Abnormal Human Activity Recognition in Video Surveillance

Assi, W. M.; Onufriev, V. A.

Abstract

The paper presents a new approach for abnormal human activity detection in videos surveillance which is important for many daily purposes. Our approach based on two types of features for detecting anomaly events. First type is motion features based on optical flow calculation, invariant motion speed, and size of moving individuals along frames. Second feature based on predefined model which is known as abnormal action (e.g. car inside park for humans, motorbike passing a road for pedestrian). In this paper we will use UCSD Anomaly Detection dataset. Our aim is to identify anomalies in video feeds using Machine Learning techniques.

Keywords: Abnormal activity recognition, video activity classification, visual surveillance, video anomaly detection.

1 INTRODUCTION

Abnormal human activity detection is very valuable, there are many researches referenced in [1]. Appropriate actions can be taken as soon as it is detected to avoid or reduce negative consequences for administration as well as public safety and security administration in crowded places (Metro stations, supermarkets, parks, airports or malls).

Anomalies can be classified into following two categories:

1. Individual Anomalies: An individual can be considered as anomalous with respect to the rest of individuals [2].
2. Contextual Anomalies: A data instance is anomalous based on a specific scenario but not otherwise [3].

2 DATASET

We will use two Datasets for our experiments:

a) University of California San Diego (UCSD) Anomaly Detection Dataset [4]. It was made by camera putted in a high point, overlooking pedestrian walkways in a park (Fig. 1). The crowd level was ranging from low (counted people) to very crowded. The Dataset consists of two subsets: Ped1 and Ped2.

The first of them, Ped1, contains 34 training and 36 testing videos, while the Ped2 contains 16 training and 12 testing videos.

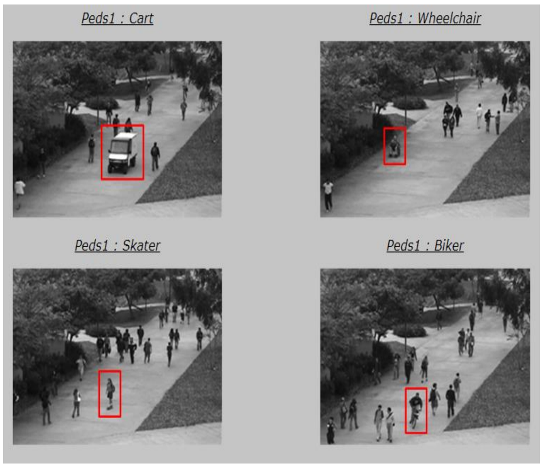

Fig. 1 Examples of Anomalies in UCSD dataset

b) University of Minnesota (UMN) dataset [5], which consists of 11 video for people walking in crowded scenes, there are two types of scenarios, indoor and outdoor scenes.

Each video starts with people walking (normal activity), each of the individuals makes a sudden quick running away which was considered unusual in this dataset (Fig. 2).

Fig. 2 Examples of Anomalies in UMN dataset, left is normal, right is abnormal

3 GENERAL APPROACH OF ABNORMAL HUMAN ACTIVITY DETECTION

There are many basic steps to detect abnormal activities (Fig. 3):

a) preprocessing (extract foreground mask);
b) feature extraction in space (frame level);
c) feature extraction in time (during period of time), and this is optional;
d) build a model/classifier with the final features from training video;
e) comparison of the final features from test video with the model.

There are many different algorithms for each step, but we will describe what we did in our approach for each step.

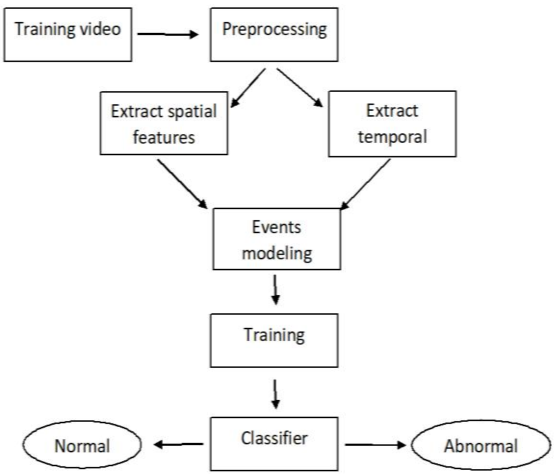

Fig. 3 General Approach

Review of the method

Preprocessing

First of all, we must extract foreground to deal with, that means we deal for benefit data (moving objects only) and ignore background which can be predefined or calculated and updated. In our system we assume that our camera is fixed, but our background maybe will be not static because of many factors like moving trees in windy weather and the difference in lightening during the day.

