**HAOHAN LI**

**1003093092**

**HANWEN LIANG**

**1003868291**

**CSC2515 Introduction to Machine Learning  
  
Project Final Report**

**Abnormal Behavior Detection**

December 1, 2017

**Abstract**

The goal of this project is to establish a video system that can distinguish abnormal behaving people from normal ones. We focused on a privacy-preserving video system and obtained the internal information of video with image processing techniques. Using optical flow features, we estimated and segmented inhomogeneous crowds composed of pedestrians that travel in different directions. With these features, we applied several classifying algorithms to tell the normal behaviour from the abnormal. In this process, we used grid search method to find the best hyper-parameter. We validated both the crowd segmentation algorithm, and the crowd counting system, on a large pedestrian dataset (200 frames of video, containing 4,988 total pedestrian instances). This project could help monitoring abnormal behaved objects in public environment with vision information.

Key words: image processing, optical flow, classification algorithms

GitHub Link: <https://github.com/davidhaohanli/csc411proj.git>

**Background**

There is currently a great interest in vision technology for monitoring all types of environments. One of the most important goals of monitoring environments is to find abnormal behaviours, which requires the identification and understanding of human behaviour. The existing technology for behaviour recognition and understanding can be divided into visual-based methods and non-visual methods based on ways of information collection. With the development of computer vision technology, vision-based human behaviour recognition and identification has attracted increasing attention. Its application to intelligent control, medical diagnosis, perception interface, human-computer interaction, motion analysis has inspired wide interest among many researchers.

Yet, the deployment of vision technology is invariably met with skepticism by society at large, given the perception that it could be used to infringe on an individual’s privacy rights. In response to the skepticism, Antoni B. Chan proposed a new formulation for surveillance technology by detecting people in the given environment and tracking over time, which is different from individual tracking. However, he did not put forward any practical methods for distinguishing the abnormal behaviours from the normal ones. Inspired by his idea, and combined with machine learning methods, our project applied this technique for solving the detection goal of detection of abnormal behaviours.

In addition, an amount of noise could be added to images in the process of formation, transmission, reception. The noise hinders human visual perception and prevents the system sensors from correctly understanding or analysing the image information, resulting in deviations between the actual value and the detected value. Hence, it is necessary to use appropriate methods to reduce noise as pre-processing before further application of image fusion, image reconstruction, image enhancement, image segmentation, feature extraction and analysis.

**Literature Review**

The abnormal behaviour detection consists of three paradigms: 1) crowd detection, 2) visual feature extraction and clustering, 3) feature-based regression. Crowd detection algorithms are based on boosting appearance and motion features [1]. Because there exists significant occupation among the crowd, the detection algorithms may perform badly, which can be partly solved by part-based detectors [2,3]. The second paradigm plays a fundamental role by identifying and tracking visual features over time. Based on features clustering, normal behaviours are gathered. Examples of this formulation include [4], which uses the KLT tracker and agglomerative clustering, and [5], which takes an unsupervised Bayesian approach. These methods typically work by: 1) subtracting the background; 2) measuring various features of the foreground pixels, such as total area [6,7,8], edge count [6,7,8], or texture [9].

Regarding clustering algorithms, they attempt to classify elements into categories based on their similarity. Several different clustering strategies have been proposed [10]. In K-means, clusters are generated by data characterized by a small distance to the cluster centre. An objective function, typically the sum of the distance to a set of putative clusters centres, is optimized [11] until the best cluster centres candidates are found. In distribution-based algorithms, one attempts to reproduce the observed realization of data points as a mix of predefined probability distribution functions; the accuracy of such methods depends on the capability of the trial probability to represent the data.

**Objectives**

This project aims to detect and mark abnormal behaved people in crowded public environment, such as runners (fast motion) among walking pedestrians (slow motion) in a park. We try to generate a robust algorithm that, given a video clip, could effectively segment each person among the crowd and extract features, thus spotting the abnormal behaved persons and tracking these persons with highlighted mark. This project should employ multiple classifiers and find the optimal one with certain evaluation metrics.

