2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Anomaly Detection in Crowded Scenes using Genetic Programming

Cheng Xie and Lin Shang

***Abstract*— Genetic programming(GP) has become an increasingly hot issue in evolutionary computation due to its extensive application. Anomaly detection in crowded scenes is also a hot research topic in computer vision. However, there are few contributions on using genetic programming to detect abnormalities in crowded scenes. In this paper, we focus on anomaly detection in crowded scenes with genetic programming. We propose a new method called Multi-Frame LBP Difference(MFLD) based on Local Binary Patterns(LBP) to extract pixel-level features from videos without additional complex preprocessing operations such as optical flow and background subtraction. Genetic programming is employed to generate an anomaly detector with the extracted data. When a new video is coming, the detector can classify every frame and localize the abnormality to a single-pixel level in real-time. We validate our approach on a public dataset and compare our method with other traditional algorithms for video anomaly detection. Experimental results indicate that our method with genetic programming performs better in detecting abnormalities in crowded scenes.**

I. Introduction

ANOMALY detection is an important issue in the fields of video behavior analysis and computer vision. Meanwhile, a lot of surveillance cameras have been installed in many public places due to the increasing attention to public safety and decreasing cost of monitoring device, resulting in a large amount of video data. In most cases, the purpose of surveillance is to analyze real-time behaviors in videos and detect deviations, which are called abnormalities, from the normal in order to ensure social public safety. Because of the lack of insufficient intelligent functions, the traditional surveillance systems require huge human effort. However, manual monitoring has the characteristics of low efficiency and poor real-time while abnormal behaviors of videos are rare and short-lived in general, so the majority of abnormalities will be missed. Therefore, intelligent video monitoring [1], which can drastically reduce manpower expense and enhance the accuracy of anomaly detection, is of great practical significance. In the real world, most cases of surveillance are crowded scenes such as stations, markets, shopping malls and etc. And anomaly detection in crowded scenes becomes a hot topic in computer vision due to their border prospects [2]–[5]. In this paper, we focus on video anomaly detection in crowded scenes.

Various approaches have been proposed for behavior analysis and anomaly detection. These approaches can be divided into two kinds according to the type of scene representation. Approaches of the first category are based on trajectory [6]–[10], which contains a lot of information that objects we detected follow in the scene. Subspace constraints and shape theory are used to recognize deviation of people from a well-controlled path they should follow in [6] [7]. Dee et al. [8] determine if a moving person is normal or abnormal according to the fact that people usually move along regular paths. Robertson et al. [9] add additional information to the tracking-based information for tennis analysis in videos. Other object and frame-based information are added to the trajectories by Porkili et al. [10], they also show anomaly detection in synthetic and simple real-life scenes. The key technology of above methods are to track the objects in videos and get their trajectories at first, which will lead to a phase called modeling learning. Then abnormalities can be detected based on this model. These approaches depend on the accuracy of tracking while it is difficult, even for human beings, to effectively distinguish different trajectories from a crowded scene which contains a large number of moving objects. Furthermore, most researchers have used the manually marked tracks, so most of the methods cannot achieve real-time results.

Many researchers have turned to approaches based on motion features [11]–[15] which are reliably extracted from videos by optical flow, pixel change histograms or some background subtraction operations. These approaches are more robust than trajectory-based approaches. The extracted features contain information about motion direction and magnitude. Then, two popular methods called sparse coding and topic model will be used to train a model for anomaly detection with these features. Sparse coding utilizes the normal scene to train a dictionary and calculate the reconstruction cost to judge whether a scene is an abnormal one or not [12] [15]. Methods of topic model assign different topics to behaviors in a video [13] [14], which can distinguish different types of abnormalities. However, all these approaches will lose abnormality information due to variations of object appearance. Moreover, methods for video feature extraction such as optical flow and background subtraction might not be applicable for crowded scenes, where exist motion background and lots of clutter.

All the methods mentioned above require some complex operations for video preprocessing such as motion detection, object tracking, video summarization, crowd counting, image segmentation and etc. In this paper, we propose a new simple method for feature representation without any complex preprocessing. The extracted features will contain spatial-temporal information of videos. After the phase of feature extraction, GP is used to address the classification task.

GP has proved to be an effective technology in problem solving and has been successfully applied in a wide variety of fields [16], especially in computer vision related tasks

Cheng Xie and Lin Shang are with the State Key Laboratory for Novel Software Technology, Nanjing University, China (Email: {clisely, lshang.nju}@gmail.com).

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

[17]–[21]. Zhang et al. [17] used GP for problems of various object detection. GP was employed by Howard et al. [18] to detect vehicles in different indoor and outdoor environments such as industrial, urban and rural in images. The GP-based motion detection method was introduced by Song et al. in [19] and has proved effective in various real world scenarios [20] [21]. It has proved to be true that the detectors generated by GP can achieve similar or even better performance to the traditional human designed models developed for those tasks. However, by far there are few researchers working on the issue of applying GP to video anomaly detection in crowded scenes. Our work will represent this application.

