2018 IEEE Winter Conference on Applications of Computer Vision

Plug-and-Play CNN for Crowd Motion Analysis: An Application in Abnormal Event Detection

Mahdyar Ravanbakhsh[1*] mahdyar.ravan@ginevra.dibe.unige.it
Moin Nabi[2,3] m.nabi@sap.com
Hossein Mousavi[4,5] hossein.mousavi@iit.it
Enver Sangineto[2] enver.sangineto@unitn.it
Nicu Sebe[2] niculae.sebe@unitn.it

[1] University of Genova, Italy [2] University of Trento, Italy [3] SAP SE, Berlin, Germany
[4]Istituto Italiano di Tecnologia, Italy [5]Polytechnique Montréal, Montréal, Canada

Abstract

Most of the crowd abnormal event detection methods rely on complex hand-crafted features to represent the crowd motion and appearance. Convolutional Neural Networks (CNN) have shown to be a powerful instrument with excellent representational capacities, which can leverage the need for hand-crafted features. In this paper, we show that keeping track of the changes in the CNN feature across time can be used to effectively detect local anomalies. Specifically, we propose to measure local abnormality by combining semantic information (inherited from existing CNN models) with low-level optical-flow. One of the advantages of this method is that it can be used without the fine-tuning phase. The proposed method is validated on challenging abnormality detection datasets and the results show the superiority of our approach compared with the state-of-the-art methods.

1. Introduction

Crowd analysis gained popularity in the recent years in both academic and industrial communities. This growing trend is also due to the increase of population growth rate and the need of more precise public monitoring systems. In the last few years, the computer vision community has pushed on crowd behavior analysis and has made a lot of progress in crowd abnormality detection [21, 43, 17, 9, 28, 27, 26, 23, 6, 3, 1, 20]. Most of these methods mainly rely on complex hand-crafted features to represent the crowd motion and appearance. However, the use of hand-crafted features is a clear limitation, as it implies task-specific a priori knowledge which, in case of a complex video-surveillance scene, can be very difficult to define. Recently, Deep Neural Networks have resurfaced as a powerful tool for learning from big data (e.g., ImageNet [34] with 1.2M images), providing models with excellent representational capacities. Specifically, Convolutional Neural Networks (CNNs) have been trained via backpropagation through several layers of convolutional filters. It has been shown that such models are not only able to achieve state-of-the-art performance for the visual recognition tasks in which they were trained, but also the learned representation can be readily applied to other relevant tasks [33]. These models perform extremely well in domains with large amounts of training data. With limited training data, however, they are prone to overfitting. This limitation arises often in the abnormal event detection task where scarcity of real-world training examples is a major constraint. Besides the insufficiency of data, the lack of a clear definition of abnormality (*i.e.*, the context-dependent nature of the abnormality) induces subjectivity in annotations. Previous work highlighted the fact that the unsupervised measure-based methods may outperform supervised methods, due to the subjective nature of annotations as well as the small size of training data [42, 38, 41, 24].

Attracted by the capability of CNN to produce a general-purpose semantic representation, in this paper we investigate how to employ CNN features, trained on large-scale image datasets, to be applied to a crowd dataset with few abnormal event instances. This can alleviate the aforementioned problems of supervised methods for abnormality detection, by leveraging the existing CNN models trained for image classification. Besides, training a CNN with images is much cheaper than with videos; therefore, representing a

*Work done while M. Ravanbakhsh was an intern at DISI, University of Trento.

