot of work has been done in this field with several different approaches.

1.3 Challenges in Object Detection

Following are some of the challenges incurred while detecting objects of a particular class in a video or image:

- *Illumination variations*: In real time videos illumination condition varies a lot, ranging from direct sunlight and shadows during the day to artificial or dim lighting on any object(or inside any building). Although some solutions for illumination variation have been provided, they are still ineffective compared to the human visual system.

- *Object deformation or Object pose*: Object changes from what it looks like and therefore template approaches for object recognition tend to suffer low performance in high changing objects.

- *Occlusion:* As some parts can't be seen, it makes hard to recognize shapes and hence objects.

- *Intra-class variation*: In-class variation makes di cult to group dissimilar objects (i.e. car types, colors, shapes widely vary but they all represent cars).

- *Viewpoint*: Aerial point of view, frontal viewpoint, makes an appearance to vary widely.

- *Miscellaneous backgrounds*: This is the main problem of object detection in the videos. Since, training, and testing video have different domains, the false positive rate may increase, if we do not learn from the test video.

- Cluttered scenes: Sometimes scale change or in-plane rotation of object in the frames may not give the expected results.

In this thesis, we mainly focus on background variation for source and target videos.

1.4 Organization of Thesis

The remaining part of this thesis is organized as follows. Chapter 2 presents work related to our thesis and a brief knowledge of feature extraction, support

vector machine, and PCA. Our novel algorithm is explained in chapter 3. It contains all about hard negative mining, background subtraction using MoG, sliding window method which we used to categorize true positive and false positive and subspace-based adaptation with online samples. Chapter 4 contains information related to datasets, experiment setup and results with different methods. Finally, Chapter 5 concludes the whole work we have done.

Chapter 2

Related Work

2.1 Related Work in Adaptation

Over the past several years, many different approaches [14,12,10,6] have been proposed to detect objects of a particular class in videos and images. The main problem in object detection is varying domains of source and target data. So the main issue is to find out the relationship between these two domains. Domain adaptation is a widely used technique in computer vision [18] and language processing [9]. In this thesis, we work on object detection methods, with as well as without the use of domain adaption. Our main focus is on the adaptation technique with learning from online samples, as it does not require any labeled data from the target domain. A common approach is to assume the existence of a domain invariant feature space and the objective of domain adaptation is to approximate this space.

A classical strategy related to our work consists of learning a new domain-invariant feature representation by a new projection space. PCA based DA methods have then been naturally investigated [6,12,13] in order to find a common latent space where the difference between the marginal distributions of the two domains is minimized with respect to the Maximum Mean Discrepancy (MMD) divergence. *J. Blitzer and R. McDonald* [9] says that discriminative learning methods work best when their training and test data are drawn from the same distribution. They adapt existing models from a resource rich source domain to a resource poor target domain and introduce structural correspondence learning to automatically induce correspondences among features from different domains. Frameworks like discourse recognizers and

programmed interpreters utilize progressively complex discriminative models, which sum up well to new information that is drawn from an indistinguishable conveyance from the preparation information. Be that as it may, as a rule we may have a source area with ample marked preparing information, however we have to process material from an objective space with an alternate circulation from the source space and no named information. In such cases, we should find a way to adjust a model prepared on the source area for use in the objective space. This work concentrates on utilizing unlabeled information from both the source and target spaces to take in a typical element portrayal that is significant crosswise over the two areas. We theorize that a discriminative model prepared in the source space utilizing this regular element portrayal will sum up better to the objective area. This portrayal is found out utilizing a technique we call auxiliary correspondence learning (SCL). The key thought of SCL is to distinguish correspondences among highlights from various areas by displaying their relationships with turn highlights. Rotate highlights are highlights which carry on similarly for discriminative learning in the two spaces. Non-turn highlights from various areas which are related with a large number of a similar rotate highlights are expected to compare, and we treat them also in a discriminative student. SCL is a general method, which one can apply to highlight based classifiers for any assignment. An imperative yet once in a while investigated setting in space adjustment is the point at which we have no named preparing information for the objective area. We first exhibit that in this circumstance SCL significantly enhances execution over both regulated and semi-administered taggers. For the situation when some in-area marked preparing information is accessible, we demonstrate to utilize SCL together with the classifier blend systems more prominent execution. Other strategies have been explored on image datasets as well, such as using metric learning approaches [3, 11] or canonical correlation analysis method [8] over different views of the data to find a coupled source-target subspace where one assumes the existence of a performing linear classifier on the two domains. The main aim of our thesis is to first generate common subspace for different domains and then use the online samples to make the model more accurate. There are several methods [19, 4] which use manifold alignment to generate the common subspace. The essential thought of customary high dimensional

correspondence learning strategies, we can partition its strategy into two principle parts. The first one is to take in the regular installing of two data sets, with the requirement that low-dimensional portrayals of ever-known balanced relating perceptions are the same. The other stride is to assign correspondence in view of their low dimensional embedding, as the substitution of the comparing relationship in high-dimensional space. Domain adaptation of deformable part-based models (DPMs) [10] for object detection was also an application of this which uses the incremental domain adaptation for object detection assuming weak-labeling. The DPM can likewise represent diverse parts which, for example, can be utilized to all the more precisely display an object under various perspectives. In this way, the capacity to adjust such a rich model between various spaces is fundamental. We can see a DPM as a specific instance of an structural model, where the components and parts define the structure. The DPM is defined by one root filter and a pre-set number of part filters. Part filters work at double the resolution of the root filter. The root goes about as reference and every single other part are associated with this reference (star model). To better catch intra-class varieties, star models can be additionally consolidated into a blend of components. Boosting- based [14] approaches are used for detection of objects in image or video. During object detection process, the two classifiers work together to determine the location of real object. The online dual-boundary Boosting fern classifier was first used to detect object based on sliding window strategy, and the detection regions with fern classifier score located between positive and negative boundaries will be further recognized by the SVM model. Except that, the detection regions, with fern classifier score above positive boundary will be considered as real objects and the rest are backgrounds. First, they train the initial model on some small labeled datasets and then use this model to collect a larger labeled dataset then train the new model iteratively by using this new dataset.

2.2 Related Work in Adaptation for Videos

Complex occasion recovery from databases of recordings is difficult on the grounds that notwithstanding the difficulties in displaying the presence of static visual concepts–e.g., objects, scenes– demonstrating occasions additionally

includes displaying worldly varieties. Notwithstanding the difficulties of speaking to movement components and time, one especially noxious test is that the quantity of potential occasions is substantially more prominent than the quantity of static visual ideas, opening up the notable long-tail issue related with question classes. Recognizing and gathering preparing information for a far reaching set of items is difficult. For complex occasions, notwithstanding, the assignment of specifying a complete arrangement of occasions is overwhelming, and gathering curated preparing video datasets for them is altogether unfeasible. Subsequently, a current pattern in the occasion recovery group is to define an arrangement of more straightforward visual ideas that are viable to model and after that consolidate these ideas to define and identify complex occasions. This is regularly done when no cases of the perplexing occasion of intrigue are accessible for preparing. In this setting, preparing information is as yet required, however just for the more constrained and less complex ideas. There are some methods, which works on videos like self-paced domain adaptation from images to video by shifting weights [12]. They used domain adaptations which iteratively adapt the detector by re-training it with discovered target domain examples by choosing easiest target video first. First, they train their detector on some image dataset and then test it on video datasets. At every iteration, their algorithm adjusts by considering an expanded number of target domain cases, and a diminished number of source domain examples. To find target domain cases from a lot of videos, they used score trajectory tracks instead of bounding boxes. They trained the detector on car, boat, bicycle, dog and Keyboard and get highest average precision for a bicycle which is 20 percent. In our algorithm, we trained detector for a car only from the source video and test it on target video using iterative subspace based adaptation with online samples and we get 56 percent of average precision for that. The method proposed in Domain Adaptive Object Detection [6] in the video using transfer component analysis (TCA) which transform source and target data to latent space that minimizes the distance between their distributions. They used unsupervised and semi-supervised settings to test their method. They get the average precision of 17 percent for a detector with unsupervised adaptation and 38 percent for the detector with semi-supervised adaptation. The domain adaptation has been achieved by using approach

based on subspace alignment that efficiently projects the source subspace to the target subspace. The proposed method used results in improved object detection. We used principal component analysis to create subspace and background subtraction method to collect target data from video. Even if we do not use the online samples to learn detector, we get 42 percent of average precision which is better than what have been achieved through TCA.

Chapter 3

Background Theory

Object detection and classification is the area which has received much attention in the research of computer vision and pattern recognition recently. It is a challenging yet promising task, which has many important applications as we have shown in chapter 1. This section captures some of the background theories like feature extraction, object detection model, and PCA, which are related to our work.

3.1 Feature Detection and Extraction

Every object in an image has some features or key points which have enough information to be used for its detection. In order to make the approach feasible, these features need to be computed easily for a large collection of images and rapid extraction. Image features provide the information about intensity variation and background invariance. Features can be based on blob, intensities, gradients, points, color or their combinations.