In this paper, we aim to investigate the effectiveness of genetic programming for video anomaly detection in crowded scenes. Firstly, we will propose a new simple method to extract pixel-level features containing spatial-temporal information from videos, while traditional methods require a slightly more complex operations for feature extraction such as optical flow and background subtraction Secondly, we will employ genetic programming to generate an anomaly detector with some training data, where a training sample is a cutout with label negative or positive, divided from a frame with grids. A negative sample means that there is no abnormality in the subregion while positive sample means that abnormal object or event exists in the subregion in contrast. When a new video is coming, the detector can classify each frame of this video and localize the abnormality to a single-pixel level. Finally, we validate our approach on the public dataset UCSD [1] and compare our method with other traditional approaches for anomaly detection in the same dataset. Experimental results indicate that our method with genetic programming performs better in detecting abnormalities in crowded scenes.

The main work in this paper are as follows:

- Represent a new study of applying genetic programming to video anomaly detection in crowded scenes.
- Propose a new simple method for feature extraction from videos without additional complex preprocessing.
- Prove that our GP-based method can perform anomaly detection on new-coming videos in real-time.

The remainder of this paper is organized as follows. In section II we will introduce the genetic programming. The proposed method for anomaly detection in crowded scenes is described in detail in section III. The dataset and experimental results are presented in section IV, and comparison with traditional methods is also provided. Finally, section V concludes the paper.

II. Genetic Programming

Genetic programming [22] is a systematic method for getting computers to automatically solve a problem starting from a high-level statement of what needs to be done using principles of Darwinian evolution. Genetic programming is derived from genetic algorithm(GA). The main difference between them is that programs in GP are expressed as tree structure rather than as lines of code in GA. The anomaly detector is also expressed in tree structure. Furthermore, a program tree in this situation for anomaly detection is regarded as a classifier which takes features of small subregions or cutouts from video frames as the input and produces an output to decide which class that input subregion should belong to. Anomaly detection only includes two classes: abnormal and normal. A normal sample, labeled with negative, means that there is no abnormality in the subregion while abnormal sample labeled with positive means that abnormal object or event exists in the subregion. However, the output of a program tree is usually a real number and the results we need are class labels. So we use dynamic range selection [23] to determine the mapping between output values and class labels. Fig.1 shows a simple GP program tree, representing the detector for anomaly detection.

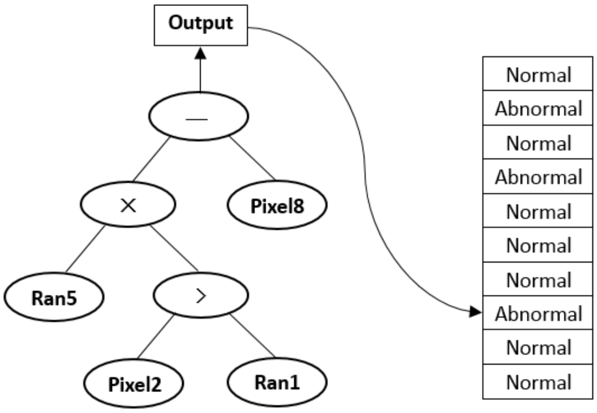


Fig. 1. A simple GP program tree. The mapping between output values and class labels is determined by dynamic range selection.

In a GP program tree, the nodes indicate the instructions to execute and the links indicate the arguments for each instruction. The internal nodes in a tree will be called $Functions$, while the tree's leaves will be called $Terminals$. For a specific problem, the ingredients include specialized functions and terminals. For example, if the goal is to get GP to automatically program a robot to mop the entire floor of an obstacle-laden room, the human user must tell genetic programming what the robot is capable of doing. For example, the robot may be capable of executing functions such as moving, turning, and swishing the mop.

GP is a domain-independent method that genetically breeds a population of computer programs to solve a problem. Specifically, given a problem, GP typically starts with a population of randomly generated computer programs composed of available functions and terminals. For each GP program, a number of samples with labels are used to evaluate its performance, which is recorded as the $Fitness$. Then, GP will transform a population of computer programs into a new generation of programs. The individual programs for the new population are created by applying analogs of naturally occurring genetic operations with specified probabilities: $Reproduction$, copying the selected individual programs

[1] http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm

withing higher fitness to the new population; $Crossover$, randomly recombining chosen parts from two selected programs; $Mutation$, randomly mutating a randomly chosen part of one selected program. Performance evaluation is also needed for the new programs on the same detection task. Iteratively perform the sub-steps, called a $Generation$, of performance evaluation and individual generation on the population until a best program is found or the maximum number of generation is reached. Then the best GP program is the solution of this problem.

