MultiClass Object Classification in Video Surveillance Systems Experimental Study

Mohamed Elhoseiny, Amr Bakry, Ahmed Elgammal
Department of Computer Science, Rutgers University
110 Frelinghuysen Road, Piscataway, NJ, USA
m.elhoseiny@cs.rutgers.edu, amrbakry@cs.rutgers.edu, elgammal@cs.rutgers.edu

Abstract

There is a growing demand in automated public safety systems for detecting unauthorized vehicle parking, intrusions, un-intended baggage, etc. Object detection and recognition significantly impact these applications. Object detection and recognition are challenging problems in this context, since the purpose of the surveillance videos is to capture a wide landscape of the scene; resulting in small, low-resolution and occluded images for objects. In this paper, we present an experimental study on geometric and appearance features ($\in R^{\approx 25000}$) for outdoor video surveillance systems. We also studied the classification performance under two dimensionality reduction techniques (i.e. PCA and Entropy-Based feature Selection). As a result, we built an experimental framework for an object classification system for surveillance videos with different configurations.

1. Introduction

Object classification is an important building block of surveillance systems that signficantly impacts reliability of its applications (e.g. the public safety application and video indexing/tagging , video semantic search). Outdoor environments are more challenging for object classification, due to the following reasons:(1) uncontrollable environment conditions (e.g. fog, rain, lighting and haze) (2) incomplete appearance details of moving objects due to occlusions, (3) large distance between the camera and the moving objects, (4) very low images resolution, since the moving object occupies a small area ($\approx$ 50 squared pixels) in the video frames. Figure 1 shows an instance run of our experimental framework on a surveillance video to illustrate the aforementioned problems. For these reasons, state-of-the-art approaches (e.g. [8]) for object detection and recognition do not perform well in outdoor surveillance systems.

This paper presents an experimental study that compares different alternatives for the components of the object recognition pipeline. As shown in the block diagram in Figure 2, the object classification process has different phases: Object detection, Feature extraction, Dimensionality reduction, and Classification. The feature extraction phase is a crucial step in object classification. Therefore, investigating different types of features and combining them is one of the important aspects of this study. We implemented two types of features: appearance features and geometric features. Due to the curse of dimensionality and redundancy of the features, dimensionality reduction is used for improving the classification accuracy and reducing the time complexity of the overall process. We have implemented two alternatives of the dimensionality reduction: feature transform and feature selection. Then we evaluated the classification performance based on two classifiers (SVM and AdaBoost). Another important evaluation metric is the time performance. This is crucial in realtime systems and in case this object recognition module is followed by further processing such as activity recognition.

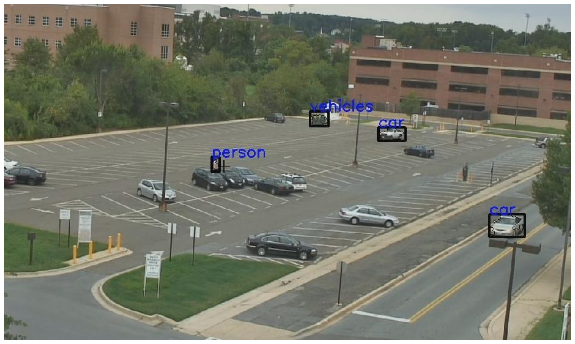

Figure 1. An instance run of our Framework

The contribution of this paper is summarized as follows: Building object detection and recognition testbed with high recognition accuracy in surveillance videos. Comparing the recognition accuracy using appearance features (e.g HOG [4]) and geometric features. Comparing the performance of different dimensionality reduction techniques. Comparing the difference in performance between SVM and AdaBoost classification algorithms in this context.

The rest of this paper is organized as follows. Section 2 presents the related work and the existing literature for object recognition and classification. In section 3, we present our experimental framework. In section 4, we show the experimental results and discussion. Finally, section 5 presents the conclusion and the future work.

2. Related Work

This section presents the related work of object classification in both still images and videos. One of the motivations behind this work is to study the combination of appearance image features and contour-based/geometric features in video surveillance systems.

