National Conference on Recent Advances in Computer Science and IT (NCRACIT)
International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 4 | Issue 1 | ISSN : 2456-3307

Subspace-Based Adaptation of Detectors for Video

Sangeeta Bhan[1], Sunil Dalal[2], Pankaj Choudhary[3]

[1]Computer Science and Engineering, Delhi Technological University, Delhi, India
[23]Department of Information Technology BGSB University, Rajouri, J&K, India
sangeetabhan7@gmail.com[1], sunildalal57@gmail.com[2], pschoudhary@bgsbu.ac.in[3]

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. Eigen vectors describe this generated source and target subspaces.

Keywords: Subspace, Detector, Adaptation, Histogram.

I. INTRODUCTION

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 Eigen vectors for both source and target dataset. 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.

II. RELATED WORK AND BACKGROUND THEORY

Over the past several years, many different have been pro-posed 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 and language processing. A classical strategy related to our work consists of learning a new domain-invariant feature representation by a new projection

space. Work on PCA based domain adaptation [1] has been done to minimize the marginal distributions between the different domains. Other strategies have been explored on image datasets as well, such as using metric learning approaches [2,3] or canonical correlation method [4] 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. Boosting-based [5] approaches are used for detection of objects in image or video. 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. Object detection and classification is the area which has received much attention in the research of computer vision and pattern recognition recently.

III. METHODOLOGY

The very first step in novel algorithm for detection 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].

At the time of testing, iteratively learn new detector by the help of detected bounding boxes [Algorithm 3].

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)
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**

Sets. Since a set of all the possible window sized images in the form of patches at different scales and locations from a frame provides a nearly endless supply of negative samples. Training with all the available data is considered impractical.

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 ← XS' XT ⊳ XS' is transpose of XS & M is Transformation Matrix
5: Xa ← XSM ⊳Xa is New Coordinate System
6: trainData ← SX
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**

IV. EXPERIMENTS AND RESULTS

The proposed algorithm was implemented in C++11 with Open CV 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. . 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.

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.

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. Figure 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.

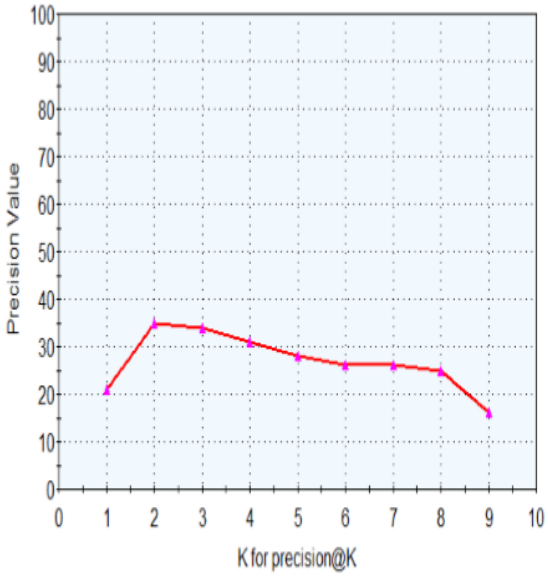

Figure 1.

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 samples.

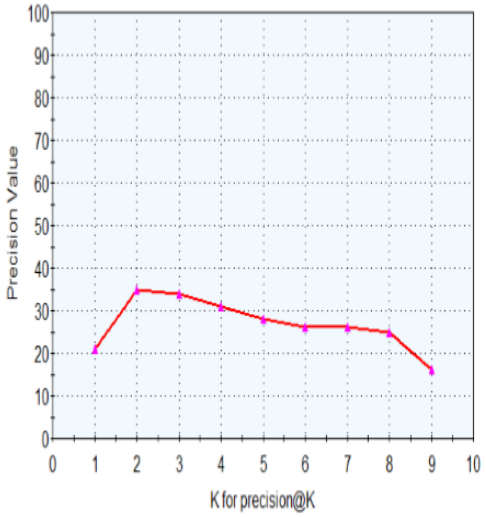

Figure 2.

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 2.