There many algorithms for background subtraction (BGS) which are supported in the OpenCV library:

a) BackgroundSubtractorMOG: It is a Gaussian Mixture-based Background / Foreground Segmentation Algorithm. It uses a method to model each background pixel by a mixture of K Gaussian distributions (K = 3 to 5). The weights of the mixture represent the time proportions that those colors stay in the scene. The probable background colors are the ones which stay longer and more static.
b) BackgroundSubtractorMOG2: It is also a Gaussian Mixture-based Background / Foreground Segmentation Algorithm. One important feature of this algorithm is that it selects the appropriate number of Gaussian distribution for each pixel. It provides better adaptability to varying scenes due illumination changes etc.
c) BackgroundSubtractorGMG: This algorithm combines statistical background image estimation and per-pixel Bayesian segmentation.

In our system we chose first algorithm because it gave us best result of background and fore ground mask with minimum noise.

Features extraction

Motion features using optical flow

For each frame, optical flow is computed for each pixel using special [6] algorithms. Optical-flow is a vector of the form (*r, θ*), where *r* represents the magnitude of the pixel and *θ* represents the direction in which each pixel has moved relative to the corresponding pixel in the previous frames.

After that we divide each frame into equal blocks (each block is $m \times n$) and calculate optical flow for each block by calculating the average of optical flow of all pixels inside the block (just foreground pixels which represent moving objects)

$$b_i = \frac{1}{j} \sum_j p_i^j ,$$

where: b_i denotes an optical flow of the jth block;
j is the number of pixels in a block;
p_i^j denotes an optical flow of the jth pixel in the ith block.

For orientation we define a vector of 6 value to calculate the orientation in 6 different directions (0, 30, 60, 90,120,150 degrees).

After that we get Optical-flow of a block which is a vector υ_i (1×7) represents how much each block has moved and in which direction compared to the corresponding block in the previous frames, first value for the magnitude of optical flow and the other six values for orientation.

Size feature

Depending only on motion features will not be effective in many scenes, especially crowded scenes (e.g. a bicycle move with the same as pedestrians) it will give similar motion features, so we will use another feature present the size of the block as a number of foreground pixels in this block with respect to the neighboring blocks, because of moving object may occupy more than one block.

For block $b(i, j)$ we define the number of foreground mask inside it as $num(a,b)$, so the size of block will be defined with the equation:

$$size(i,j) = \sum_{a=i-1}^{i+1} \sum_{b=j-1}^{j+1} G(a-i+1, b-j+1) \times num(a,b)$$

Where G is a 3×3 Gaussian mask which is used for placing prominence on the center block and reduce the effect of neighboring blocks.

In the end of this stage, each block is represented by vector υ_i (1×8) and the eighth value denotes to size of the block.

Appearance feature (based on textures or the color of objects)

In crowded scenes, foreground mask will contain pixels for many objects which are moving close to each other causing overlapping, so the false alarm rate will be increased, so we will use another features vector which represents the appearance of the block.

For each block which has foreground pixels we filter a given image using 2D Gabor wavelets at 6 orientations: 0,30,60,90,120,150 degrees, then we get (1×6) vector and each value represent the magnitudes of the wavelet oriented in specific degrees.

These six value will be combined with previous vector to get a new vector V, which component can be numbered from 0 to 13 and contains features:

1. first seven components (from 0 to 6) are the Motion features;
2. the eighth one (7) is size feature;
3. the last six (from 8 to 13) are the appearance features

After that we create $m\times n\times t$ blocks over the most recent t number of frames and calculate features by joining features of each block across all the t frames, so here we have to choose the best value of t which gives the best results (Fig. 4).

So each 3D block will be presented by vector of features $V_{3d}\,(14\times t)$.

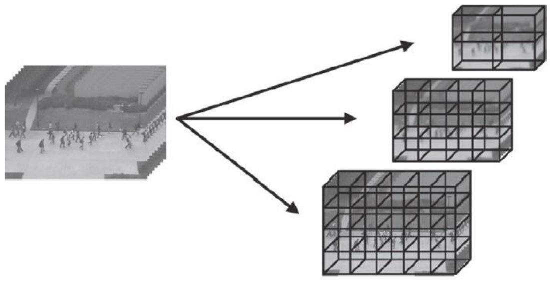

Fig. 4 3D blocks creation

In the end of this stage, from each frame we get feature matrix

$$\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{m1} & \cdots & v_{mn} \end{pmatrix}.$$

Each element of the matrix is V_{3d}.