2.1. Object Classification in still Image

There have been various methods used for object classification in still images. Methods for object classification typically extract features by applying interest point detectors on images. The survey by Schmid et al. [17] evaluated the repeatability rate and information content of various interest-point detectors. They also compared contour-based, intensity-based and parametric model-based methods. The conclusion was that the Harris point detector [18] and its multi-scale variation perform the best in two aspects: the repeatability and the information content. Matas et al. [19] proposed detection algorithm using maximally stable extremal regions (MSER), integrated with the SIFT descriptor in [20] as a key point detector. The difference of Gaussian (DoG) has been widely used as a keypoint detector with SIFT. The experiments in [20] show that the such interest points are detected no matter if it belongs to objects or noisy background. Some other methods for scene categorization [22] used a regular grid on images to extract features from rectangle patches. In this systems, salient regions are detected in the image. However, Some of them lie on the background or cluttered. The successful usage of these points after detection highly depends on descriptors and classification.

2.2. Object Classification in Videos

Object classification in videos has been majorly addressed with silhouette features, namely shape-based classification. This type of classification is commonly used for surveillance systems, generally, or action recognition specially. Dedeoğlu, Yiğithan et al presented an approach in [10] that is able to classify objects as human, human group, and vehicle; based on a silhouette template database. Distance function is measured between the query silhouette to be classified and the database. Jianpeng Zhou et al presented a human classification algorithm based on codebook learning named DSCL (distortion sensitive competitive learning) [11] as a part of a human tracking system. Similar methods have been used to categorize and classify postures of the same object[12], where posture of a human is classified using Support Vector Machine (SVM) with affine invariant Fourier descriptor. The descriptor is built on the human contour that corresponds to the posture. The same idea was used to determine posture of hand or sign language alphabet (e.g. [13, 14]). Another direction that uses various motion features is the work by Yehezkel and Boaz [15]. Some of the geometric features used in this study follow from this work.

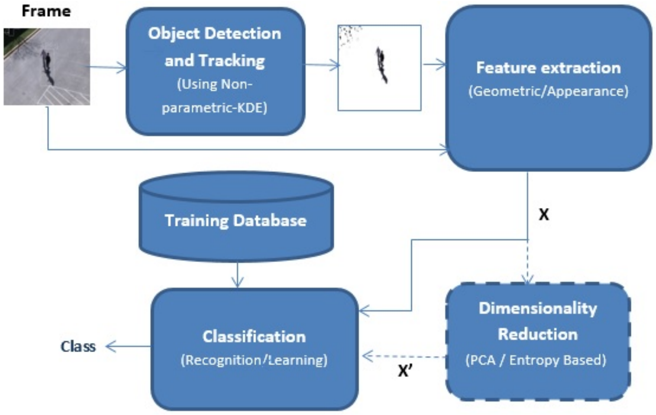
Frame
Object Detection and Tracking (Using Non-parametric-KDE)
Feature extraction (Geometric/Appearance)
Training Database
X
Classification (Recognition/Learning)
Class
X'
Dimensionality Reduction (PCA / Entropy Based)

Figure 2. Experimental Framework

3. Proposed Framework

In this section, we present the different phases of the proposed testbed, illustrated in Figure 2. Briefly, the moving objects are detected and segmented from the input video through a motion detection module. Afterwards, information of the object detection (*e.g*. bounding boxes, contour of extracted objects, current frame, and binary frame) is relayed to a feature extraction module that extracts the objects' features (geometric and appearance features). The third step is the dimensionality reduction phase that produces new feature vector X'; this phase is optional. Finally, object classification phase learns object categories or classifies the objects depending on the mode of the framework (Learning or Classification). Each of the following subsections details the framework components.