A standout amongst the best object detection depends on the learning of a deformable part-based model (DPM) utilizing HOG-style features and a latent SVM learning method [44]. The DPM can likewise represent distinctive parts which, for example, can be utilized to all the more precisely show a object under various perspectives. Along these lines, the capacity to adjust such a rich model between various areas is fundamental. The proposed work in [10] give strategies to performing domain adaptations of DPMs, relating adjustment of the DPM structure. They see a DPM as a specific instance of an structural model, where the segments and parts define the structure. In like manner, and

formulate the learning of a DPM as a general latent basic SVM.

There are various kind of approaches to determine the image features. In our work, we mainly focus on Histogram of Oriented Gradients [5].

Histogram of Oriented Gradients descriptor uses the way that an object's appearance and shape inside an image can be very much portrayed by the conveyance of its intensity gradients as the votes in favor of prevailing edge directions. HOG can be utilized as highlight descriptor where the nearness of gradient orientation in confined parts of an image assumes critical parts. The use of orientation histograms has various precursors [26,22,27], be that as it may it just accomplished improvement when joined with local spatial histogramming and standardization in Lowe's Scale Invariant Feature Transformation (SIFT) approach to manage wide baseline image coordinating [21], in which it gives the central image patch descriptor for coordinating scale invariant key points. SIFT style approaches perform astoundingly well in this application [21,25]. The Shape Context work [1] inspected alternative cell and block shapes, yet at first using simply edge pixel counts without the orientation histogramming that makes the portrayal so fruitful. The accomplishment of these sparse component based representations has decently eclipsed the power and ease of HOG's as dense image descriptors. We assume that our examination will change this. In particular, our easygoing examinations prescribe that even the best current key point based philosophies are likely going to have false positive rates no under 1–2 requests of extent higher than our thick system approach for human disclosure, mainly because none of the key point identifiers that we think about recognize human body structures reliably. The HOG/SIFT depiction has a couple of focal points. It gets edge or slant structure that is particularly typical for adjacent shape, and it does all things considered in an area depiction with a successfully controllable level of invariance to neighborhood geometric and photometric changes: understandings or upsets have little impact if they are considerably humbler that the area spatial or introduction holder estimate. For human area, rather coarse spatial looking at, fine presentation testing and strong adjacent photometric institutionalization winds up being the best methodology, presumably in light of the way that it licenses limbs and body pieces to change appearance and move from side to side a significant sum surrendered that they keep a by and large

upright presentation. HOG varies from scale-invariant feature transformation (SIFT) (include extraction strategy) in a way that the previous works on a thick network of consistently separated cells and utilizations neighborhood differentiate standardization on covering hinders for enhanced exactness.

HOG feature descriptor can be gotten by first partitioning the picture into little bordering areas of equivalent size, called cells, at that point gathering a histogram of gradient directions for the pixels inside every cell, and finally joining every one of these histograms. A gathering of cells is known as a block. Following are the means to object using HOG:

- **Gradient Computation**

Detector performance is delicate to the path in which gradients are processed, however the least complex plan ends up being the best. We tried gradients computing utilizing Gaussian smoothing took after by one of a few discrete derivatives. Simple 1-D [−1,0,1] masks at σ=0 work best. Utilizing bigger masks dependably appears to diminish execution and smoothing harms it significantly. For extraction of HOG feature, image gradients are computed after converting the image into gray color. The most widely recognized technique is to apply the 1-D focused, point discrete derivative masks in either of the horizontal and vertical directions. This technique requires sifting the color or intensity information of the picture with the following filter kernels:

$$[\,-1,\ 0,1]\ \text{ and }\ [\,-1,\ 0,\ 1]^T$$

This is the mask which is used in computing the gradient of image.

For color images, we calculate separate gradients for each color, channel,

and take the one with the largest norm as the pixel's gradient vector.

- **Orientation binning**

The next step in HOG is to create histograms for each cell of the image. Cells are pixel regions that are either rectangular or radial in shape, and the histogram bins are evenly expanded from 0^o to 180^o or from 0^o to 360^o. 0^o to 360^o used in the case of signed orientation only. Every pixel in the cell has a weighted voting into one of the 9 histogram bins to which its orientation belongs. Weights can either be the gradient magnitude itself, or some function of the magnitude, for example, the square root or square of the gradient magnitude, or

some clipped version of the magnitude. Magnitude itself gives the best outcomes. Taking the square root diminishes execution marginally, while utilizing binary edge nearness voting diminishes it significantly. Along these lines, when all is said in done, the gradient magnitude is specifically utilized. Expanding the quantity of introduction bins enhances execution significantly up to around 9 bins, however has little effect before this. This is for bins spaced more than 0◦–180◦, i.e. the "sign" of the gradient is ignored. Counting marked gradients(orientation go 0◦–360◦, as in the first SIFT descriptor) diminishes the execution, notwithstanding when the quantity of bins is additionally multiplied to protect the first introduction determination. For people, the extensive variety of attire and foundation hues probably makes the indications of complexities uninformative. However take note of that including sign data helps considerably in some other question acknowledgment undertakings, e.g. autos, motorbikes.

- **Descriptor blocks**

Now, group the cells into blocks to normalize the gradient strengths. These blocks overlap with neighboring blocks so every cell can contribute its orientation distribution more than once. For the most part, there exists two types of blocks : R-HOG and C-HOG. R-HOG blocks are generally square grids which are represented to by three parameters: the quantity of cells per block, the quantity of pixels per cell and the quantity of channels per cell histogram. C-HOG blocks are especially of circular grid which can be depicted with four parameters: the quantity of angular bins, the quantity of radial; bins the radius of the center bin and the extension factor for the radius of additional radial bins..

R-HOG block: R-HOG block have numerous similarities to SIFT descriptors [21] yet they are utilized in an unexpected way. They are computed on dense grids at a single scale without prevailing orientation arrangement and utilized as a feature of a bigger code vector that implicitly encodes spatial position in respect to the detection window, while SIFT's are figured at an inadequate arrangement of scale-invariant key points, turned to adjust their dominant orientations, and utilized separately. SIFT's

are upgraded for sparse wide baseline matching, R-HOG's for dense robust coding of spatial form. Different precursors incorporate the edge orientation histogram of freeman and Roth [22].

C-HOG block: Our circular Contexts [23] aside from that, vitally, each spatial cell contains a pile of inclination weighted introduction cells rather than a solitary introduction autonomous edge nearness number. The log-polar matrix was initially recommended by the possibility that it would permit fine coding of close-by structure to be consolidated with coarser coding of more extensive setting, and the way that the change from the visual field to the V1cortex in primates is logarithmic [24]. However little descriptors with not very many spiral containers end up giving the best execution, so by and by there is little in homogeneity or setting. It is most likely better to consider C-HOG's basically as a propelled type of Center-encompass coding. We assessed two invariants of the C-HOG geometry, ones with a solitary round focal cell (like the GLOH highlight of [25]), and ones whose focal cell is isolated into precise segments as fit as a fiddle settings. We exhibit comes about just for the roundabout focus variations, as these have less spatial cells than the isolated focus ones and give a similar execution by and by. A specialized report will give additionally points of interest. The C-HOG format has four parameters: the quantities of rakish and spiral canisters; the sweep of the focal receptacle in pixels; and the extension factor for ensuing radii. No less than two outspread containers (a middle and an encompass) and four precise receptacles (quartering)are required for good execution. Counting extra outspread receptacles does not change the execution much, while expanding the quantity of precise canisters diminishes execution (by 1.3% at 10−4 FPPW while going from 4 to 12 rakish containers). 4 pixels is the best range for the focal container, yet 3 and 5 give comparative outcomes. Expanding the development reality or from 2 to 3 leaves the execution basically unaltered. With these parameters, neither Gaussian spatial weighting nor backwards weighting of cell votes by cell territory

changes the execution, however joining yet consolidating these two lessens somewhat. These qualities accept fine introduction testing. Shape settings (1orientationbin) require much finer spatial subdivision to function admirably.

- **Block Normalization**

There are four different ways to normalize the blocks. Let *v* be the non-normalized feature vector that collects all cell histograms from a given block, be its *k-norm* for *k* = 1,2 and *eps* is some constant. Then the normalization schemes have the following forms:

$$\text{L1-norm : } \hat{v} = \frac{v}{(||v||_1 + eps)} \tag{3.1}$$

$$\text{L2-norm : } \hat{v} = \frac{v}{\sqrt{||v||_2^2 + eps^2}} \tag{3.2}$$

$$\text{L1-sqrt : } \hat{v} = \sqrt{\frac{v}{(||v||_1 + eps)}} \tag{3.3}$$

L2-Hys is computed by re-normalizing the clipped L2-norm. All these normalization schemes provide much better performance than the non-normalized case. At last, the final HOG feature descriptor is then the vector containing elements of normalized cell histograms from all of the block regions.