978-1-5386-4886-5/18/$31.00 ©2018 IEEE
DOI 10.1109/WACV.2018.00188

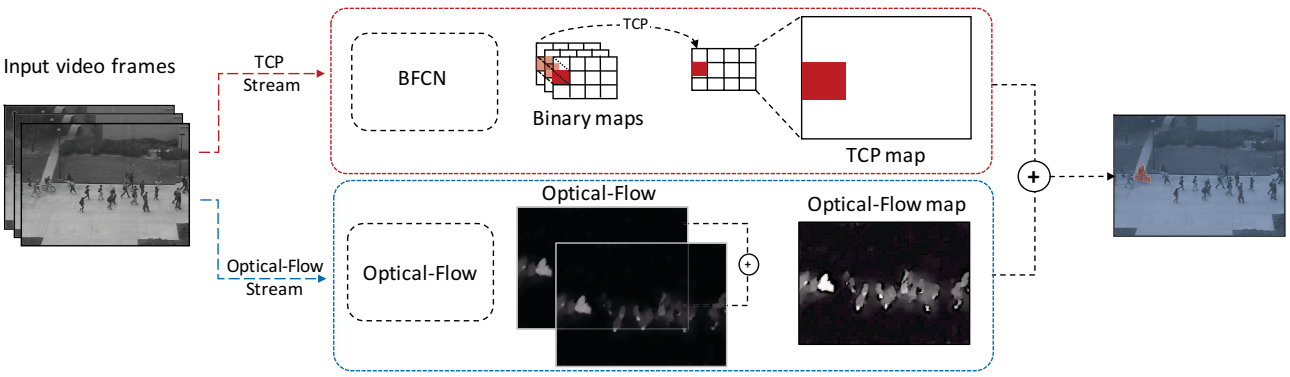


Figure 1. Overview of the proposed method

video by means of features learned with static images represents a major saving of computational cost.

The key idea behind our method is to track the changes in the CNN features across time. We show that even very small consecutive patches may have different CNN features, and this difference captures important properties of video motion. To capture the temporal change in CNN features, we cluster them into a set of binary codes each representing a binary pattern *(prototype)*. Intuitively, in a given video block consecutive frames should have similar binary patterns unless they undergo a significant motion. We introduced a simple yet effective statistical measure which captures the local variations of appearance in a video block. We show that combining this measure with traditional optical-flow information, provides the complementary information of both appearance and motion patterns.

Previous Work: Our method is different from [21, 22, 18, 5, 12, 36, 17, 20], which focus on learning models on motion and/or appearance features. A key difference compared to these methods is that they employ standard hand-crafted features (*e.g.*, optical-flow, Tracklets, etc.) to model activity patterns, whereas our method proposes using modern deep architectures for this purpose. The advantages of a deep learning framework for anomalous event detection in crowd have been investigated recently in [43, 35]. Nevertheless, deep networks are data-hungry and need large training datasets. In our work, however, a completely different perspective to abnormality detection is picked out. We specifically propose a measure-based method which allows the integration of semantic information (inherited from existing CNN models) with low-level optical-flow [2], with minimum additional training cost. This leads to a more discriminative motion representation while maintaining the method complexity to a manageable level. Most related to our paper is the work by Mousavi *et al*. [24], which introduced a similar measure to capture the commotion of a crowd motion for the task of abnormality detection. Instead of capturing the local irregularity of the low-level motion features (e.g., tracklets in [24]) or high-level detectors [25], we propose to represent the crowd motion exploiting the temporal variations of CNN features. This provides the means to jointly employ appearance and motion. Very recently Ravanbakhsh *et al*. [30] proposed a complex feature structure on top of CNN features which can capture the temporal variation in a video for the task of activity recognition. However, to our knowledge this is the first work proposing to employ the existing CNN models for motion representation in crowd analysis.

Method Overview: Our method is composed of three steps: 1) Extract CNN-based binary maps from a sequence of input frames, 2) Compute the Temporal CNN Pattern (TCP) measure using the extracted CNN-binary maps 3) The TCP measure fuse with low-level motion features (optical-flow) to find the refined motion segments.

More specifically, all the frames are input to a Fully Convolutional Network (FCN). Then we propose a binary layer plugged on top of the FCN in order to quantize the high-dimensional feature maps into compact binary patterns. The binary quantization layer is a convolutional layer in which the weights are initialized with an external hashing method. The binary layer produces binary patterns for each patch corresponding to the receptive field of the FCN, called *binary map*. The output binary maps preserve the spatial relations in the original frame, which is useful for localization tasks. Then, a histogram is computed over the output binary maps for aggregating binary patterns in a spatio-temporal block. In the next step, an *irregularity* measure is computed over these histograms, called TCP measure. Eventually, all the computed measures over all the video blocks are concatenated, up-sampled to the original frame size, and fused with optical-flow information to localize possible abnormalities. In the rest of this paper we describe each part in detail.

Contributions: Our major contributions: *(i)* We introduce a novel Binary Quantization Layer, *(ii)* We propose a Tem-

poral CNN Pattern measure to represent motion in crowd, *(iii)* The proposed method is tested on the most common abnormality detection datasets and the results show that our approach is comparable with the state-of-the-art methods.

The rest of the paper is organized as follows: the Binary Quantization Layer is introduced in Sec. 2. In Sec. 3 we show the proposed measure, while our feature fusion is shown in Sec. 4. The experiments and a discussion on the obtained results is presented in Sec. 5.