3.1. Object Detection and Background Subtraction

In surveillance systems, the main goal is to detect motion and monitor activities. Therefore, the first step, in most of these systems, is to segment moving objects. We can categorize segmentation algorithms into background subtraction (*e.g*. [23]), dense motion segmentation (*e.g*. [24]), video segmentation (*e.g*. [26]), and specific object detector (*e.g*. [25]). For stationary camera, as in surveillance system, the background subtraction is the most effective approach (high accuracy and low false alarm). In this framework, we adopt a background subtraction approach [23] for detecting moving objects and segmenting them out. This approach uses non-parametric Kernel Density Estimation (KDE) for building the background model. Based on this model, each pixel is classified independently into background or foreground. The result of this process is a bunch of connected foreground areas. These foreground areas are further segmented into color homogeneous subareas (Blobs). The color distribution and geometry for these blobs is used for tracking.

3.2. Feature Extraction

As aforementioned, one main task of this work is to study the effectiveness of several feature descriptors and compare the performance of different machine learning algorithms on the extracted features. This stages takes as input the detected foreground regions. Then the surrounding contours and oriented bounding boxes are computed for each region through the eigen vectors of the point set. The remaining of this subsection describes the features utilized in our study.

HOG Features

Histogram of Oriented Gradients (HOG) [4] is a widely used feature descriptor in computer vision for the purpose of object detection and classification. This technique builds histogram of discrete values of gradient orientations in localized portions of the detected object.

Luminance Symmetry

In [15], Luminance Symmetry feature was proposed to measure the brightness symmetry of an object. In our study, we computed the Luminance Symmetry around the axis using the oriented bounding box as follows

$$L_{sym} = \frac{1}{C}\frac{2}{w}\sqrt{\sum_{i=1}^{h}(\sum_{j=1}^{w/2} I(i,j).B(i,j) - \sum_{j=w/2+1}^{w} I(i,j).B(i,j))^2} \tag{1}$$

where I is the intensity image, B is the mask image (i.e. $B(i,j) = 1$ if location (i,j) is a foreground pixel), h and w are the size of the oriented bounding box of the object segmented from Background subtraction, and C is the maximal luminance level. Asymmetric objects (e.g. clutter, body organ, bikes) will have smaller luminance symmetry compared to symmetric objects (e.g. cars, humans).

Central Moments

Hu moments [5] are well known scale, translation and rotation, invariant moments. We extracted the seven Hu moments on the ADI object's image, where ADI object's image is the absolute difference image after background subtraction (before thresholding), constrained to the oriented bounding box of the object.

ART Moments

Angular radial transform (ART) is an image descriptor adopted in MPEG7 [6]. It is advantageous in capturing both connected and unconnected regions in a compact way. We extracted ART descriptors with the standard configuration $nAngle = 12, nRadius = 6$ which gives in total $6*12-1=$ 72 features.

Cumulants

Following [15], three textural properties were computed on the object after applying the foreground mask on it: (1) Mean value ($E\big[X\big]$) of the intensity , (2) Standard deviation ($E\big[(X - \mu)^2\big]$) of the intensity histogram, (3) Skewness ($\frac{E\big[(X-\mu)^3\big]}{E\big[(X-\mu)^2\big])^{3/2}}$) of the intensity histogram. It was shown in [15] that the mean is mostly low for clutter compared to objects like bags (most bags have high contrast with the background), while the standard deviation is high for clutter (because intensity is often non-homogeneous) and low for bags (because intensity is homogenous). The skewness is negative for bags and positive for the remaining classes in VIRAT dataset.

Horizontal and Vertical Projection

Horizontal $HP_{i,+}$ (or vertical $VP_{+,j}$) projection is a histogram in which each bin, corresponds to the sum of the pixels in row i (or column j), where $HP_{i,+} = \sum_i B^{'}(i,j)$,$VP_{+,j} = \sum_j B^{'}(i,j)$. This feature captures histogram variation, which can discriminate between many objects with low resolution.