3.2 Detection Method

In our work we use **Linear SVM** as a detection model. SVMs are set of related supervised learning strategies utilized for classification and regression [2]. They have a place with a group of generalized linear classification. An uncommon property of SVM will be, SVM all the while limit the empirical classification error and maximize the geometric margin. So SVM called Maximum Margin Classifiers. SVM depends on the Structural risk Minimization (SRM). Support Vector Machines (SVM) recently became one of the most popular methods for

detection and classification. They have been used in a wide variety of applications such as facial expression recognition [13], gene analysis [7] and many others. A few late investigations have revealed that the SVM (support vector machines) by and large are fit for conveying higher execution as far as characterization precision than the other information arrangement calculations. A SVM model is a representation of points in a space with the goal that the new purposes of various classifications are separated by a line or gap. New points are then mapped into that space and their classification is anticipated based on the side of the gap they fall on. An uncommon and similarly essential property of SVMs is that they at the same time minimize the empirical classification and expand the geometric margin. Such SVMs are called Maximum Margin Classifiers. In SVM, two parallel hyperplanes are built on each side of the separating hyperplane. The separating hyperplane is the one that maximizes the margin i.e. the separation between the two parallel hyperplanes. Bigger the margin, better will be the generalization error of the classifier or detector.

Consider training set of n samples in the form of:

$$(x_1,y_1),(x_2,y_2),(x_3,y_3),\ldots..,(x_n,y_n). \tag{3.4}$$

where y_i is the label of class to which x_i belongs, where x_i is a *d*-dimensional vector and $y_i \epsilon \{+1,-1\}$. We aim at identifying a hyperplane which can separate x_i with label $y_i = +1$ with those having labels -1. Any hyperplane can be written as the set of points x satisfying.

$$w.x + b = 0 \tag{3.5}$$

Where b is scalar and w is d-dimensional vector. Including the offset parameter b enables us to expand the margin. Missing of b, the hyperplane is compelled to go through the starting point, limiting the solution. As we are interested in the maximum margin, we are interested in SVM and the parallel hyperplanes. Parallel hyperplanes can be depicted by equation

$$w.x + b = 1 \tag{3.6}$$

$$w.x + b = -1 \quad (3.7)$$

In the event that the training data are linearly separable, we can choose these hyperplanes so that there are no points between them and afterward try to expand their distance. By geometry, we discover the separation between the hyperplane is 2/|w|. So we need to limit |w|. To excite data points focuses, we have to guarantee that for all I either

$$w.\ x_i - b \geq 1 \quad \text{or} \quad w.\ x_i - b \leq -1 \quad (3.8)$$

This can be written as

$$yi\ (w.\ x_i - b) \geq 1 \quad ,\ 1 \leq i \leq n \quad (3.9)$$

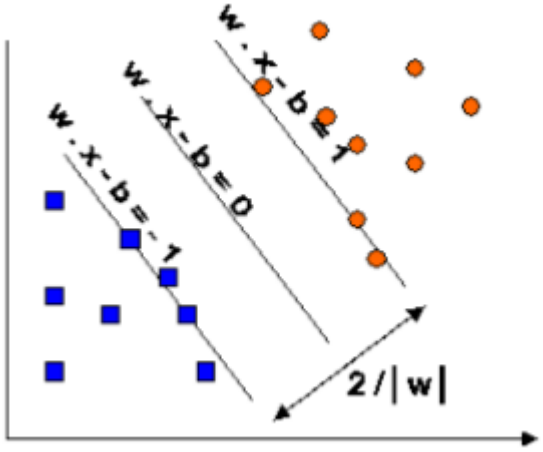

Figure 3.1 Maximum margin hyperplanes for a SVM trained with samples from two classes

Samples along the hyperplanes are called Support Vectors (SVs). A separating hyperplane with the biggest margin characterized by $M = 2/|w|$ that determines support vectors implies training data points closets to it.

SVM-based methods, including supervised, semi-supervised and unsupervised approaches.

3.2.1 Kernal Selection of SVM

Training vectors xi are mapped into a higher (might be limitless) dimensional space by the capacity Φ. At that point SVM finds a straight isolating hyperplane

with the maximal edge in this higher measurement space .C > 0 is the punishment parameter of the blunder term. Besides, K(x_i,x_j) ≡ Φ(x_i)^T, Φ(x_j) is known as the part work [2]. There are numerous portion capacities in SVM, so how to choose a decent part work is additionally an examination issue. Be that as it may, for general purposes, there are some well-known part works [2] and [3].

• Linear kernel: K (x_i , x_j) = x_i^T x_j.

• Polynomial kernel: K (x_i , x_j) = (x_i^T x_j + r)d , γ > 0

• RBF kernel : K (x_i , x_j) = exp(-γ $\|x_i - x_j\|^2$) , γ > 0

• Sigmoid kernel: K (x_i , x_j) = tanh(γ $x_i^T x_j$ + r)

Here, γ, r and d are kernel parameters. In these popular kernel functions, RBF is the main kernel function because of following reasons:

1.The RBF part nonlinearly maps tests into a higher dimensional space not at all like the genuine bits.

2. The RBF part has less hyper parameters than the polynomial portion.

3. The RBF part has less numerical challenges.

3.2.2 Model selection of SVM

Model selection is likewise a critical issue in SVM. As of late, SVM have indicated great execution in information grouping. Its prosperity relies upon the tuning of a few parameters which influence the speculation mistake. We regularly call this parameter tuning methodology as the model determination. In the event that you utilize the straight SVM, you just need to tune the cost parameter C. Tragically, direct SVM are frequently connected to straightly distinct issues. Numerous issues are non-directly detachable. For instance, Satellite information and Shuttle information are not directly divisible. In this manner, we regularly apply nonlinear bit to take care of grouping issues, so we have to choose the cost parameter (C) and piece parameters (γ, d). We for the

most part utilize the framework seek strategy in cross approval to choose the best parameter set. At that point apply this parameter set to the preparation dataset and afterward get the classifier. From that point forward, utilize the classifier to characterize the testing dataset to get the speculation exactness.

SVM-based methods, including supervised, semi-supervised and unsupervised approaches:

Supervised Methods:

The most well-known approach comprises of a weighted mix of SVMs learned in the source domain and SVMs learned in the target domain [35], [36], [37], [38]. The vital downside of these strategies is that they require both source and target domain training data for the adjustment, which makes it computationally costly. It might even outcome in negative transfer (i.e., the exactness diminishes for the target domain) as revealed in [38]. Then again, an feature replication approach is proposed in [40], which together learns classifiers in the both domains with augmented features, i.e., source-area data is additionally required. Another approach, the cross-area SVM (CD-SVM) [39], chooses the source domain support vectors that are near the target space and furthermore includes new support vectors from the target domain to take in another classifier. In any case, for the situation that the target domain data are rare, the educated classifier may at source domain oriented.

Semi-supervised / Unsupervised Methods:

The domain transform SVM (DT-SVM) of [41] limits the appropriation mismatch of labeled and unlabeled examples between various domains. The transductive SVM (TSVM) is utilized in [37] for enhancing the exactness of classifiers prepared with weakly labeled web images. The transform based techniques [42], [43] utilize labeled source and unlabeled target data to build a complex and take in a classifier from an projected space.

3.3 PCA

Principal component analysis (PCA) is probably the most prominent multivariate statistical procedure and it is utilized by all scientific disciplines. It is likewise

liable to be the most established multivariate technique.

The principle thought behind utilizing Principal Component Analysis is to outline the basic fluctuation covariance structure of a substantial arrangement of factors through a couple of straight mixes of these factors. Key Component Analysis is a prevalent system for information pressure and has been effectively utilized as an underlying stride in numerous PC vision assignments like face or protest acknowledgment. Since designs in high dimensional information can be elusive, PCA is an effective apparatus for investigating information and lessen the dimensionality.

Since it is a variable lessening technique, primary part investigation is comparative in many regards to exploratory factor examination. Indeed, the means took after when directing a foremost part investigation are for all intents and purposes indistinguishable to those took after when leading an exploratory factor examination. Be that as it may, there are critical theoretical contrasts between the two systems.

There are some critical calculated contrasts between primary segment examination and factor investigation that ought to be comprehended at the start. Maybe the most critical manages the supposition of a basic causal structure: factor investigation expect that the variety in the watched factors is because of the nearness of at least one inert (factors) that apply causal impact on these watched factors.

When all is said in done terms, PCA utilizes a vector space change to lessen the dimensionality of substantial informational indexes. Utilizing numerical projection, the first informational index, which may have included numerous factors, can frequently be deciphered in only a couple of factors (the foremost segments). It is in this way regularly the case that an examination of the diminished measurement informational index will enable the client to spot patterns, examples and exceptions in the information, much more effortlessly than would have been conceivable without playing out the important part investigation.

Utilizing the Regression display with numerous factors that are very corresponded each other won't restore the best estimators [28]. In such situations when we endeavor to investigate an extensive arrangement of p factors that are typically abundantly connected and produce the multicollinearity

wonder, the PCA is prescribed.

Objectives OF PCA

The objectives of PCA are to

(1) remove the most essential data from the information table;

(2) compress the size of the informational index by keeping just this essential data

(3) improve the depiction of the informational collection; and

(4) investigate the structure of the observations and the variables.

Keeping in mind the end goal to accomplish these objectives, PCA registers new factors called main segments which are acquired as direct mixes of the first factors. The first key segment is required to have the biggest conceivable difference (i.e., dormancy and thusly this segment will "clarify" or "separate" the biggest piece of the idleness of the information table). The second segment is processed under the requirement of being orthogonal to the first segment and to have the biggest conceivable inactivity. Alternate segments are processed moreover. The estimations of these new factors for the perceptions are called factor scores, and these variables scores can be translated geometrically as the projections of the perceptions onto the essential parts.