This matrix will be the input to train the classifier for normal and abnormal detection.

Fig. 5 explains the flowchart of features extraction steps.

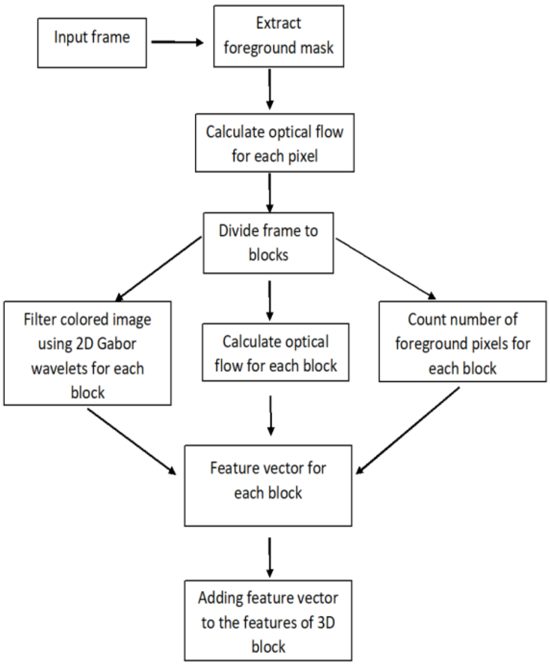

Figure 5: Features extraction

4 CLASSIFICATION

We use support vector machine (SVM) with kernel function and two classes (normal and abnormal).

SVM is classification technique which are based on the concept of decision boundaries to separates between a set of objects having different class memberships in another meaning constructing hyperplanes in a multidimensional space that separates cases of different class labels. SVM with kernel function which is used to convert not separable problem to separable problem, it is mostly useful in non-linear separation problem by introducing additional feature and convert from low, inseparable dimension to higher, separable dimension.

We will try many kernel function and choose the best kernel which give us the best accuracy.

We chose SVM because it works well with low number of classes, and it is easy to train and test, and in our problem we have just two classes.

5 CONCLUSION

We developed the approach for abnormal activities in crowded scenes by mixing features of motion and appearance together, and this way is useful for crowded scenes. Usage of the appearance features is important because there is always individual overlapping in crowded scenes. The presented approach is based on spatial-temporal features and SVM classifier.

Our approach works here because we use small number of classes (just two classes) and we do not have training vectors with high dimensions. In case of using high dimension vectors it will not effective, so in future if we need high dimension we will try to follow an appropriate way to reduce dimension and keep good data.

Also in the future we can improve our work by adding some features depends on tracking, trajectory analysis, and by using other classification algorithms, for example the convolutional neural network algorithms.

6 REFERENCES

[1] Saligrama, V. and Zhu, C: "Video anomaly detection based on local statistical aggregates", 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA: IEEE, 2012, pp. 2112-2119.

[2] Mahadevan, V.; Li, W.; Bhalodia V. and Vasconcelos N.: "Anomaly detection in crowded scenes," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2010.

[3] Gunduz, A. E.; Temizel, T. T. and Temizel, A.: "Pedestrian zone anomaly detection by non-parametric temporal modelling", 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Seoul, South Korea: IEEE, 2014, pp. 131-135.

[4] University of California San Diego ped dataset, 2013, http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm, accessed on 27 July 2018.

[5] Monitoring Human Activity. A project of the Artificial Intelligence, Robotics and Vision Laboratory University of Minnesota, Department of Computer Science and Engineering, http://mha.cs.umn.edu/, accessed on 27 July 2018.

[6] Kondermann D. et al. (2012) On Performance Analysis of Optical Flow Algorithms. In: Dellaert F., Frahm JM., Pollefeys M., Leal-Taixé L., Rosenhahn B. (eds) Outdoor and Large-Scale Real-World Scene Analysis. Lecture Notes in Computer Science, vol 7474. Springer, Berlin, Heidelberg, pp. 329-355.

Assi, Wassim

Peter the Great St. Petersburg polytechnic university

The Higher school of Cyber-physical systems and control

wassimof.86@gmail.com

Onufriev, Vadim

Peter the Great St. Petersburg polytechnic university

The Higher school of Cyber-physical systems and control

ovavadim@gmail.com