**Project Overview**

The project contains 4 phases. In the data collection phase, a video sample from database of UCSD was obtained as the training and testing samples. It is a 200-frame (20 seconds) video collected from a stationary digital camcorder overlooking a park with pedestrian walkway at UCSD. The original video was captured at 30fps with a frame size of 740×480, and was later down-sampled to 238 × 158 and 10 fps [1]. In pre-processing and feature extraction phase, we applied Canny edge detection, Hough transformation, morphological operation and optical flow dense computation to detect and segment the moving objects from each frame, and then extracted and normalized their optical flow features. Next, in the training phase, three different models – logistic regression, KNN and SVM were trained based on the features extracted in previous phase. We used grid search to obtain the optimal hyper-parameters. Last, in the testing phase, we used the rest of video frames as test data, computed the testing accuracy and plot the ROC curve of each model. The best model that should be used to develop the program is determined from the largest area under the curve (AUC) of ROC.

The program is coded in Python with 3 dependencies

- OpenCV and Scikit-Image for image pre-processing and feature extraction

- Scikit-Learn for:

- Logistic regression, KNN, SVM modelling

- Hyper-parameters tuning

- ROC plot and AUC calculation.

**Methodologies**

1. Data collection

1.1 Downloaded Pedestrian Traffic Database from Statistic Visual Computing Lab of UCSD

1.2 Clipped the first 200 frames (20 seconds), the first 97 of which only have normal behaviours and the rest of which have both normal and abnormal behaviours

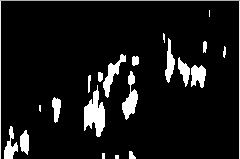
1.3 Down-sampled Images to 238\*158 and 10 fps.

1. Image preprocessing and Feature extraction

2.1 Foreground segmentation

2.1.1 Apply morphological filter on the foreground images

The project applied cv2.BackgroundSubtractorMOG2 for foreground extraction (figure 1). It could be observed that the foreground image has amount of noise. Therefore, the project applied open (erosion→dilation) and close (dilation→erosion) operation to break the linkage between groups of people and clear the in-group false negative point (figure 2).

*Figure 2*

*Figure 1*

2.1.2 Segment each moving object on the foreground images

After foreground extraction, the project applied skimage.measure on the images to detect connected areas, which were labeled by different number. However, people adjacent to each other would be labeled in the same group (in figure 3 with arrow). The project gridded each enormously large connected group according to multiples to normal people height and width (normalized) and re-tagged each split small area positive, if the fulfillment (the ratio of true area to the area of rectangle which surrounds it) of foreground positive points in the area exceeds the threshold value (figure 4).

*Figure 4*

*Figure 3*

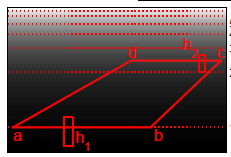
2.2 Computation of weight matrix for normalization

2.2.1 Apply Gaussian blur on the images to smoothen and reduce the noise

2.2.2 Use Canny edge detector to generate the edge-only-images

2.2.3 Use Hough transformation to find lines in edge-only-images

Due to perspective angle of camera, objects would be downsized with longer distance to camera optical centre line. To scale the actual size of objects, a certain matrix is required. The first step applied a pipeline (Gaussian blur→Canny edge detector→Hough transformation) to find the road shoulder edges (figure 5). Since road is presumed to have equal width in this video, it could be used as reference.

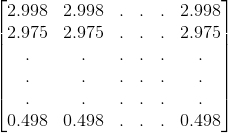
 

*Figure 6*

*Figure 5*

2.2.4 Compute the weight matrix for each row referred to the equal-length of road width (ab, cd) and same height of people (h1 and h2). The project only extracted 2 samples (figure 6) from a single frame and used linear insertion fit

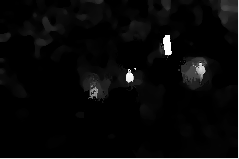
to generate the total weight matrix.