Attributes of vital segments: The main part separated in a central segment investigation represents a maximal measure of aggregate fluctuation in the watched factors. Under run of the mill conditions, this implies the primary segment will be related with at any rate a portion of the watched factors. It might be related with numerous.

The second segment extricated will have two critical qualities. In the first place, this part will represent a maximal measure of fluctuation in the informational index that was not represented by the primary segment. Again under commonplace conditions, this implies the second part will be connected with a portion of the watched factors that did not show solid relationships with segment 1. The second normal for the second part is that it will be uncorrelated with the primary segment. Truly, if you somehow happened to register the relationship between segments 1 and 2, that connection would be zero.

For PCA to work appropriately, standardize the information over mean esteem and create an informational index whose mean is zero.Consider a set of N, d-

dimensional data images. Each image I_i is represented by a d-dimensional vector v_i. The mean and covariance matrix is defined by:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{3.10}$$

$$C = \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T \tag{3.11}$$

Now calculate the eigenvectors and eigenvalues of the covariance matrix C. The principal components are basically eigenvectors of C.

We have a $d \times d$ covariance matrix C. Solve for,

$$|C - \lambda I| = 0 \tag{3.12}$$

to get d eigenvalues, $\lambda_1, \lambda_2 \ldots \lambda_d$.

Since the eigenvectors corresponding to the lowest eigenvalues have the least information about the distribution of data. These are the eigenvectors which can be dropped without losing much information for the construction of lower-dimensional subspace. Eigenvectors corresponding to top *K* eigen values form the new lower dimensional space.

Chapter 4

Methodology

Here in this chapter, we presents novel algorithm for detection which we have done. The very first step is to collect all the labeled data from source video and unlabeled data from target video[Algorithm 1]. Next calculate subspace using PCA by selecting the top *d* eigenvectors, to train the detector[Algorithm 2]. We learn transformation matrix by minimizing the Bregman matrix divergence [1]. This matrix is used to transforms the source subspace coordinate system into the target based source subspace coordinate system by aligning the source basis vectors with the tar-get ones. Now map the source data to the new subspace to get training data. Learn the SVM using this training data in d-space. At the time of testing, iteratively learn new detector by the help of detected bounding boxes[Algorithm 3].

4.1 Generating Training Samples Using Hard Negative Mining

Generate a detector model which have enough samples to detect the object in the test video. We need training samples from source video to generate the initial detector model. As an input, we have train and test video along with their annotation files. We need test video annotation file only to calculate the accuracy(precision and recall) of the model. Competitive sliding window detectors require vast training sets.

Algorithm 1 Data Collection

1: **Given:** Source Video V_S, Target Video V_T , Source Annotation File A_S,

Object Class C

2: **Init:** Positive labeled data P = {}, Negative labeled data N = {}, Number of random samples R = 100, δ= 100

3: **for** each frame $i \in V_S$ **do**

4: $H_n \leftarrow 0$

5: $P \leftarrow$ Extract & Compute features of object ∈ C from i using A_S

6: $N \leftarrow$ Extract & Compute features of object ∉ C from i using A_S

7: $N \leftarrow$ Extract & Compute features of δ random samples from i

8: simpleSVM ← trainSVM(P,N)

9: boundingBoxes ← runsimpleSVM(i)

10: N ← FP(boundingBoxes) using A_S

11: δ ← R - CountofFP(boundingBoxes)

12: end for

13: S ← merge(P ,N) . S is Source Data

14: for each frame $i \in V_T$ **do**

15: matrixM ← ForegroundMaskUsingMoG(i)

16: matrixM ← Filters(matrixM)

17: AllContours ← detectContours(matrixM)

18: **for** each contour $k \in$ AllContours **do**

19: T ← Extract & Compute features of k . T is Target Data

20: end for

21: end for

Since a set of all the possible window sized images as patches at various scales and location from a frame gives an about unending support of negative samples. Preparing with all the accessible data is viewed as impractical. A staple of current methodologies is Hard Negative Mining, a strategy for choosing hard and relevant samples, which is nevertheless costly. Given that samples at marginally unique areas have overlapping support, disparity between the resolution of prediction and learning has been handled by mining for hard negative examples. In this iterative procedure, an underlying model is prepared utilizing every positive case and an arbitrarily chose subset of negative examples, and this initial training set is progressively augmented with false positive cases created while examining the images with the model learned so

far. Hard negative mining is considered expensive as it trains the model repeatedly.

Algorithm 2 Generate Subspace & Train Detector

1: **Given:** Source data *S*, Target data *T* , Subspace dimension *d*
2: $X_S \leftarrow$ runPCA(*S,d*)
3: $X_T \leftarrow$ runPCA(*T ,d*)
4: $M \leftarrow X_S' X_T$. X_S' is transpose of X_S & *M* is Tranformation Matrix
5: $X_a \leftarrow X_S M$. X_a is New Coordinate System
6: trainData $\leftarrow SX_a$
7: detectorModel ← LinearSVM(trainData, Labels)

Algorithm 3 Use of Online samples to Generate Detector

1: **Given**: Source data *S,* Target data *T* , Subspace dimension *d*, Initial detector-Model, Positive data *P* , Negative data *N*, Target Video V_T
2: **for** each frame $i \in V_T$ **do**
3: boundingBoxes ← rundetectorModel(*i*)
4: **for** each bounding box B ∈ boundingBoxes **do**
5: Use Sliding Window Method for *B* in *i* to Categorize into TP & FP
6: Add B into S with Label
7: **end for**
8: Generate subspace and Calculate trainData using Algorithm 2
9: detectorModel ← trainSvm(newtrainData)
10: **end for**

4.2 Background Subtraction Using MOG

Background subtraction is fundamentally detecting moving objects in videos utilizing static cameras. The fundamental thought in this approach is to identify the moving items by taking the contrast between current frame and a reference frame, which is called "background image". The background image must be adequate to represent the scene with no moving items. Get the foreground

mask using *MIXTURE OF GAUSSIANS*. This model is initialized using an EM(expectation maximization) algorithm. The Gaussians are manually labeled in a heuristic manner as follows: the one with the largest variance or variance higher than some threshold is labeled as foreground else background. Each pixel is compared with each Gaussian and is classified according to its corresponding Gaussian to find foreground. The maintenance is made using an incremental EM algorithm for real-time consideration [20, 16].

We know that pixels are characterized by its intensity in the RGB color space. So each pixel in the image is modeled by a mixture of K Gaussian distributions. The probability of observing current pixel value is given by the following formula:

$$P(X_t)=\sum_{i=1}^{K} w_{i,t} * g(x_t, \mu_{i,t}, \sum i,t) \quad (4.1)$$

Where $w_{i,t}$ is the estimate of the weight of the i^{th} Gaussian at time t, $\mu_{i,t}$ is the mean value of the i^{th} Gaussian at time t, $\sum i,t$ is the covariance matrix of the i^{th} Gaussian probability density function and *g* is a Gaussian probability density function:

$$g(x_t, \mu, \sum) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu_t)^T \Sigma^{-1}(X_t-\mu_t)} \quad (4.2)$$

Where covariance matrix is represented as:

$$\sum i,t = \sigma_i^2 I \quad (4.3)$$

Along these lines, every pixel is described by mixture of K Gaussians. The K distribution depend on wellness value w_t/σ_t. Every new pixel value X_t is checked against the current K Gaussian distribution until the point that a match is found. K chose the multimodality of the foundation and by the accessible memory and computational power. Once the parameters instatement is made, a first frontal area identification can be made and after that the parameters are refreshed. This requesting assumes that a foundation pixel relates to a high weight with a

frail fluctuation because of the way that the foundation is more present than moving articles and that its esteem is for all intents and purposes consistent. A match is characterized as a pixel esteem inside 2.5 standard deviations of distribution. The Gaussian model will be updated by the following update equations,

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t \tag{4.4}$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \tag{4.5}$$

where,

$$\rho = \alpha g(X_t \mid \mu_k, \sigma_k) \tag{4.6}$$

Then, two cases can occur:

Case 1: A match is found with one of the K Gaussians. For this circumstance, if the Gaussian conveyance is recognized as a background one, the pixel is classified as background else the pixel is named foreground.

Case 2: No match is found with any of the K Gaussians. For this circumstance, the pixel is classified foreground**.**

At this progression, a binary mask is gotten. By then, to make the following foreground detection, the parameters must be revive.