III. Our Approach For anomaly detection in crowded scenes

In this section, we are going to explain our GP-based method for anomaly detection in crowded scenes in detail. The overview of our method is illustrated in Fig.2. There are three main phases in our method:

- Feature Extraction. Extract pixel-level features from videos using our proposed method and divide them into training set and testing set with manual labels.
- GP Evolution. Include two stages: Generating, which is to generate GP programs as anomaly detectors given the training data. Evaluating, which is to evaluate the generated anomaly detector on testing data. The best performing program with highest fitness during the evaluating stage is then selected as the anomaly detector.
- Anomaly Prediction. The produced anomaly detector is applied to perform anomaly prediction on new-coming video streams and localize the abnormality to a single-pixel level.

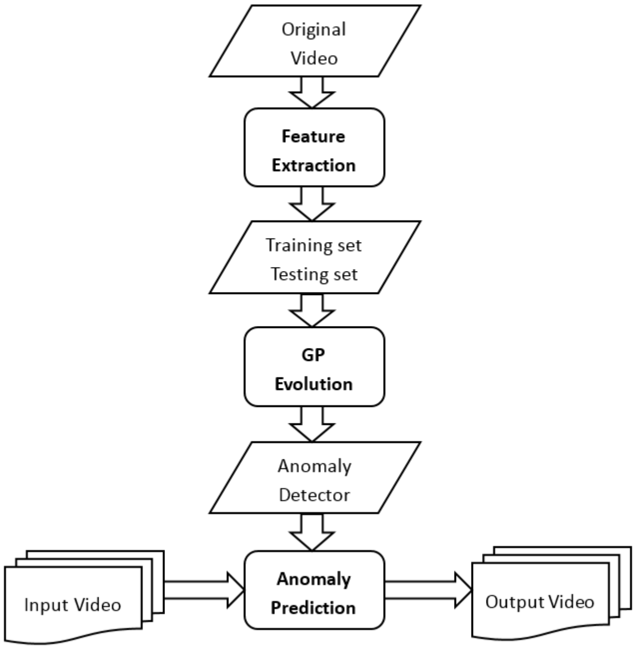


Fig. 2. Overview of our method

A. MFLD

Traditional methods for anomaly detection require some operations for video preprocessing such as motion detection, object tracking, video summarization, crowd counting, image segmentation and etc. However, all these operations are complex and time-consuming. To avoid the shortage of traditional methods, we propose a new simple method for feature representation without any complex preprocessing based on local binary patterns(LBP).

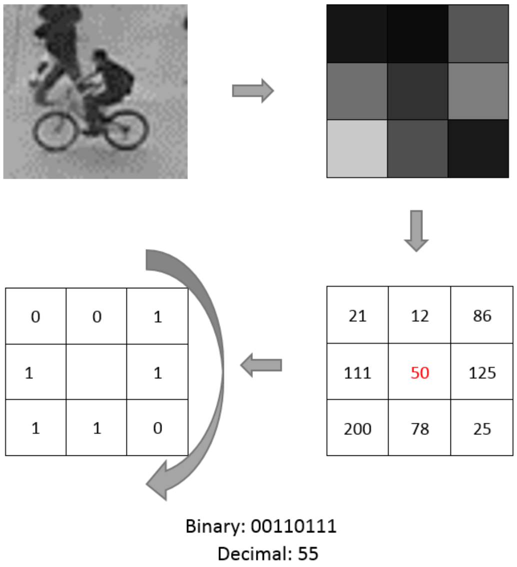
The original LBP operator was introduced by Ojala et al. [24], designing for texture description originally. The operator assigns a label to every pixel of an image by thresholding the 3×3-neighborhood of each pixel with the center pixel value and considering the result as a binary number. The LBP value of each pixel is calculated as follows:

$$LBP(x) = \sum_{k=0}^{7} B(G_k - G_x) \times 2^k \tag{1}$$

In Equation 1, G_x is the gray-scale value of the center pixel while the G_k is the gray-scale value of the eight corresponding neighborhood pixels. $B(x)$ is a binary function, based on the following definition:

$$B(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2}$$

Since then LBP has become a basic representation for pixel-level feature, obtaining continuous improvement and optimization. Also it has been applied in various fields, especially in face detection [25] [26] . Fig.3 shows the representation of LBP.



0	0	1
1		1
1	1	0

21	12	86
111	50	125
200	78	25

Binary: 00110111
Decimal: 55

Fig. 3. Representation of LBP. In a 3×3 pixels image, the gray-scale value of the center pixel is 50, and the LBP value of this pixel is equal to 55 in decimal.

Equation 1 calculates the LBP value of each pixel in a frame. Then we propose a new feature representation of each pixel based on the LBP. This representation calculates the average difference of LBP between each pair of consecutive frames over n number of frames. We define this new representation as Multi-Frame LBP Difference(MFLD). The MFLD value of each pixel x in a frame is calculated as follows:

$$MFLD(x) = \frac{\sum_{i=1}^{n-1}(|LBP_{x^i} - LBP_{x^{i+1}}|)}{n-1} \qquad (3)$$

In Equation 3, each pixel x of a frame is calculated as the average difference between the LBP of pixel x for n previous frames. LBP_{x^i} refers to the LBP of pixel x in frame i, where i = 1 refers to the current frame, i = 2 refers to the previous frame, i = 3 refers to the second previous frame and so on. Fig.4 shows the representation of MFLD.