2.3 Feature extraction and labelling

2.3.1 Obtain and normalize the optical flow dense

The project aimed to distinguish abnormal objects by the motion, thus it chose optical flow as major features. It applied cv2.calcOpticalFlowFarneback, which implemented the algorithm: two-frame motion estimation based on polynomial expansion (Farneback,2003) to obtain every pixel’s vertical and horizontal optical flow dense with couple of frames in given sequence. Figure 7&8 demonstrated the optical flow (u and v) of frame 105 after normalized by weight matrix from 2.2.

*Figure 8*

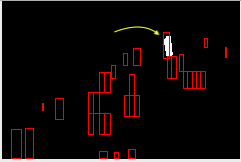
*Figure 7*

2.3.2 Compute the mean of optical flow values to obtain 2-dim features of each object

Applied foreground segmentation from 2.1, the optical flow features were obtained for each rectangular area of person. However, due to size variation of detected rectangles, the raw features vary in dimensionality. The project only computed the mean value of u (horizontal) and v (vertical) as features.

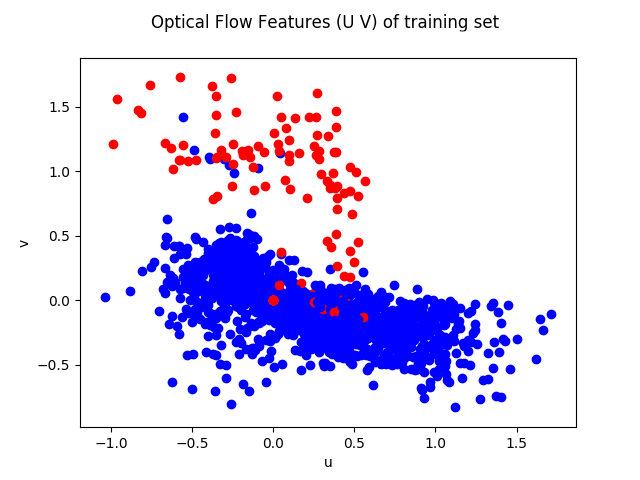
2.3.3 Labelling data

The project obtained abnormal-object-only frames on training set, which were generated by manual drawing on original images. The objects whose fulfilment exceeds certain threshold value of pixel dense would be labelled 1 (arrow on figure 9) and the rest 0.



*Figure 9*

The final U & V distribution demonstrated as follows.



3 Model training and testing

We chose three machine learning methods: logistic regression, KNN, and SVM. In each method, we considered objects from 80th to 140th frames as training samples and objects from 140th to 199th frames as testing samples. For the best hyper-parameters, we used GridSearchCV method. Given an estimator, we did exhaustive search over specified parameter values, which are optimized by cross-validated grid-search over the parameter grid. This method would choose the best one with highest score, which we chose the “accuracy” of 10-fold cross-validation for evaluation. After exhaustive searching, it returns best-estimator along with optimized hyper-parameters. In the end, we computed testing accuracy and marked the abnormal behaviours (predicted as label 1) in all the frames.

3.1.1 Logistic Regression

The parameter set given to GridSearchCV involved inverse of regularization strength. A smaller value specifies stronger regularization.

Training state result:

The best hyper-parameter for -- LogisticRegression is {'C': 1.5000000000000002}, the corresponding mean accuracy through 10-Fold test is 0.9754716981132076

LogisticRegression train accuracy = 0.9754716981132076

3.1.2 KNN

The parameter set given to GridSearchCV involved k which ranged from 1-100.

Training state result:

The best hyper-parameter for -- KNN is {'n\_neighbors': 9}, the corresponding mean accuracy through 10-Fold test is 0.9816037735849057

KNN train accuracy = 0.9853773584905661

3.1.3 SVM

The essential goal of SVM is to maximize distance between projected class means and minimize projected class variance. The parameter set given to GridSearchCV involved linear kernel and penalty parameter, which ranged from 0.01 to 2.01.