4.2.1 Relevance and approximations made:

Demonstrating the foundation utilizing the MOG suggests the supposition that the foundation conveyances and the foreground ones are Gaussians however it isn't generally the case. For instance, Kitahara et al. [29] demonstrate that the appropriation in indoor scene is considerably more like a Laplace show than a Gaussian one. In another way, Wang et al. [30] comment that the power doesn't fit in with the Gaussian disseminations when the force differs unexpectedly like on account of glinting trees (MB) in open air scene. For the introduction, the MOG needs that the quantity of Gaussians K is settled and is the same for all pixels. It isn't ideal in light of the fact that the multimodality is variable spatially

and transiently. For the instatement of the mean, the change and the weight, a progression of preparing outlines missing of moving articles is required however in some condition, it isn't conceivable to get outlines without moving items. Moreover, this stockpiling causes a measure of memory required in this progression. For the support stage, Greiffenhagen et al. [31] portrays it factual conduct making diverse parameters introduction utilizing genuine information and recreated information. The examination demonstrates that lone the methods are evaluated and followed accurately. The fluctuation and the weights are unsteady and questionable however Greiffenhagen et al. [31] comment this is not by any means an issue in light of the fact that the weights and the changes are not utilized as a part of a resulting handling step. For the frontal area identification, the primary disadvantage is fundamentally because of the coordinating test as clarified in [32]. In reality, the upkeep is made by the grouping utilizing this coordinating test which is a guess of the MAP. The outcome is that the tail of the circulation is not refreshed when the support is just made when the new esteem is between this interim given by the mean and standard deviation. At the point when just the piece of the appropriation characterized by this interim is utilized, another Gaussian part is evaluated which are a lower standard deviation. Along these lines, the standard deviation moves toward becoming belittled and qualities in the tail are wrongly classified foreground. This reason issue when the foundation is not refreshed because of the mistake in the grouping. This gives more pixels classified closer view causing foreground location. This issue expanded after some time. For the element size, Stauffer and Grimson [34] have choosen the pixel however this pixel-wise angle has the fundamental disservice that the worldly and spatial requirements aren't handled. For the element sort, Stauffer and Grimson [34] utilized the RGB space however these shading segments aren't autonomous thus the improvement made in Equation (3.3) for the covariance network isn't right. This simplification conducts to false positive and false negative detections.

4.2.2 Dealing with the challenges:

The MOG display manages the development out of background (MB) because of the multimodality in the portrayal step. The support step licenses to adapt up to the steady enlightenment changes (TD) and the learning rate a decides the

speed of adjustment to light changes (TD) yet additionally the speed of the consolidation of background objects moved or embedded (MBO, IBO) and the speed of fuse of a moving item which halted (SFO) [33]. This is one of the disservices of the MOG and for the most part in the writing the writers make a trade-off between the two procedures. Another disservice is that the pixel-wise angle anticipates to deal with some basic circumstances (LS, B) which can be just distinguished spatially and transiently. Besides, some basic circumstances require pre-preparing or post-handling (NI, CJ, CA). For these two sorts of basic circumstances, Stauffer and Grimson [34] proposed nothing to manage it. Another burden is the utilization of the RGB which can allow to make well shadows recognition (S). In continue, the first pixel-wise MOG display is configuration well for (TD, MB), is medium for the (MBO, IBO, SFO), and isn't outline for the (NI, CJ, CA, LS, B, C, FA, WFO, S).

To understand these diverse restrictions, numerous upgrades can be discovered that can be named natural and extraneous model changes.

- Intrinsic demonstrate enhancements

Intrinsic model upgrades concern specifically the MOG show like the initialization and the support of the parameters, the foreground detection and by expansion the elements utilized.

- Extrinsic demonstrate upgrades

The efficacy and strength can be enhanced by utilizing the learning of temporal and spatial data in the external strategies. The diverse methodologies can be found by utilizing:

- Markov Random Fields
- Hierarchical approaches
- Multi-level methodologies
- Multiple backgrounds
- Multi-layer approach

4.3 Sliding Window Method

After running the model on the test frame, we get some detected bounding boxes. We use these bounding boxes as training samples to train the model

again. In experiments we have shown that accuracy of initial model is not well. So it may be possible that some bounding boxes generated by the model are false positive. To categories the bounding boxes in True positive and false positive, we use Sliding Window Method. We use the frame again to show whether the box is true positive or false positive. Extract the new window by sliding the bounding box for some threshold in the frame and find the score by running the detector for that extracted window. Choose some δ value which indicates the number of times, we slide original window in the frame.

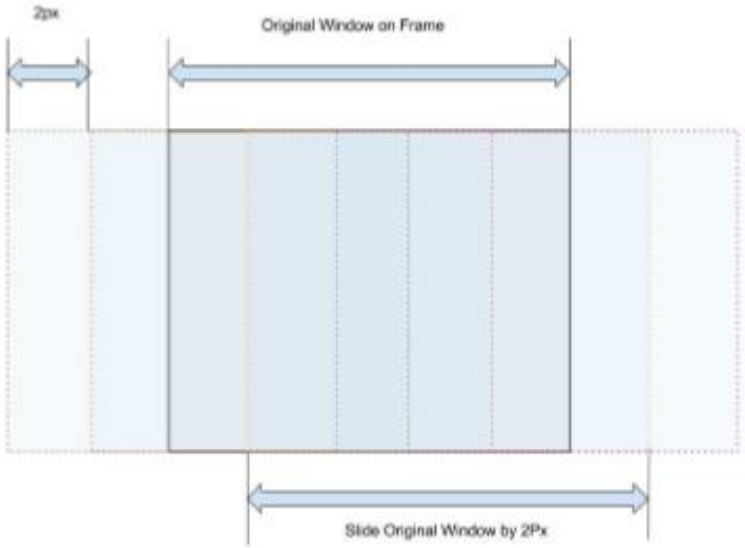

Figure 4.1: *Slide Window by 2Px for δ=4.*

After choosing the δ value, iterate the method for δ times. Here first choose the test window by sliding the original window by some i^{th} pixels. Then check it whether it is in the frame or not. If it is in the frame then extract the portion of the frame for the corresponding window. Normalize this RGB image into grayscale image. Then find the score by running the detector model over this matrix. If the score is above some threshold, we mark this sliding window as positive, else negative. Take the average of positives and negatives found by this procedure. If the average taken is above zero or some *threshold,* call the original window as true positive otherwise false positive.

4.4 Subspace Based Adaptation with Online Samples

Even though source and target data lie in same D-dimensional space, they have been placed according to different marginal distributions. Adaptation technique shifts the space between these two domains. To do this, generate subspace for both source and target subspace. Now normalize the every D-dimensional source and target data by shifting it to zero mean and unit standard deviation. We select d eigenvectors corresponding to the d best eigenvalues by using PCA. These eigenvectors are used as bases of the source and target subspaces. These bases denoted by X_S and X_T where X_S and X_T is in R^{Dxd}. This X_S and X_T are used to learn the shift between the source and target domains.

Project each of source and target data into their respective subspace X_S and X_T. Next align the source subspace coordinate system into target one, using linear transformation function. In this new generated subspace we can directly compare our both subspaces, without any projections. Use a subspace alignment approach to do this. We learn transformation matrix by minimizing the Bregman matrix divergence [1] :

$$F(M) = \|X_S M - X_T\|_F^2 \quad (4.7)$$

Where $\|\ldots\|_F^2$ is the Frobenium norm. Our optimal transformation matrix M is given as:

$$M^* = argmin_M (F(M)) \quad (4.8)$$

Optimal M is obtained as $M^* = X_S' X_T$ [1]. Matrix M changes the source subspace coordinate system into the target subspace coordinate system by adjusting the source basis vectors with the target ones. In the event that a source basis vector is orthogonal to all target basis vectors, it is ignored. On the other hand, a high weight is given to a source basis vector that is well aligned with the target basis vectors. So, the new coordinate system is same as $X_S M^*$ or $X_S X_S' X_T$ called as target aligned source coordinate system. Now we need to project the source data via $X_S X_S' X_T$ into target aligned source subspace to train the detector model.

Then learn a detector from this d-dimensional space. To test an image whether it is true or false, we need to project the target data into the target subspace(by the help of X_T). One more problem comes here, that is we can't project out target data into target subspace(using X_T) as it, not a classification task but an object detection. We have a frame of the target data and to detect the class object we used sliding window protocol. To run the detector on target data, we need to project each of the windows into the target subspace. It is very complex and costlier to do so. We can always project our Support vectors from lower dimension to higher dimension. So instead of doing the projection for each of window, we project support vectors into the target subspace. Consider S_V as the original support vectors then we get new support vectors S_{NV} by projecting S_V on target subspace(using X_T):

$$S_{NV} = S_V X_T \tag{4.9}$$

Now we test our target video on this new detector.

Chapter 5

Experiments and Results

Here in this section, we explain about experimental setup, the dataset used for experiments and produced results.

5.1 Experimental Setup and Dataset

The proposed algorithm was implemented in C++11 with OpenCV 3.0. We used *VIRAT* video dataset to train and test our detector models. These videos are captured by stationary HD cameras and each video has its own annotation file which carry information about bounding box of objects in each frame of the video. These annotation files help us to collect the training dataset for a particular class of object from the video. Annotation files of VIRAT dataset contain information about 5 different classes of objects. Train and test video have different domains in our experiments. The *virat_s*_050000_03 sequence is used as training video and *VIRAT_S*_040103_02 sequence as test video. These sequences have 1607 and 2392 frames respectively, each of size 1920 X 1080. All extracted objects from source and target video, are scaled to the size of 64x64. Features are extracted and computed using HOG, and thus, descriptor vectors are 1764-dimensional. As a detector model, we used Linear SVM with C=0.01.

5.2 Results

Experiments are conducted extensively for different methods using adaptation and online samples on videos. Inferences obtained from results are analyzed in

detail. For each method, average precision over all the frames is plotted against the number of top k bounding boxes in precision@K.