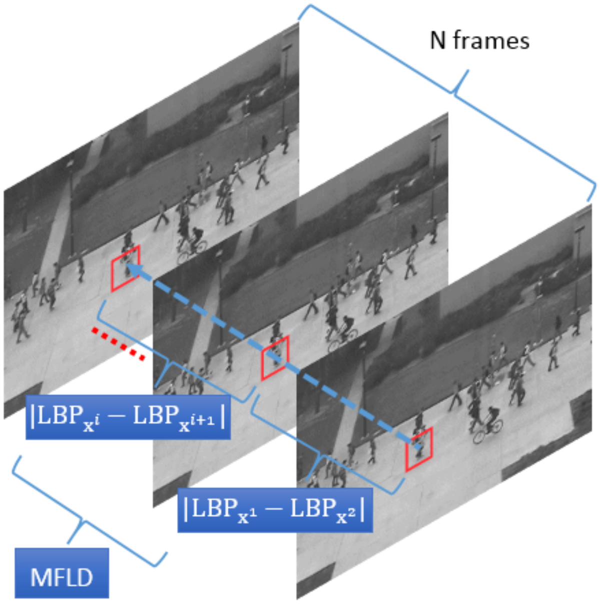


Fig. 4. Representation of MFLD. The MFLD value of a pixel is equal to the average difference of LBP between each pair of n consecutive frames.

Since LBP contains local spatial information of each frame and the MFLD takes consecutive frames into account. The proposed MFLD based on LBP for feature extraction can extract pixel-level features containing spatial-temporal information from video with no need for complex operations compared to traditional methods.

B. Feature Extraction

Before performing genetic programming, a series of actions need to be performed to deal with the given videos:

- Randomly select several frames containing normal and abnormal objects from the video.
- Calculate the MFLD value of each pixel in frames according to Equation 3.
- Divide the selected frames into a grid with a fixed cell to extract feature vectors. For example, if the resolution is 200 × 100, and each cell is 10 × 10, then the grid is 20 × 10. That means a frame contains 200 samples and a sample contains 100-dimensional features. The value of each dimension is equal to the MFLD value of corresponding pixel. The number of features grows with the increasing size of the cell, which is problem and video dependent. A large size would require more computational resource and increase the search space. Otherwise, a small size may leads to low accuracy for classification despite decreasing the computational-complexity.
- Label the samples manually. The classes of anomaly detection only include abnormal and normal. A normal sample regarded as negative means that there is no abnormality in the subregion while abnormal sample regarded as positive means that abnormal object or event exists in the subregion.
- Randomly divide the generated samples into two data sets: the training set and the testing set.

Each sample of the data set consists of several features and only one label, which represent the input and output respectively in GP. Each input item is a pixel represented in MFLD representation and the corresponding output is the expected label for that cutout from the video frame.

C. GP Evolution

The main purpose of the GP evolution is to evolve a program as the anomaly detector. After finishing the phase of feature extraction, we generate the training data set and the testing set, which are used to evolve and to evaluate the anomaly detector respectively. For the problem of anomaly detection, regions where abnormal objects or events occur usually occupy a small proportion of a video frame compared to regions of normal cases. Therefore negative samples are the majority. To avoid imbalance, we randomly remove some negative samples both in training set and testing set. So the number of negative samples is almost the same amount as the positive samples. The best detector with the highest fitness on the testing data set is then selected as the anomaly detector to process videos in the phase of anomaly prediction.

First of all, we should specify several major preparatory steps of GP.

1) Function Set: As described in section II, each GP program is represented in tree structure, on which the internal nodes are made up of functions from the function set and the external nodes are made up of terminals from the terminal set. The function set used in our experiments is described in Table I. The input represents the number and the type of values that operator can deal with. The output represents the type of returned value of that operator. In our experiments, we use arithmetic operators, comparison operators and operator If. Arithmetic operators contains addition(+), subtraction(−), multiplication(×) and division(/). For the division operator, if a number is divided by zero, then the result is zero. All these four arithmetic operators have two real input values and one real returned value. Comparison operators includes equal to(=), larger than(>) and less than(<). The input of them is same to the arithmetic operators, but the returned type is boolean. The operator If needs three inputs, representing that if the first input is true or equal to one, then the second input is returned, else the third input is returned. The first input of operator If can be any one of the output of comparison operators. Each real input is either terminal from the terminal set or output of one of the operators, whose returned type is real.

TABLE I
FUNCTION SET

Function	Output	Input
$+$	Real	Real, Real
$-$	Real	Real, Real
$\times$	Real	Real, Real
$/$	Real	Real, Real
$=$	Boolean	Real, Real
$>$	Boolean	Real, Real
$<$	Boolean	Real, Real
If	Real	Boolean, Real, Real

2) Terminal set: Table II shows the terminal set used in our experiments. Apart from the $Feature$ terminal that represents the MFLD representation of a video frame cutout, another terminal is $Rand$, which is a random number set between 0 and 1, which would act like coefficient in normal functions.