Morphological Features

Similar to [15], we extract four morphological features, as follows: (1) Anthropometry ($A_{th} = \frac{H}{P}$), which is a stable ratio for human body. (2) Compactness ($Cmpct = \frac{Ar}{P^2}$), which measures complexity of the shape, (3) Aspect ratio ($AR = \frac{W}{H}$) , (4) Solidity ($SD = \frac{Ar}{Ar_{CH}}$), which measures the portion of concave parts in the shapes. Where H, W are the width and height of the bounding box of the object respectively, P is the perimeter of the object's contour, Ar is the contour area of the object, and Ar_{CH} is area of the convex Hull containing the object.

3.3. Dimensionality Reduction

Dimensionality reduction techniques are used for increasing the robustness of data analysis. There are two main categories for dimensionality reduction: feature transform (supervised/unsupervised) and feature selection. In feature selection techniques, the dimensions of the new low-dimensional space is a subset of the dimensions of the original high-dimensional space, while in feature transform, the dimensions of the new space is a linearly or nonlinearly transformation of the dimensions of the original space. In our framework, we use PCA, as an unsupervised feature transform method, for projecting the points in the high-dimensional feature space into a low-dimensional space. We used Entropy Based Discretization (EBD) [16, 15] as a feature selection method.We compared the results of three different configurations: feature transform, feature selection and the original extracted features.

3.4. Extracting Training Samples

For training purposes, we extracted the features described in Section 3.2 from VIRAT dataset [1, 2]. We automatically chose frames for five classes {Human, Car, Vehicle, Object, Bicycle}. The class "Vehicle" is any moving vehicle other than cars, such as van and truck, while the class "Object" is anything man can carry like boxes and backpacks. We aim to extract training samples, such that the classes are fairly balanced and to increase the features extraction accuracy. For doing that we impose four constraints to select suitable frames from the dataset videos for training. The four constrains were imposed during training, while only the first three are imposed on the test videos.

1. **Detection Percentage:** ($Dp > 30\%$) Detection percentage of an object in a specific frame is the percentage ratio between the contour area (C) of the object (resulting from background subtraction) and the bounding box area (BB)of the detected object ($Dp_i = 100 * \frac{C_{area_i}}{BB_{area_i}}$).
2. **Overlapping Percentage:** ($OP < 10\%$) Overlapping percentage of an object is defined as follows. $OP_i = 100 * \frac{\sum_{i \neq j} \cap(BB_i, BB_j)_{area}}{BB_{i_{area}}}$, where BB_k is the bounding box of object segment k in the current frame and $\cap(BB_i, BB_j)_{area}$ is the area of intersection between bounding box i and bounding box j.
3. **Motion Constraints:** We only obtain training samples from objects whenever the distance that the object has moved is greater than a threshold th. (th is 5 pixels in this framework)
4. **Object Instance Constraint:** This constraint is applied while extracting the data for learning phase only to ensure the inclusion of various objects in training. Extracted training samples for a given object instance is limited to at most 10 feature vectors.

3.5. Object Classification

For training, we have a list of pairs $\{(x_i, y_i)\}_{i=1}^{N}$ where $x_i \in \Re^d$ is the feature vector, and $y_i \in \{1, 2, ..., K\}$ is the sample label. Let $X = [x_1, x_2, ..., x_N]^t$ is $N \times d$ matrix and $Y = [y_1, y_2, ..., y_N]^t$ is N dimensional column vector. For classifying test feature vector, we used two different classification techniques *SVM* and *AdaBoost*, and we compare their results. We used C-SVM for multi-object classification [7]. AdaBoost [3, 9] technique is based on combining the results of many weak classifiers to get a single more powerful classifier. In our case, we used AdaBoost of stump (one-level binary tree). We trained one stump for every class $k \in \{1, 2, ..., K\}$.

4. Experimental Results

This section presents the experimental results of our study. We evaluated on surveillance videos from VIRAT [1] dataset. We performed the following object classification experiments: (1) Appearance-based classification based on HOG features, (2) PCA based SVM classification, (3) Feature selection based SVM classification, (4) Feature selection based AdaBoost classification. HOG features only were used in the first experiment to study appearance features, while the full set of features were used in the remaining three experiments.