5.2.1 Test using HOG and SVM

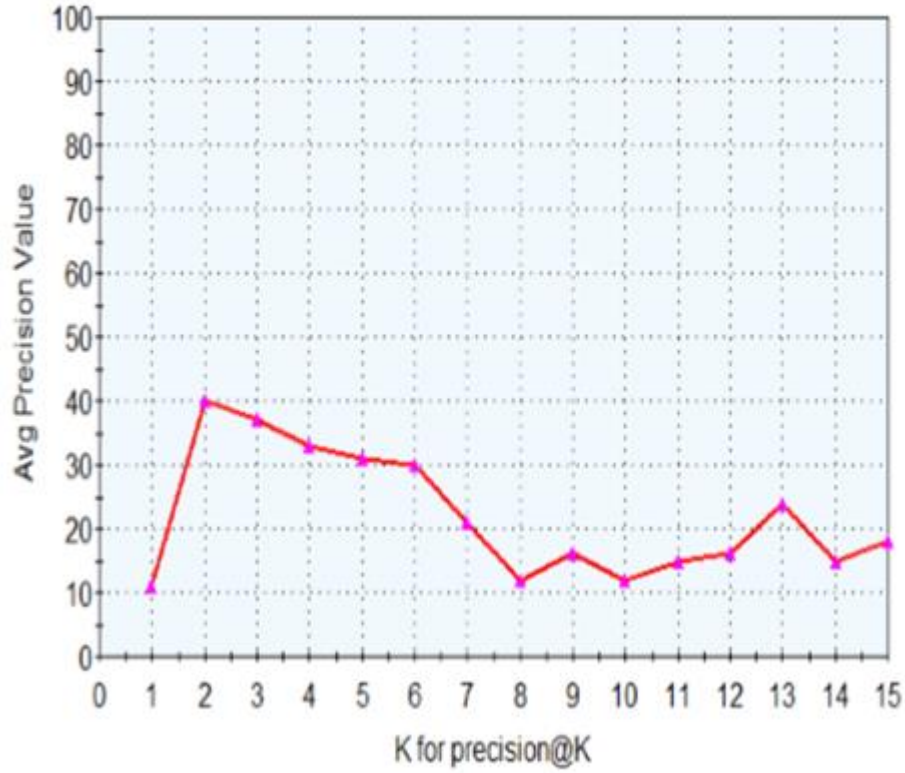

Figure 5.1: Method1 *(Object detection using HOG+SVM)*

Here in this method, we train the model on HOG features, extracted from source or target video. Some of the parameters are to be set in HOG for the best results. The *padding* switch controls a number of pixels the ROI is padded with prior to HOG feature vector extraction and SVM classification. *cellSize* and *blockSize* set to be [8 x 8] and [16 x 16] respectively. Our SVM detector was trained on “car” object class. Difference in domains of videos accounts for the low accuracy. Figure 5.1 shows the relation between precision value and K where K indicates the top bounding boxes. The mean precision and recall over target video using this method are 51 and 19 percent respectively.

5.2.2 Test using Adaptation with Target Random Samples

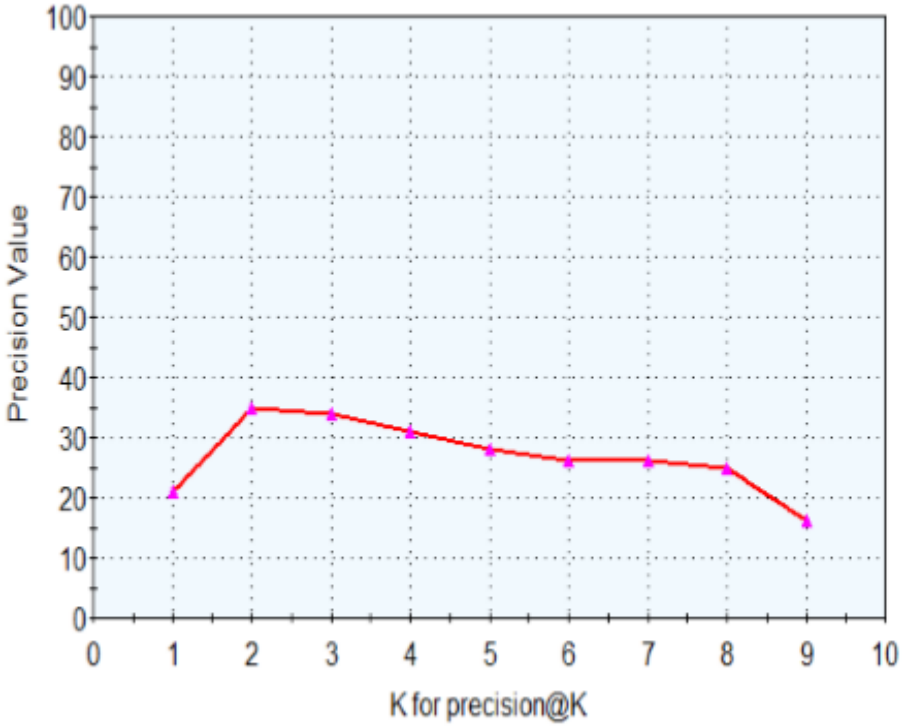

Figure 5.2: Method2 *(Object detection using HOG + DA (Random Samples) +SVM)*

Adaptation is based on the subspaces generated from source and target video. Here in this section, we take 100 (it may vary) samples from each of the frame target video and then generate target subspace using this random sample. Overall accuracy is not affected much, since it may happen that no target class object appears in the subspace due to randomness. The mean precision graph for this method is shown in Figure 5.2.

5.2.3 Test using Online Samples

In this section, we use bounding boxes generated by the weak detectors. Iterate the method over δ times to get a good detector model using online samples. To get the best δ value, we experimented with a range of values and for $\delta = 20$, we get best detector model. Given Figure 5.3 shows that nearly all top bounding boxes belong to true positives. This method shows a slight improvement in accuracy (Figure 5.3) but still not as expected. The mean precision and recall for this method is 28 and 53 percent respectively.

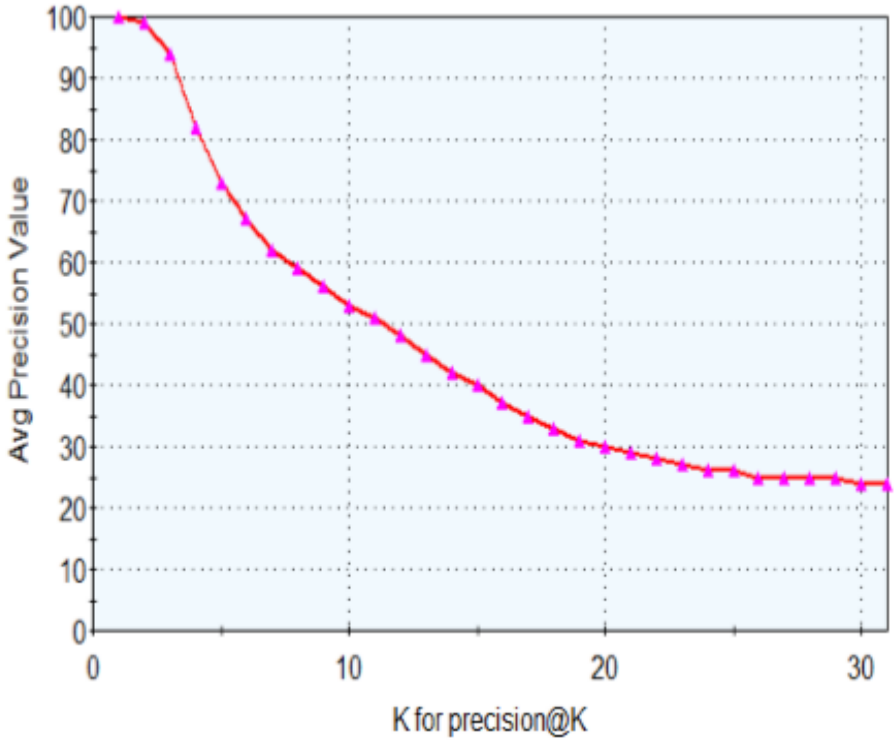

Figure 5.3: Method3 *(Object detection using HOG + SVM + Online Samples)*

5.2.4 Test using Adaptation with foreground mask

As from the previous experiments, we know that random samples from target video will not work well to generate subspace. Here we go with the Mixture of Gaussians (MoG), for subtracting background from a frame to get foreground mask. It extracts all the moving objects in the target video. This data generates a good target subspace, resulting in higher accuracy. The mean precision and recall in this method are 42 and 41 respectively which are greater than the values achieved in all the previous methods. For this method, the plot of precision is shown in Figure 5.4.