TABLE II
TERMINAL SET

Terminal	Type	Value
Rand	Real	[0,1]
Feature	Real	MFLD

3) Fitness: The fitness measure, evaluating the performance of individual programs, is the primary mechanism for communicating the high-level statement of the problem's requirements to the genetic programming system. In addition, the function and terminal set define the search space whereas the fitness implicitly specifies the desired goal of the search. In our experiments, we regard the accuracy on classifying the training data as the fitness, calculated as Equation 4.

$$Fitness = \frac{TP + TN}{NUM} \times 100 \tag{4}$$

In Equation 4, the fitness is normalized between 0 to 100. TP and TN represent true positive and true negative respectively. TP stands for the number of correctly classified positive samples. Analogously, TN stands for the number of correctly classified negative samples. NUM is the total number of samples in testing set. This evaluation criterion is basically the classification accuracy or anomaly detection accuracy. GP programs with higher fitness are desired.

4) Runtime Parameter: Table III shows the specific control parameters for the run of GP in our experiments. The population size is set to 1000 here, which can maintain the balance between time cost and detector's performance. The maximum number of generations is set to 500. The depth of a program tree of each individual is limited between 3 and 9. Furthermore, the rate of crossover, reproduction and mutation is set to 0.8, 0.1 and 0.1 respectively to generate new programs. These parameters were found to be effective according to amount of experiments.

TABLE III
GP RUNTIME PARAMETERS

Parameter	Value
Population Size	1000
Generations	500
Minimum Depth	3
Maximum Depth	9
Crossover Rate	0.8
Reproduction Rate	0.1
Mutation Rate	0.1

With these parameters, we conduct the GP for classification on ECJ [2], a public Java-based evolutionary computation research system. During the iterative runtime, if the fitness of a program calculated by Equation 4 reaches 100, meaning that this detector can classify all the testing samples correctly, then the GP will be terminated. Otherwise, it will run until the preset number of generations has been reached. The best GP program with highest fitness will be selected as the anomaly detector for anomaly prediction.

D. Anomaly Prediction

With the best GP program selected in the evolution phase, it can be regarded as the anomaly detector in the problem of anomaly prediction and will be applied to new videos of same scenes with the video for feature extraction.

Fig.5 shows the procedure of anomaly prediction. When a video is coming, it will be processed as follows:

- Retrieve all frames of the video without any preprocessing work.
- Convert each frame into MFLD representation.
- For each frame, sample subregions by a sliding window, which is with the same size as the cell for feature extraction. The sliding window moves from the top-left corner to the bottom-right corner of the frame with the moving step of m pixels, where m can be adjusted manually.
- Convert each 2D window of the frame into an array of features, which are given as input to the anomaly detector.
- The anomaly detector will classify the subregion as either positive or negative for anomaly according to the input.
- Since the moving step is usually less than the size of window, sliding windows in a frame will overlap

[2] Available from http://cs.gmu.edu/∼eclab/projects/ecj/

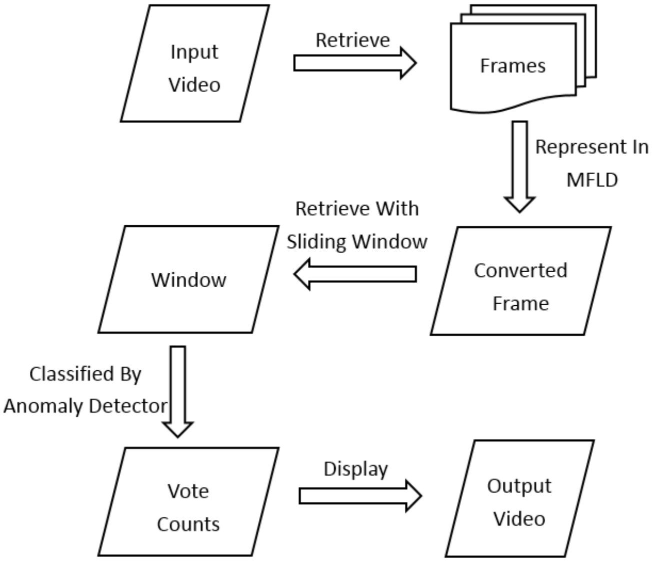


Fig. 5. Procedure of Anomaly Prediction

to some extent. A pixel will be classified by different overlapped sliding windows, so a vote count is adopted to record the number of positive and negative votes for each pixel.

- Once a frame has been retrieved by all sliding windows, each pixel of this frame will be colored red if it has more positive than negative votes.
- Display the final results intuitively in video format. The red regions of video contain the detected abnormal objects or events.

IV. Experiments

In this section, we will show how our method will work to detect abnormalities in videos under crowd scenes.