Training state result:

The best hyper-parameter for -- SVM is {'C': 1.01}, the corresponding mean accuracy through 10-Fold test is 0.9759433962264151

SVM train accuracy = 0.9759433962264151

3.2 Model analysis and comparison

We used objects from 140th to 199th for test dataset and computed the accuracy for each model. For each model, in a Receiver Operating Characteristic (ROC) curve the true positive rate (Sensitivity) is plotted in function of the false positive rate (100-Specificity) for different cut-off points. Since our dataset has much more positive samples than negative samples, and our goal is to pick out negative samples from them, the AUC is more persuasive than accuracy. Therefore, the best model is determined from the largest area under the curve of ROC (AUC).

**Results**

With proper implementation of the image pre-processing, feature extraction, and classification algorithm, the project obtained the re-generated sequential frames and a video, in which the abnormal behaved people (predicted by classifiers) is tagged by red rectangle. The following images demonstrate 3 clipped frames from the video.



*Figure 12*

*Figure 11*

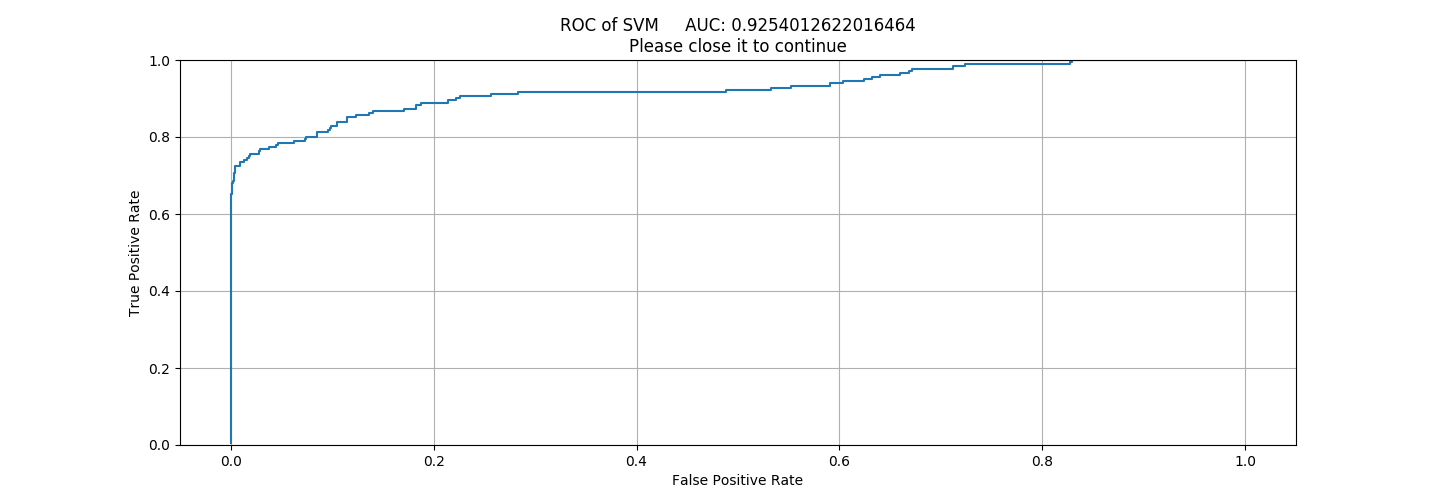
*Figure 10*

From testing frames 140-199, 3 classifiers generate different values on the metrics:

SVM:

SVM test accuracy = 0.9675396532644781

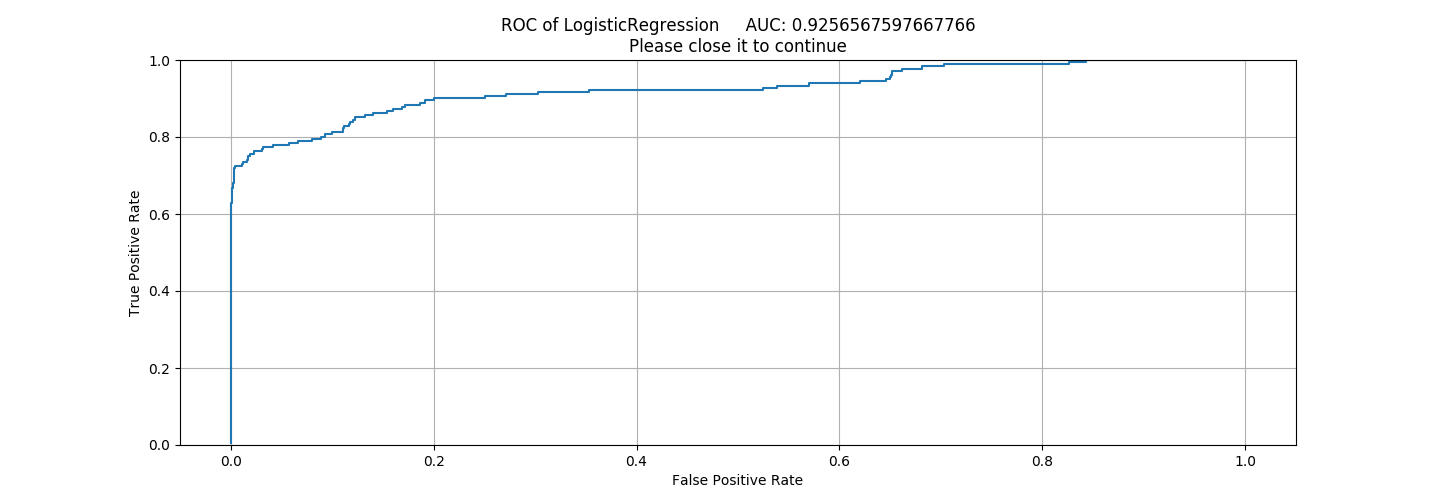
Classifier SVM area under curve of ROC is 0.9254012622016464



Logistic Regression:

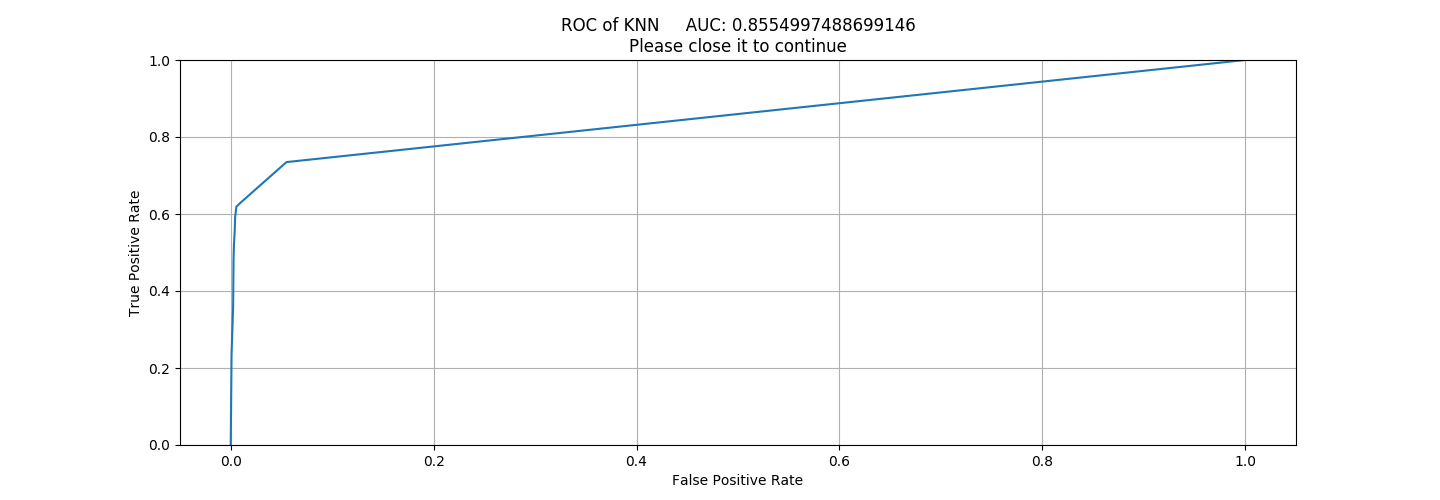
LogisticRegression test accuracy = 0.9645887126521578