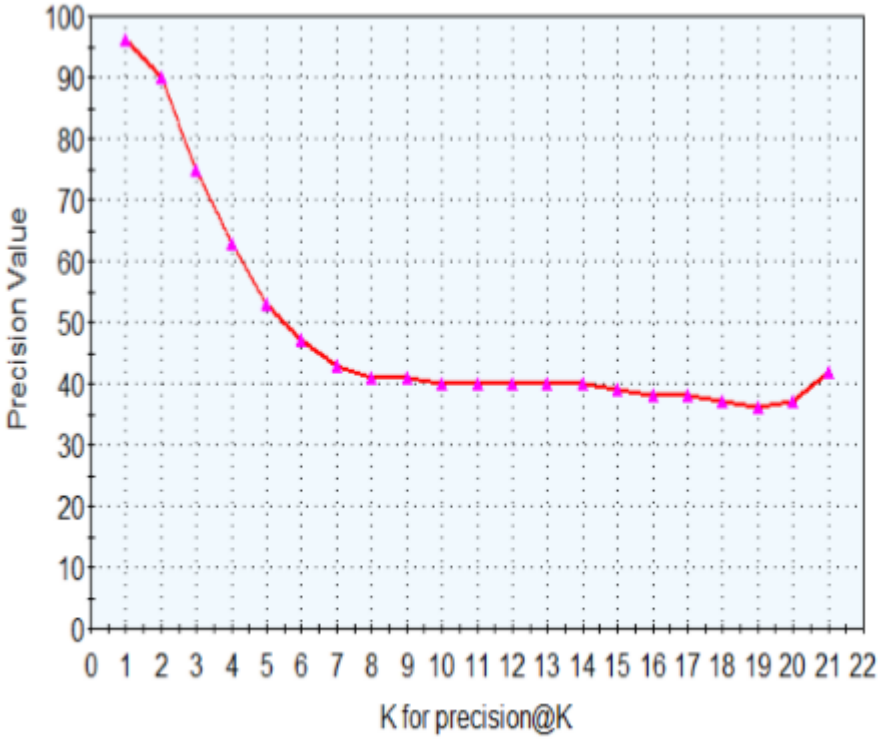

Figure 5.4: Method4 *(Object detection using HOG + DA (Foreground Mask) + SVM)*

5.2.5 Test using Adaptation with foreground mask and On-line Samples

As we notice in earlier experiments, domain adaptation and learning from online samples work well in object detection. We brought both of them together to get good results. The initial model is trained on source and target subspace, generated by foreground mask of target video. Run this model on the first frame of target video and get all the detected bounding boxes. Since all these detected bounding boxes may not be true positive, we use another method called sliding window, to categorize these detected bounding boxes into true positive and false positive. After getting these weakly labeled data, we train the detector model again. We have done this for first 20 frames and results were extremely well. A number of iteration may vary as it depends on target video. The mean precision and recall for this method is 56 and 60 percent respectively, which is extremely well as compared to all the previous methods. Precision@K graph for this method is shown in Figure 5.5.

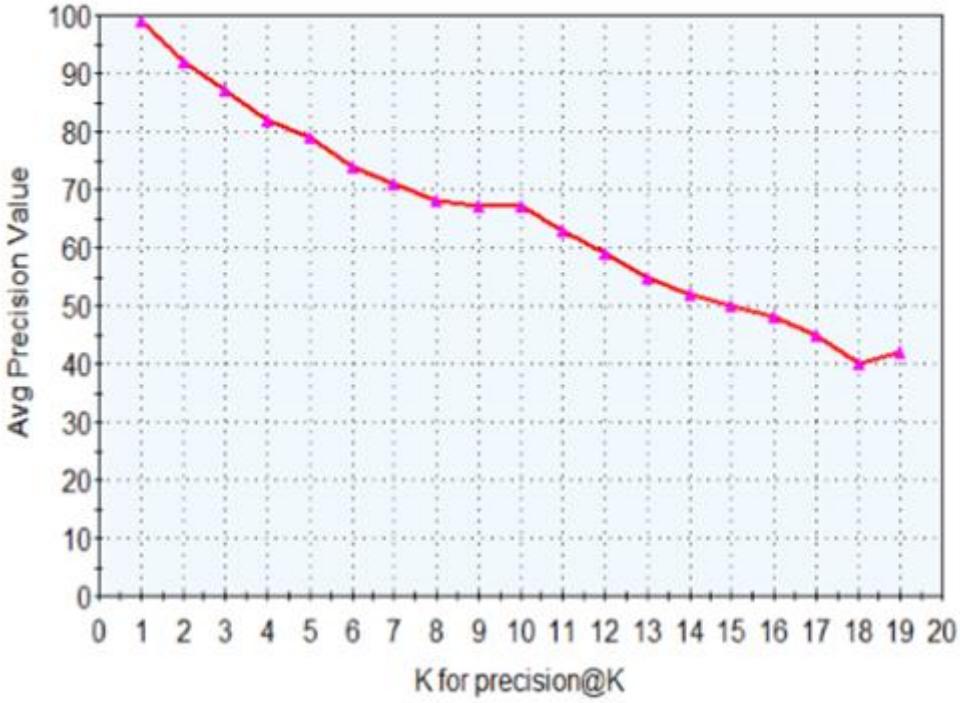

Figure 5.5: Method5 *(Object detection using HOG + DA (Foreground Mask) + SVM +Online Samples)*

5.2.6 Comparison of all Methods

The very first method we used is based on simply HOG and SVM (Method1). Here we extract the features of training data (From source video) without any use of target data and train the Linear SVM model on these features. Now use this model directly to detect objects in target video. So, this is the simplest method which we used in detection and as we see in the graph (Figure 5.6), results obtained are not so good. As video dataset has different domains, we align source subspace to target subspace. Choosing the random samples to calculate subspace of target video does not affect much in accuracy (Method2). So, we used background subtraction on target video, using MoG, to get foreground mask (Method4). This foreground mask is further used to calculate target subspace for adaptation. Now here we saw an improvement in accuracy. Iterative training using weakly labeled online samples gives fairly well results, as we see in Figure 5.6

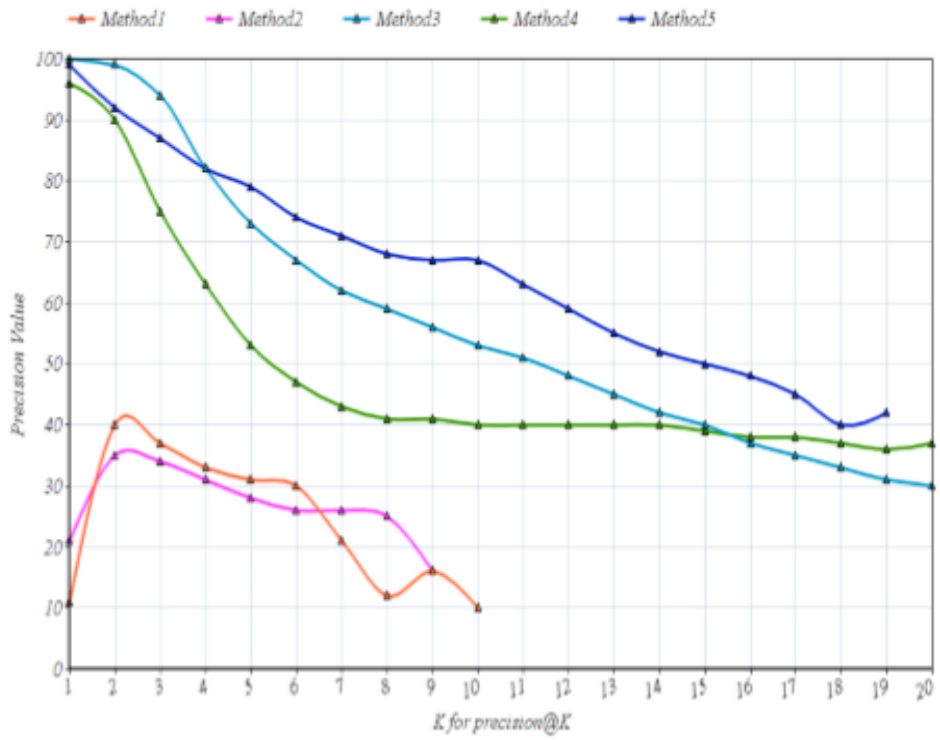

Figure 5.6: Precision@K graph of all methods

The graphs represents that use of adaptation in videos, using background subtraction to collect target data and learning from online samples, improves the accuracy of object detection. So we merge both of the method (Method3 and Method4) which gives extremely well results. The bar graph (Figure 5.7) shows mean precision and recall value for all the methods and both looks fairly well in the Method5.