A. Dataset

We test our approach on the UCSD anomaly dataset in crowded scenes presented in [3]. This dataset was gathered in UCSD campus by a fixed camera, overlooking pedestrian walkways. The crowd density of the walkways ranges from sparse to very crowded. The normal objects in this dataset are only pedestrians while the abnormal objects include bikers, skaters, carts and people walking across a walkway or in the surrounding grass. That means non-pedestrian objects accessing the walkway and pedestrians moving in anomalous patterns or in non-walkway regions are abnormalities. Fig.6 shows the normal and the abnormal events of this video.

The dataset consists of two subsets, corresponding to different scenes. We use the second subset called Ped2 to verify our method. This subset contains scenes with moving pedestrians, which are parallel to the camera. The camera is recorded at 10 FPS with a resolution of 360×240. Moreover, the video was split into 16 clips of training sets and 12 clips of testing sets, each of which contains between 120 and 200 consecutive frames. Corresponding manually generated pixel-level binary masks are also provided, which identify the regions containing anomalies. This is intended to enable the performance evaluation with respect to the ability to localize anomalies.

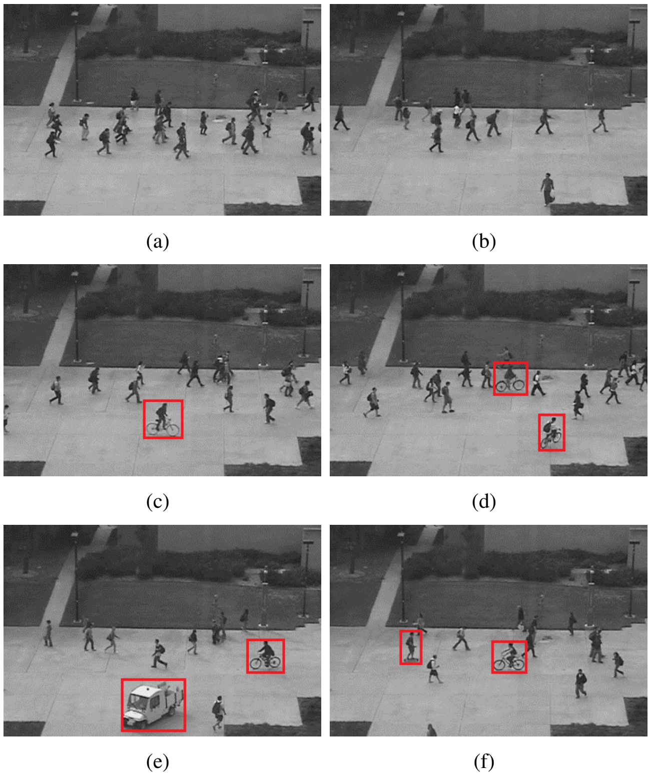

(a) (b) (c) (d) (e) (f)

Fig. 6. Normal and Abnormal events in Ped2 of UCSD dataset. (a) and (b) only contain normal event of pedestrians. (c), (d), (e) and (f) contain abnormalities in red boxes such as bikers, skaters and carts.

B. Anomaly Detection

In order to validate our approach, we conduct experiments in terms of three phases discussed in Section III.

1) Feature Extraction: Firstly, randomly select 100 frames containing both normal and abnormal events and represent them in MFLD representation. The parameter n representing the consecutive frames is set as 15. Then divide each frame into a grid with 15×15 pixels cells. Afterwards, transform each cell into a feature vector. Each vector consists of 225-dimensional features. The value of each dimension is equal to the MFLD value of corresponding pixel. Not all samples are employed. A total of 1100 positive and 1500 negative samples of them are created, among which, 500 positive and 800 negative samples are randomly selected as the training set and the remaining are for the testing set.

2) GP Evolution: With the training and testing set, which are used to evolve and to evaluate the anomaly detector respectively. We can conduct GP for classification task according to the parameters defined in Table III. The best program with the highest fitness on the testing data set is then selected as the anomaly detector. Phase of GP evolution is the most significant phase of our method. In addition, GP has the properties that each run is probabilistic, and it will virtually never produce the same result for different runs, which attempt to solve the same problem. Ten different programs are generated from ten independent runs by genetic programming, the best of which is then selected for anomaly prediction.

3) Anomaly Prediction: When a new video is coming, transform each frame of the video into MFLD representation. Then employ a 15×15 pixels sliding window to move from the top-left corner to the bottom-right corner of the frame with the moving step of 3 pixels. Feature vectors of subregions will be produced. Anomaly detector can generate classes according to these data. A vote count is also adopted to record the number of positive and negative votes for each pixel. Once a frame has been retrieved by all sliding windows, each pixel of this frame will be colored red if it has more positive than negative votes. Parts of the visible results are presented in Fig.7. The red regions contain the detected abnormal objects or events.

C. Performance Evaluation

In order to evaluate the performance of the anomaly detector generated by GP, two criteria called $Frame-Level\ Criterion$ and $Pixel-Level\ Criterion$ are emploied [4].