Classifier LogisticRegression area under curve of ROC is 0.9256567597667766



KNN:  
KNN test accuracy = 0.966801918111398

Classifier KNN area under curve of ROC is 0.8554997488699146



**Conclusion and Discussion**

From the video generated and metrics for classifiers, we could conclude that the objectives are achieved. All 3 classifiers successfully predicted the abnormally behaved person, and tagged it in the frames. The test accuracies are significantly high, which results in all above 0.9.

By comparison, we may conclude that the SVM classifier performs best because it holds the largest accuracy value, and the smoothest plot of the ROC. It implies that SVM could separate the walking pedestrian and abnormal object at the greatest level without overfitting. The conclusion meets our presumption since SVM is robust for small training sample size. In our case, the positive data in ground truth (abnormal behavior) is dramatic less than negative data. SVM only considers the support vector points, therefore, it is the most suitable classifier for this case.

**Future Development**

1. In the pre-processing and feature extraction phase, we did morphological operation on foreground images. There still exists some occupation between adjacent persons, resulting in a wrongly segmented target. We would do more work to segment almost perfectly.

2. In the training phase, due to distinguishing features of positive and negative samples, the algorithms performed well. Yet considering those positive samples which mixed in negative ones, we would try using clustering method to solve this problem.

3. In our project, we considered runners among pedestrian as abnormally behaved persons. We would collect more videos and do research on more kinds of behaviors.

**References**

[1] P. Viola, M. Jones, and D. Snow. (2005) “Detecting pedestrians using patterns of motion and appearance,” *IJCV*, vol. 63(2), pp. 153–61.

[2] B. Wu and R. Nevatia. (2005) “Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors,” in *ICCV*, vol. 1, 2005, pp. 90–7.

[3] S.-F. Lin, J.-Y. Chen, and H.-X. Chao. (2001) “Estimation of number of people in crowded scenes using perspective transformation,” *IEEE Trans. System, Man, and Cybernetics*, vol. 31, no. 6, 2001.

[4] V. Rabaud and S. J. Belongie. (2006) “Counting crowded moving objects,” in *CVPR*, 2006.

[5] G. J. Brostow and R. Cipolla. (2006) “Unsupervised bayesian detection of independent motion in crowds,” in *CVPR*, vol. 1, 2006, pp. 594–601.

[6] N. Paragios and V. Ramesh. (2001) “A mrf-based approach for real-time subway monitoring,” in *CVPR*, vol. 1, 2001, pp. 1034–40.

[7] S.-Y. Cho, T. W. S. Chow, and C.-T. Leung. (1999) “A neural-based crowd estimation by hybrid global learning algorithm,” *IEEE Trans. Syst, Man, Cybern.*, vol. 29, pp. 535–41, 1999.

[8] A. C. Davies, J. H. Yin, and S. A. Velastin. (1995) “Crowd monitoring using image processing,” *Electron. Comm. Eng. J.*, vol. 7, pp. 37–47, 1995.

[9] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. (2003) “Dynamic textures,” *Intl. J. Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.

[10]1. R. Xu, D. Wunsch 2nd. (2005) IEEE Trans. *Neural Netw*. 16, 645–678 ,2005.

[11] Rodriguez A, Laio A. (2014) Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191):1492.