4.1. Appearance Features (HOG)

In this experiment, we used the selection constraints[1], detailed in subsection 3.4 on VIRAT dataset [1] for building an image dataset with positive and negative examples for each object. Figure 3 shows samples of generated dataset for training. We used the settings in Table 1 for multi-object classification using HOG features. We used C-SVM with five-fold cross-validation using $80\% - 20\%$ training-test split. 71.4% accuracy was achieved.

	Win Size	block size	cell size,strid	Bin Size
Human	64 x 128	16×16	8×8	9
Car	104 x 56	16×16	8×8	18
Vehicle	120 x 80	16×16	8×8	18
Bike	104 x 64	16×16	8×8	9
Object	64 x 64	16×16	8×8	18

Table 1. HOG Settings

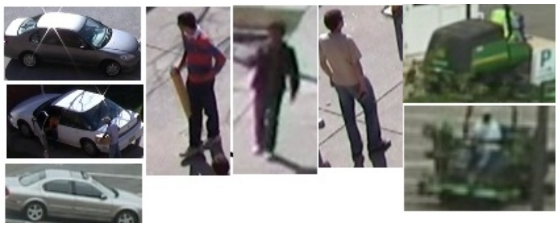

Figure 3. Example of extracted training images

4.2. PCA based SVM Classification

In this experiment, we project the full set of features into a 30-Dimensional principal subspace. The dataset was split into two subsets (80% training and validation,

[1] sample extracted image frames, https://www.dropbox.com/sh/hiv9xu4tm365rjg/rd9MGhKN-k/HOGtrainingIm

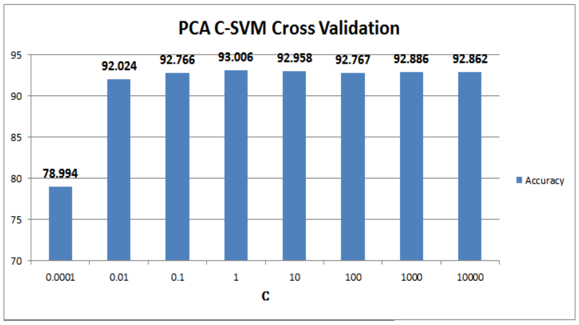

Figure 4. PCA based SVM (80% training percentage)

20% testing). Figure 4 shows 5-fold cross-validation using C-SVM with different values of $C = 10^e$, where $e \in \{-4, -2, -1, 0, 1, 2, 3, 4\}$. Figure 4 shows that the best value $C = 10^3$, which is used to test on the remaining 20% of the data resulting in test accuracy of 89.9%. However, there is computational drawback of using PCA. Since the original full length 25,000-D feature vector has to be computed, then projected to the principal low-dimensional subspace. Both tasks are computationally intensive which might violate the real time requirements of surveillance systems.

4.3. Feature Selection based SVM Classification

The objective of feature selection is three-fold:(1) improving the prediction performance of the predictors, (2) providing faster and more cost-effective predictors, and (3) providing a better understanding of the underlying process that generated the data. We performed an entropy based discretization approach on the features. An observation for the computed entropy is that appearance features have little entropy and hence we performed the following experiment in which we have to compute only 142 features out of more than 25,000 features. Table 2 shows 5-fold cross validation of C-SVM performance on (90% − 10% split). The best recognition rate is (92.3%) in the test data and it was achieved by setting $C = 100$. We computed also the recognition accuracy at lower training percentage (e.g 80%−20% split, 60% − 40% split). The recorded test accuracy are 91.5% and 87.5% for 80% − 20% and 60% − 40%, respectively as illustrated in Table 3.

C	60% − 40%	80% − 20%	90% − 10%
0.0001	89.840	92.748	93.010
0.01	89.361	92.125	92.890
0.1	89.297	92.176	92.610
1	89.616	92.046	92.860
10	89.435	92.129	92.907
100	89.552	92.129	92.907
1000	89.552	92.129	92.907
10000	89.552	92.129	92.907

Table 2. Feature Selection based SVM: Validation Accuracy %

Split	Test Accuracy %
60% − 40%	87.5
80% − 20%	91.5
90% − 10%	92.3

Table 3. Feature Selection based SVM: Test Accuracy %

4.4. Feature Selection based AdaBoost Classification

As mentioned in Section 3.5, we have trained AdaBoost of stumps. We used 5-fold cross-validation for training the classifier. Table 4 shows the results for different values of weight trim rate $\rho \in 0.1, 0.3, 0.5, 0.7, 0.9, 0.95, 1.0$. All rows except the last one shows the average validation error for the five folds. The last row shows the test accuracy for different category.