- $Frame-Level\ Criterion$, an abnormal frame is considered correctly detected if at least one abnormal pixel of the frame is detected as anomalous, which is compared to the corresponding frame-level ground-truth anomaly annotations.
- $Pixel-Level\ Criterion$, an abnormal frame is considered correctly detected if at least the 40 percent of its anomalous pixels are detected correctly, which is compared to the corresponding pixel-level ground-truth anomaly annotation.

However, a lucky phenomenon happens when a region different from the one that generated the anomaly is detected as anomalous in the same frame. The frame-level detection evaluation does not takes this phenomenon into account. The pixel-level criterion is much stricter and more rigorous. By evaluating both the temporal and spatial accuracy of the anomaly predictions, it rules out these lucky co-occurrences. Performance is also summarized by the equal error rate(EER), the ratio of misclassified frames for the frame-level criterion, or rate of detection(RD) for the pixel-level criterion. A method with lower EER and higher RD is desired.

We compare our method with other traditional state-of-the-art approaches on the same dataset, whose results are reported in [27]: Kim et al. [28], Adam et al. [29], Mehran et al. [30], Mahadevan et al. [3] and Bertini et al. [27]. The performance of the different descriptors, under both the frame-level (EER) and pixel-level (RD) criteria is summarized in Table IV.

As seen from the Table IV, the performance of our method at the frame-level is close to the Mehadevan et al. and is better than the others. However, it is noted that the approach of Mehadevan et al. includes complex preprocessing steps. Moreover, it is not appropriate for real-time detection because it takes almost 25 seconds to process a single frame, while our approach can deal with unseen videos in real-time with the simple feature extraction method based on MFLD and the pre-generated anomaly detector. Our method is far superior in the anomaly localization task to all other methods including the Mehadevan et al. The good results in anomaly localization imply that we are not taking advantage of lucky guesses, but that we accurately localize the abnormal objects in videos.

Furthermore, from the Fig.7 which presents the visual results of anomaly detection in unseen videos, we can see that our approach can not only detect the abnormal frames but also localize the majority pixels of abnormal regions in real-time.

TABLE IV

ANOMALY DETECTION PERFORMANCE COMPARISON WITH STATE-OF-THE-ART ON PED2 DATASETS. EER IS REPORTED FOR FRAME-LEVEL ANOMALY DETECTION. RD IS PRESENTED FOR THE PIXEL-LEVEL CRITERION.

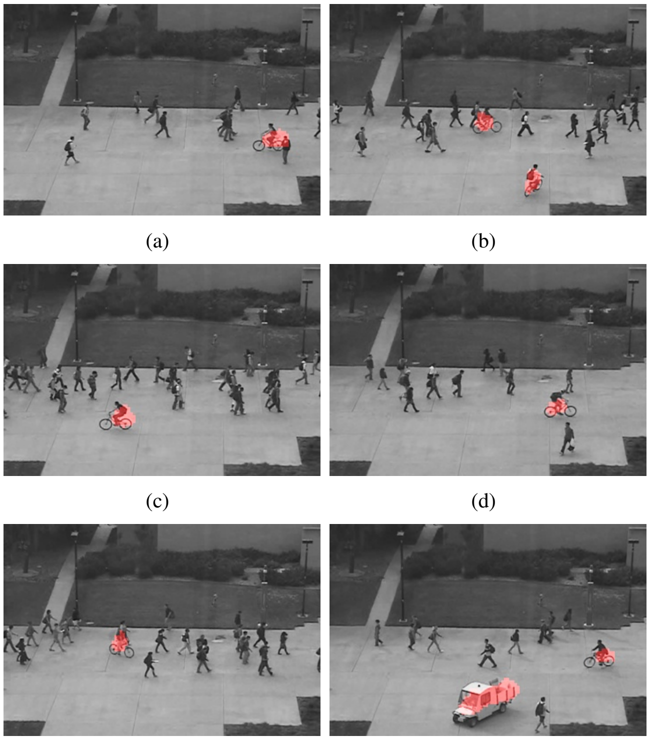
Approach	EER	RD
Kim et al.	30 %	18 %
Mehran et al.	42 %	21 %
Adam et al.	42 %	24 %
Bertini et al.	30 %	29 %
Mehadevan et al.	25 %	45 %
Proposed Method	28 %	65 %

V. CONCLUSION

In this paper, we have proposed a new method for anomaly detection in crowded scenes based on genetic programming. Firstly, without often-needed traditional preprocessing such as noise removal, optical flow and background subtraction, we use our proposed method MFLD, which is based on LBP, to extract features containing spatial-temporal information from videos. Then, GP is employed to automatically construct anomaly detection programs with the data in MFLD representation. Finally, the best performing program can be directly applied on unseen video to detect abnormal objects or events and show them in video.

In our approach, manual algorithm development process and traditional preprocessing steps for videos can be avoided or at least significantly reduced. It is less dependent on hypotheses and models from traditional vision and image processing methods. Instead a search process is performed to find the best possible program. The experimental results compared with methods under traditional video anomaly detection framework indicate that our approach is comparable with the traditional methods, even better than most of them. Moreover, due to the simple proposed feature extraction method based on MFLD, we can detect abnormalities on new-coming videos in real-time.