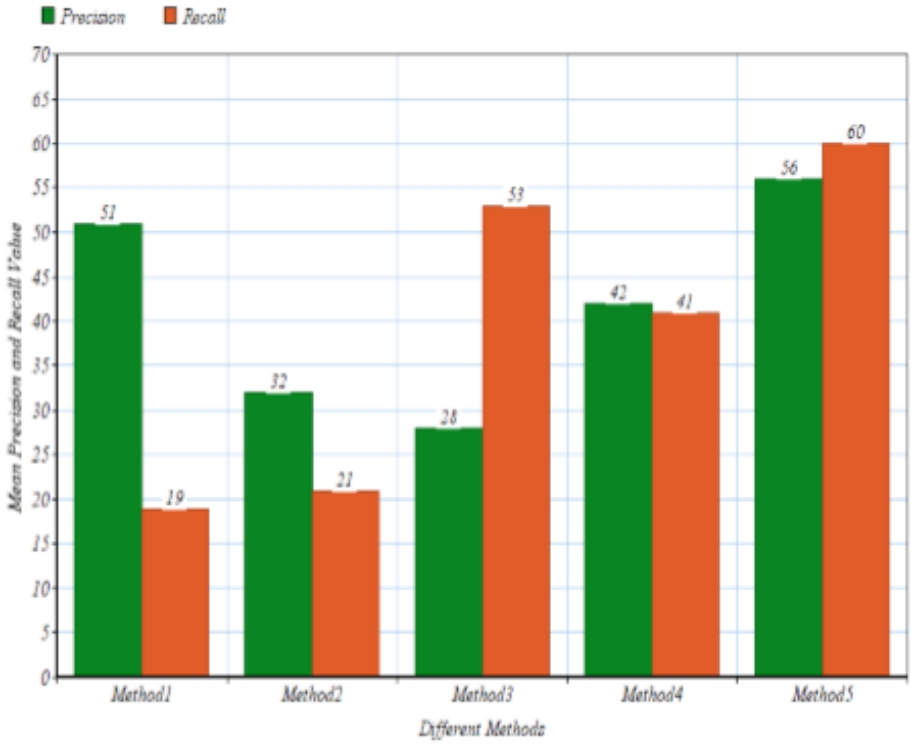

Figure 5.7: Mean Precision and Recall Value

Chapter 6

Conclusion

We introduced a novel method which uses online samples with subspace based adaptation in videos. Here in this method, first we created subspaces for both source and target video domains and then learned a linear mapping that aligns the source subspace with the target subspace. This allows us to build a detector model on source data in target based space which can be applied to the target video. We have collected data from target video using several different methods from each of the target video frames: extracting 100 random windows of 64x64, extracting 64x64 window using sliding window and background subtraction method which extracts all the foreground contours and resizes them to 64x64. Learning in this way from online samples gave fairly good results. Different methods to generate detectors with different parameters like block Size, cell Size, n bins(Number of bins) in HOG , *C*-value, ϵ in SVM and *d*-value to generate subspace using PCA, were compared simultaneously for VIRAT video datasets having different domains. Experimental results show that the introduced adaptation with learning from online samples method outperforms subspace based adaptation methods without learning which use HOG features and SVM classifier model.

Chapter 7

Future Scope

We have collected data from target video using several different methods from each of the target video frame and detected objects, as future work, we intend to improve performance of object detection in videos by using some other methods.

The novel method introduced by us by creating subspaces for both source and target video domains and builds a detector model on source data in target based space can be improved by using some other improved training methods.

.

Bibliography

[1] M. Sebban B. Fernando1, A. Habrard2 and T. Tuytelaars1. Unsupervised visual domain adaptation using subspace alignment. In ICCV, 2013.

[2] F. Sha B. Gong, Y. Shi and K. Grauman. Geodesic ow kernel for unsupervised domain adaptation. In CVPR, 2012.

[3] K. Saenko B. Kulis and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In CVPR, pages 1785{1792, 2011.

[4] H. Chang S. Shan X. Chen D. Zhai, B. Li and W. Gao. Manifold alignment via corresponding projections. In BMVC, 2010.

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection.

[6] B. Siddiquie R. S. Feris L. S. Davis F. Mirrashed, V. I. Morariu. Domain adaptive object detection.

[7] S. Barnhill I. Guyon, J. Weston and V. Vapnik. Gene selection for cancer classi cation using support vector machines. Machine Learning, 46(1-3):389{ 422, January 2002.

[8] D. Foster J. Blitzer and S. Kakade. Domain adaptation with coupled subspaces. In Conference on Arti cial Intelligence and Statistics (AISTATS), 15, 2011.

[9] R. McDonald J. Blitzer and F. Pereira. Domain adaptation with structural correspondence learning. In Conference on Empirical Methods in Natural Lan-guage Processing, pages 120{128, July 2006.

[10] Vzquez D L. AM J Xu, S Ramos. Domain adaptation of deformable part-based models. 36(12), 2014.

[11] M. Fritz K. Saenko, B. Kulis and T. Darrell. Adapting visual category models to new domains. In Computer Vision ECCV, 6314:213{226, 2010.

[12] L. Fei-Fei D. Koller K. Tang, V. Ramanathan. Shifting weights: Adapting object detectors from image to video. In IARPA.

[13] P. Michel and R. E. Kaliouby. Real time facial expression recognition in video

using support vector machines. In Proceedings of ICMI03, pages 258{264, 2003.

[14] S. Ali O. Javed and M. Shah. Online detection and classi cation of moving ob-jects using progressively improving detectors. Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pages 1063{6919, 2015.

[15] R. Li R. Gopalan and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In ICCV, 2011.

[16] B. Vachon T. Bouwmans, F. El Baf. Background modeling using mixture of gaussians for foreground detection - a survey. In Recent Patents on Computer Science, 1(3):219{237, 2008.

[17] A. Gupta T. Malisiewicz and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In ICCV, 2011.

[18] A. Torralba and A. Efros. Unbiased look at dataset bias. In CVPR, 2011.

[19] C. Wang and S. Mahadevan. Heterogeneous domain adaptation using manifold alignment. In IJCAI, 2011.

[20] H. Peng X. Wang, J. Sun. Foreground object detecting algorithm based on mixture of gaussian and kalman lter in video surveillance. 8(3):693{700, 2013.

[21] D. G. Lowe. Distinctive image features from scale-invariant key points. IJCV,60(2):91–110, 2004.

[22] W.T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. Intl.Workshop onAutomaticFaceand Gesture- Recognition, IEEE Computer Society, Zurich, Switzerland,pages296–301,June1995.

[23] S.Belongie, J. Malik, and J.Puzicha. Matching shapes. The 8thICCV, Vancouver,Canada,pages454–461, 2001.

[24] EricL.Schwartz. Spatial mapping in the primate sensory projection: analytic structure and relevance to perception. BiologicalCybernetics,25(4):181–194, 1977.

[25] K.Mikolajczyk and C.Schmid. A performance evaluation of local descriptors. PAMI,2004. Accepted

[26] R.K.McConnell. Method of and apparatus for pattern recognition, January1986. U.S.PatentNo.4,567,610.

[27] W. T. Freeman, K. Tanaka, J. Ohta, and K. Kyuma. Computer vision for computer games. 2nd International Conference on Automatic Face and Gesture Recognition, Killington, VT,USA,pages100–105, October1996.

[28] Smith, K., Sasaki, M.S.: Decreasing Multicollinearity: A Method for Models with Multiplicative Functions. In: Sociological Methods & Research, 8 (1979), no.1,

p. 35-56.

[29] Kim H, Sakamoto R, Kitahara I, Toriyama T, Kogure K. Robust silhouette extraction technique using background subtraction. 10th Meeting on Image Recognition and Understand (MIRU 2007), Hiroshima, Japan, July 2007.

[30] Wang D, Xie W, Pei J, Lu Z. Moving area detection based on estimation of static background. J Inform Comput Sci 2005; 2(1): 129-134.

[31] Greiffenhagen M, Ramesh V, Niemann H. The systematic design and analysis cycle of a vision system: A case study in video surveillance. IEEE Computer Society Conf on Computer Vision and Pattern Recognition (CVPR 2001), 2001; 2: 704.

[32] Withagen P, Groen F, Schutte K. EMswitch: A multi -hypothesis approach to EM background modeling. Proc of the IEEE Advanced Concepts for Intelligent Vision Systems (ACIVS 2003), September 2003; 199-206.

[33] Zhang Y, Liang Z, Hou Z, Wang H, Tan M. An adaptive mixture gaussian background model with online background reconstruction and adjustable foreground mergence time for motion segmentation. ICIT 2005, December 2005; 23-27.

[34] Stauffer C, Grimson W. Adaptive background mixture models for real-time tracking. Proc IEEE Conf on Comp Vision and Patt Recog (CVPR 1999) 1999; 246-252.

[35] M. Wang and X. Wang, “Automatic adaptation of a generic pedestrian detector to a specific traffic scene,” in IEEE Conf. on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 2011.

[36] ——, “Transferring a generic pedestrian detector towards specific scenes,” in IEEE Conf. on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012.

[37] A. Bergamo and L. Torresani, “Exploring weakly-labeled web images to improve object classification: a domain adaptation approach,” in Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 2010.

[38] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, “Learning object class detectors from weakly annotated video,” in IEEE Conf. on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012

[39] W. Jiang, E. Zavesky, C. Shih-Fu, and A. Loui, “Cross-domain learning methods for high-level visual concept classification,” in IEEE Int. Conf. on Image Processing, San Diego, CA, USA, 2008.

[40] H. D. III, “Frustratingly easy domain adaptation,” in Meeting of the Association

for Computational Linguistics, Prague, Czech Republic, 2007.

[41] L. Duan, I. Tsang, D. Xu, and S. Maybank, “Domain transfer svm for video concept detection,” in IEEE Conf. on Computer Vision and Pattern Recognition, Miami Beach, FL, USA, 2009.

[42] R. Gopalan, R. Li, and R. Chellappa, “Domain adaptation for object recognition: An unsupervised approach,” in Int. Conf. on Computer Vision, Barcelona, Spain, 2011.

[43] B. Gong, Y. Shi, F. Sha, and K. Grauman, “Geodesic flow kernel for unsupervised domain adaptation,” in IEEE Conf. on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012.

[44] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part based models,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627–1645, 2010.