2. Binary Fully Convolutional Net (BFCN)

In this section, we present the sequential Fully Convolutional Network (FCN) which creates the binary maps for each video frame. The proposed architecture contains two main modules: *1) the convolutional feature maps*, and *2) binary map representations of local features*. In the following, we describe each part in details.

2.1. Frame-based Fully Convolutional Network

Early layers of convolutions in deep nets present local information about the image, while deeper convolutional layers contain more global information. The last fully connected layers in a typical CNN represent high-level information and usually can be used for classification and recognition tasks. It has been shown that deep net models trained on the ImageNet [34] encode semantic information, thus can address a wide range of recognition problems [33, 8]. Since, FCNs do not contain fully-connected layers they preserve a relation between the input-image and the final feature-map coordinates. Hence a feature in the output map corresponds to a large receptive field of the input frame. Moreover, FCNs can process input images of different sizes and return feature maps of different sizes as output. In light of the above, this deep network typology is useful to both extract local and global information from an input image and to preserve spatial relations, which is a big advantage for a localization task.

Convolutional Feature maps: To tackle the gap between the raw-pixel representation of an image and its high-level information we choose the output of the last convolutional layer to extract feature maps. These components provide global information about the objects in the scene. To extract convolutional feature maps, we used a pre-trained AlexNet [13] model. AlexNet contains 5 convolutional layers and two fully connected layers. In order to obtain spatially localizable feature maps, we feed the output feature maps of the last convolutional layer into our binary quantization layer. Fig. 2 illustrates the layout of our network.

2.2. Binary Quantization Layer (BQL):

In order to generate a joint model for image segments there is a need to cluster feature components. Clustering the extracted high-dimensional feature maps comes with a high computational cost. The other problem with clustering is the need to know a priori the number of cluster centres. One possible approach to avoid extensive computational costs and to obtain reasonable efficiency is clustering high-dimensional features with a hashing technique to generate small binary codes [11]. A 24-bits binary code can address 2^{24} cluster centres, which is very difficult to be handled by common clustering methods. Moreover, this binary map can be simply represented as a 3-channels RGB image. Dealing with binary codes comes with a lower computational cost and a higher efficiency compared with other clustering methods. The other advantage of using a hashing technique in comparison with clustering is the ability of embedding the pre-trained hash function/weights as a layer inside the network.

Encoding feature maps to binary codes is done using Iterative Quantization Hashing (ITQ) [11], which is a hashing method for binary code unsupervised learning. Training ITQ is the only training cost in the proposed method, which is done only once on a subset of the train data. ITQ projects each high-dimensional feature vector into a binary space. We use the hashing weights, which are learned using ITQ, to build our proposed Binary Encoding Layer (denoted by $hconv6$). Specifically, inspired by [29, 16] we implement this layer as a set of convolutional filters (shown in different colors in Fig. 2), followed by a sigmoid activation function. The number of these filters is equal to the size of the binary code and the weights are pre-computed through ITQ. Finally, the binarization step has been done externally by thresholding the output of the sigmoid function.

Specifically, if $X = \{x_1, x_2, ..., x_n\}$ is a feature vector represented in $pool5$, the output of $hconv6$ is defined by $hconv6(X) = XW_i$, where $W_i = \{w_{i1}, w_{i2}, ..., w_{in}\}$ are the weights for the i^{th} neuron. The non-linearity is provided by a sigmoid function $\upsilon = \sigma(hconv6(X))$, and the threshold function is defined by:

$$g(\upsilon) = \begin{cases} 0, & \upsilon \leq 0.5 \\ 1, & \upsilon > 0.5 \end{cases} \quad (1)$$

Eventually, for any given frame our network returns a binary bitmap, which can be used for localization tasks.

Such a binary quantization layer can be plugged into the net as a pre-trained module, with the possibility of fine-tuning with back-propagation in an end-to-end fashion.However, in our abnormality task, due to the lack of data, fine-tuning is difficult and can be harmful because of possible overfitting, so all our experiments are obtained without fine-tuning.