ρ	Human (3942)	Car (4197)	Vehicle (286)	Objects (1188)	Bicycle (108)
0.10	59.433	43.191	02.933	53.066	01.098
0.30	15.175	27.780	02.933	53.066	01.098
0.50	15.175	27.780	02.933	53.066	60.375
0.70	66.784	69.095	61.597	87.788	99.298
0.90	99.331	99.623	100.00	87.788	100.00
0.95	99.228	99.452	100.00	87.788	100.00
1.00	99.125	97.805	100.00	99.794	99.863
T. Acc	98.998	99.512	100.00	99.794	100.00

Table 4. AdaBoost: Cross-Validation Accuracy (%) against ρ, and Test Accuracy(T.Acc %)

From Table 4, we can notice many points. Choosing $\rho \in [0.90, 0.95]$ works good with almost all classes. Recall, ρ is percentage of samples that are used in training the next stage's weak classifier. So this finding means that the algorithm tends to keep almost all sample for learning the upcoming weak classifier.

We also get the following findings: first as we increase number of weak classifiers as we get better results. Table 5 shows affect of changing the weak classifiers count for the training and test accuracy for every category. However, this significantly increases the processing time. Therefore, choosing the number of weak classifier to be close to the dimensionality of the features vector ($W \approx K$) gives good enough results.

W	Human (3942)	Car (4197)	Vehicle (286)	Objects (1188)	Bicycle (108)
25	95.868	95.251	99.606	99.743	99.966
100	98.473	98.371	100.00	99.794	100.00
150	98.988	99.040	100.00	99.794	100.00
200	99.331	99.452	100.00	99.7941	100.00
250	99.571	99.777	100.00	99.794	100.00
300	99.880	99.897	100.00	99.794	100.00
T. Acc	99.589	99.872	100.00	99.794	100.00

Table 5. AdaBoost: Cross-Validation Accuracy (%) against W, and Test Accuracy (T.Acc %)

On the other hand, in Table 5, we can see that for the *balanced classes*[1] (Human and Car), we need large number of stumps. Though, for unbalanced classes (Vehicle, Objects and Bicycle), small number of stumps is enough for getting the maximum accuracy.

For multi-label classification. We combined the results of all binary classifiers for producing the multi-label classifier, we get a final accuracy=95.0782%.

4.5. Discussion

Table 6 summarizes the performance of the four experiments on the $80\% - 20\%$ training-test split. One conclusion from the experiments is that appearance based features (HOG in our case) did not perform well in our context. The intuition behind that is the low resolution of the detected object as apparent in Figure 3. Another important conclusion is that geometric features performs significantly better in surveillance systems.

HOG-SVM	PCA-SVM	FSel-SVM	FSel-Adaboost
71.4%	89.9%	91.5%	95.08%

Table 6. Comparison on $80\%-20\%$ trainig-test split (FSel denotes feature selection)

5. Conclusion and future work

Due to the small size and low resolution of the objects in Surveillance Systems, the experiments shows that using appearance features like HOG features is less discriminative for recognizing object classes. Yet, If it is combined with geometric features, this leads to high recognition accuracy. We extracted many geometric features (e.g. luminance symmetry, central moments, ART moments) in addition to the HOG features for each class setting. After applying the feature selection, this combination of features is proved empirically to be effective for object recognition. Entropy based feature selection outputs very small dimensions in HOG (<10 out of 25,000). This indicates the that geometric features are more dominant in surveillance settings. Finally, SVM and AdaBoost classification techniques performed well for recognizing objects. Yet, AdaBoost performed better than SVM. Feature selection has computational advantage in the recognition time, as only the selected features have to be computed. In contrast, if PCA is used, all the features have to be computed. As a future work, we plan to apply similar study in more challenging surveillance datasets.