In short, the proposed GP-based anomaly detection method in crowded scenes is easy to use, fast in development, and can produce reliable detection outcomes at the same time. In future, we will test the method on more different scenarios and study other methods of feature extration for anomaly detection.



(a) (b) (c) (d) (e) (f)

Fig. 7. Anomaly prediction results on unseen videos by our approach. Objects in red regions are abnormal ones.

ACKNOWLEDGEMENTS

The work is supported by the NSFC programs of China (No. 61170180 and No. 61035003).

REFERENCES

[1] Singh V, Kankanhalli M. Adversary aware surveillance systems. IEEE Transactions on Information Forensics and Security. 2009: 552-563.

[2] Najafabadi M M, Rahmati M. Anomaly detection in structured/unstructured crowd scenes. In: ICDIM. 2012: 79-83.

[3] Mahadevan V, Li W, Bhalodia V, Vasconcelos N. Anomaly detection in crowded scenes. In: CVPR. 2010: 1975-1981.

[4] Li W, Mahadevan V, Vasconcelos N. Anomaly Detection and Localization in Crowded Scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2013: 1-15.

[5] Cong Y, Yuan J, Tang Y. Video Anomaly Search in Crowded Scenes via Spatio-Temporal Motion Context. IEEE Transactions on Information Forensics and Security. 2013(8): 1590-1599.

[6] Chowdhury A, Chellappa R. A Factorization Approach for Activity Recognition. In: CVPR. 2003: 41.

[7] Vaswani N, Chowdhury A, Chellappa R. Activity Recognition Using the Dynamics of the Configuration of Interacting Objects. In: CVPR. 2003.

[8] Dee H, Hogg D. Detecting Inexplicable Behaviour. In:BMVC. 2004: 1-10.

[9] Robertson N, Reid I. Behaviour Understanding in Video: A Combined Method. In: ICCV. 2005: 808-815.

[10] Porkili F, Haga T. Event Detection by Eigenvector Decomposition Using Object and Frame Features. In:CVPR. 2004: 114.

[11] Tran D, Yuan J. Optimal spatio-temporal path discovery for video event detection. In: CVPR. 2011: 3321-3328.

[12] Cong Y, Yuan J, Liu J. Sparse reconstruction cost for abnormal event detection. In: CVPR. 2011: 3449-3456.

[13] Haines T S F, Xiang T. Delta-dual hierarchical dirichlet processes: A pragmatic abnormal behaviour detector. In: ICCV. 2011: 2198-2205.

[14] Wang X, McCallum A. Topics over time: a non-markov continuous-time model of topical trends. In: KDD. 2006: 424-433.

[15] Zhao B, Li F F, Xing E P. Online detection of unusual events in videos via dynamic sparse coding. In: CVPR. 2011: 3313-3320

[16] Poli R, Langdon W W B, McPhee N F. A field guide to genetic programming. Lulu.com. 2008.

[17] Chin B, Zhang B. Object detection using neural networks and genetic programming. LNCS: Applications of Evolutionary Computing. 2008: 335-340.

[18] Howard D, Roberts S C, Ryan C. Pragmatic genetic programming strategy for the problem of vehicle detection in airbourne reconnaissance. In: ECVIP. 2006: 1161-1306.

[19] Song A, Fang D. Robust method of detecting moving objects in videos evolved by genetic programming. In: GECCO. 2008: 1649-1656.

[20] Pinto B, Song A. Detecting motion from noisy scenes using genetic programming. In: IVCNZ. 2009: 322-327.

[21] Shi Q, Song A. Selective motion detection by genetic programming. In: CEC. 2011: 496-503.

[22] Koza J R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT press. 1992.

[23] Loveard T, Ciesielski V. Representing classification problems in genetic programming. In: CEC. 2001: 1070-1077.

[24] Ojala T, Pietikainen M, Harwood D. A comparative study of texture measures with classification based on featured distributions. Pattern recognition. 1996. 29(1): 51-59.

[25] Wang X, Han T X, Yan S. An HOG-LBP human detector with partial occlusion handling. In: ICCV. 2009: 32-39.

[26] Ahonen T, Hadid A, Pietikainen M. Face recognition with local binary patterns. In: ECCV. 2004: 469-481.

[27] Bertini M, Del B A, Seidenari L. Scene and crowd behaviour analysis with local space-time descriptors. In: ISCCSP. 2012: 1-6.

[28] Kim J, Grauman K. Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates. In: CVPR. 2009: 2921-2928.

[29] Adam A, Rivlin E, Shimshoni I, Reinitz D. Robust real-time unusual event detection using multiple fixed-location monitors. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2008. 30(3): 555-560.

[30] Mehran R, Oyama A, Shah M. Abnormal crowd behavior detection using social force model. In: CVPR. 2009: 935-942.