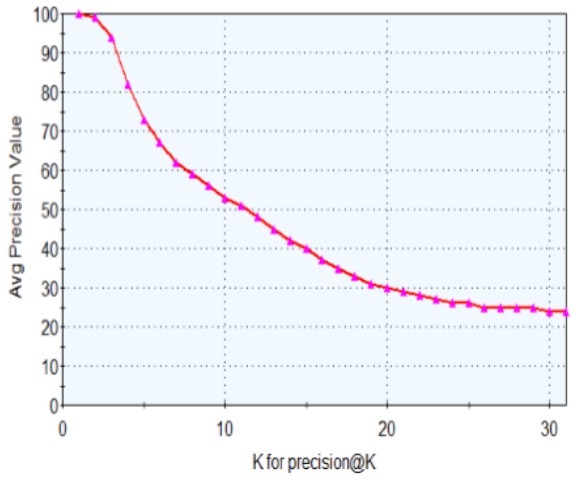

Figure 3.

Given Figure 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.

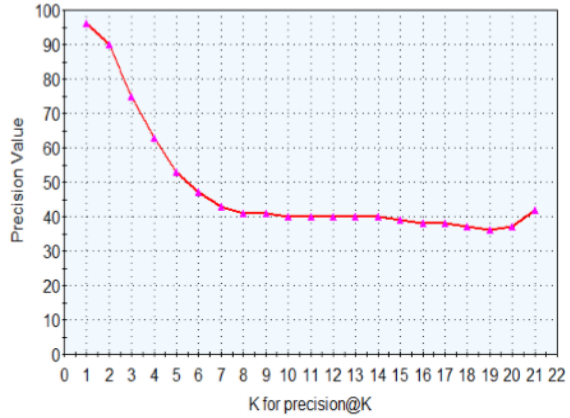

Figure 4.

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 4.

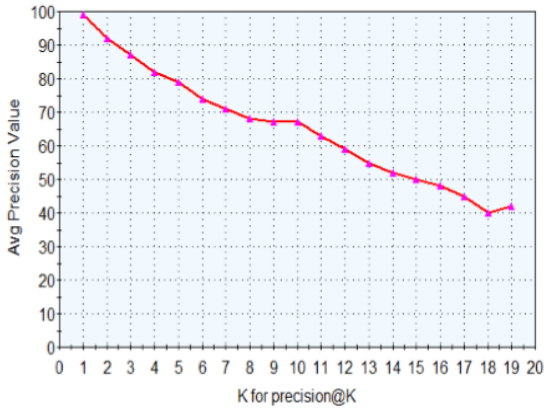

Figure 5.

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.

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 4.6), 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 6.

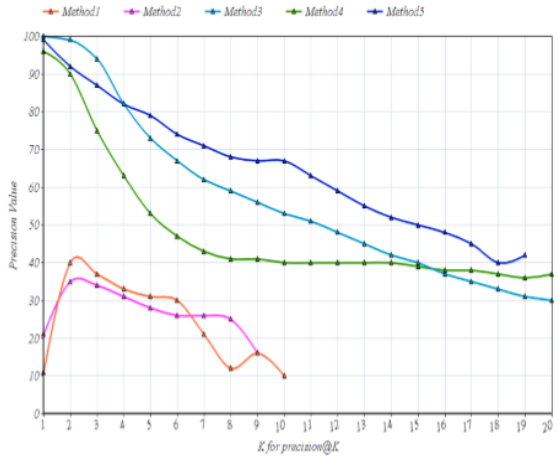

Figure 6.

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 (Method4 and Method3) which gives extremely well results. The bar graph (Figure 7) shows mean precision and recall value for all the methods and both looks fairly well in the Method5.

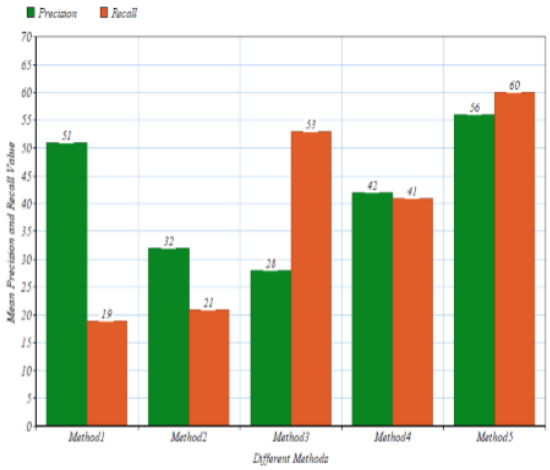

Figure 7.

V.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. 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, and were compared simultaneously for VIRAT video datasets having different domains. 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.

VI.REFERENCES

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

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

[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] D. Foster J. Blitzer and S. Kakade. Domain adaptation with coupled subspaces.

[5] 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.