References

[1] Sangmin Oh et al, “A Large-scale Benchmark Dataset for Event Recognition in Surveillance Video”, CVPR 2011.

[2] VIRAT Dataset http://www.viratdata.org/

[3] Friedman, J. H., Hastie, T. and Tibshirani, R. “Additive Logistic Regression: a Statistical View of Boosting”. Technical Report, Dept. of Statistics, Stanford University, 1998.

[4] Navneet Dalal , Bill Triggs. “Histograms of Oriented Gradients for Human Detection”. CVPR, 2005.

[5] Hu, M.K, “Visual pattern recognition by moment invariants”. IRE Trans. Inform. Theory IT-8, 179–187,1962.

[6] Manjunath, B., Salembier, P., Sikora, T. “Introduction to MPEG-7: Multimedia Content Description Interface”. Wiley & Sons, 2002.

[7] Fan, Rong-En and Lin, C. J, “A Study on Threshold Selection for Multi-label Classification”, National Taiwan University, 2007

[8] P. Felzenszwalb, R. Girshick, D. McAllester, “Cascade Object Detection with Deformable Part Models”, CVPR, 2010.

[9] Christopher M. Bishop, “Pattern Recognition And Machine Learning, Information Science and Statistics”, Springer, 2006

[10] Y. Dedeoğlu, ”Moving Object Detection,Tracking and Classification for Smart Video Surveillance (MSc Thesis)”, bilkent university, 2004.

[11] J. H. Jianpeng Zhou,” Real Time Robust Human Detection and Tracking System,” CVPR, 2005.

[12] V. Kellokumpu, M. Pietikinen, and J. Heikkil,” Human Activity Recognition Using Sequences of Postures,” MVA , 2005.

[13] M. Zahedi,” Robust Appearance-based Sign Language Recognition” , Aachen University, 2007.

[14] R. Akmeliawatil, M. P.-L. Ooi, and Y. Chow Ku,” Real-Time Malaysian Sign Language Translation using Colour Segmentation and Neural Network,” Instrumentation and Measurement Technology Conference , 2007.

[15] Raanan Yehezkel, Boaz Lachover, “Multiclass object classification for real time video surveillance systems”, Pattern Recognition Letters, 2011.

[16] Bose, B., Grimson, E.” Improving object classication in far-eld video”, CVPR,2004.

[17] Schmid C., Mohr R., Bauckhage C.: “Evaluation of interest point detectors.” IJCV, 2000

[18] Harris C., Stephens M.: “A combined corner and edge detector”. Alvey Vision Conference, 1988

[19] Matas J., Chum O., Urban M., Pajdla T.: ”Robust wide-baseline stereo from maximally stable extremal regions.” Image and Vision Computing , 2002.

[20] Lowe D. G.: “Object recognition from local scale-invariant features”. ICCV,1999

[21] Vogel J., Schiele B.: “A semantic typicality measure for natural scene categorization”., DAGM-Symposium,2004

[22] Li, Fie-Fie., Perona P.: “A bayesian hierarchical model for learning natural scene categories,” CVPR, 2005

[23] Elgammal, Duraiswami, R., Harwood, D., Davis, L.S.: ”Background and foreground modeling using nonparametric kernel density estimation for visual surveillance.” Proceedings of the IEEE,2002

[24] Brox, T.; and Malik, J. : ”Object segmentation by long term analysis of point trajectories.”,ECCV,2010.

[25] Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.: ”Cascade object detection with deformable part models,” CVPR, 2010

[26] Grundmann, M.; Kwatra, V.; Mei Han; Essa, I.: ”Efficient hierarchical graph-based video segmentation,” CVPR, 2010

[1] We mean by balanced that for *One-vs-All* setting, number of samples representing the positive class and number of samples representing the negative class are almost equal.