Sequential BFCN: Let $\mathbf{v} = \{f_t\}_{t=1}^{T}$ be an input video, where f_t is the t-th frame of the video, and T is the number of frames. The frames of the video are fed into the network sequentially. The output for a single frame $f_t \in \mathbf{v}$, is an

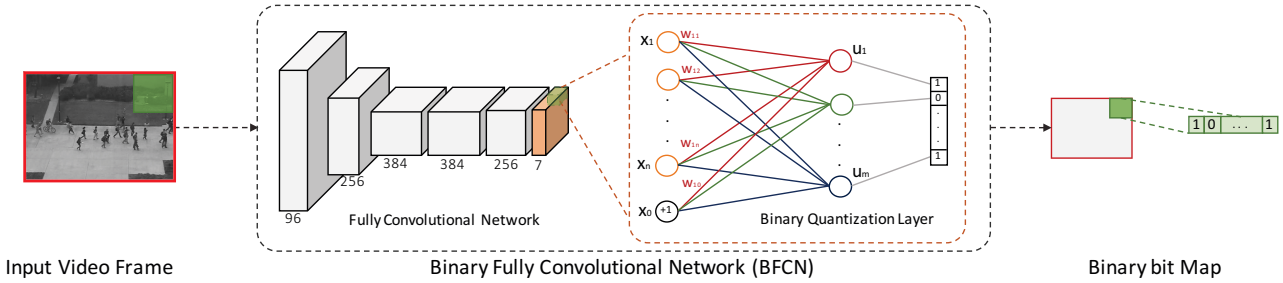


Figure 2. The Fully Convolutional Net with Binary Quantization Layer, is composed of a fully convolutional neural network followed by a Binary Quantization Layer (BQL). The BQL (shown in orange) is used to quantize the $pool5$ feature maps into 7-bit binary maps.

encoded binary bit map (prototype), denoted as m_t. All the binary maps are stacked and provided as the final representation of a given video, *i.e.*, $\mathbf{M} = \{m_t\}_{t=1}^{T}$.

3. Temporal CNN Pattern (TCP)

In this section we describe our proposed method to measure abnormality in a video sequence.

Overlapped Video Blocks: The first step is extracting video blocks from the video clip. As mentioned in Sec. 2, for a given video frame f_t the output of our FCN is a binary bit map denoted by m_t, which is smaller in size than the original image. In the binary bit map representation, each pixel describes a corresponding region in the original frame. This feature map partitions the video frame into a certain number of regions, which we called $patch$ denoted by p_t^i where t is the frame number and i is the i-th patch in the frame. b_t^i is a set of corresponding patches along consecutive frames. The length of a video blocks is fixed, and the middle patch of the video block is selected to indicate the entire video block. If the length of a video block b_t^i is $L+1$, it starts $L/2$ frames before the frame t and ends $L/2$ frames after that, namely $\{b_t^i\} = \{p_l^i\}_{l=t-L/2}^{t+L/2}$. To capture more fine information, the video block b_t^i has n frames overlapped with the next video block b_{t+1}^i.

Histogram of Binary Codes: A video block is represented with a set of binary codes (prototypes), which encode the similarity in appearance. We believe that observing the changes of prototypes over a time interval is a clue to discover motion patterns. Toward this purpose, for each video block b_t^i a histogram h_t^i is computed to represent the distribution of prototypes in the video block.

TCP Measure: Similarly to the commotion measure [24], to obtain the TCP measure for a given video block b_t^i, the *irregularity* of histogram h_t^i is computed. This is done by considering the fact that, if there is no difference in the appearance, then there is no change in descriptor features and consequently there is no change in the prototype representation. When the pattern of binary bits changes, it means that different appearances are observed in the video block and this information is used to capture motion patterns. The *irregularity* of the histogram is defined as the non-uniformity of the distribution in the video block. A uniform distribution of a histogram shows the presence of several visual patterns in a video block. The higher diversity of the prototypes on a video block leads to a low *irregularity* of the histogram. More uniform histograms increase the chance of abnormality. Such *irregularity* in appearance along the video blocks either is generated by noise or is the source of an anomaly. We took advantage of this fact to present our TCP measure.

The TCP measure for each video block b_t^i, is computed by summing over the differences between the prototype samples in h_t^i and the dominant prototype. The dominant prototype is defined as the most frequent binary code in the video block, which has the maximum value (mode) in the histogram h_t^i .

Let H^n represent the histogram of binary codes of all patches $\{p_t^i\}$ in the video block b_t^i denoted by $\{H^n\}_{n=1}^{N}$, where N is the number of patches in the video block. The aggregated histogram for block b_t^i compute as $\mathcal{H}_t^i = \sum_{n=1}^{N} H^n$. The aggregated histogram $\mathcal{H}_t^i$ represents the distribution of the appearance binary codes over the video block b_t^i, and the TCP measure compute as follows:

$$tcp(b_t^i) = \sum_{j=1}^{|\mathcal{H}_t^i|} ||\mathcal{H}_t^i(j) - \mathcal{H}_t^i(j_{max})||_2^2 \qquad (2)$$

where $|.|$ is the number of bins of the histogram, $||.||_2$ is the L2-norm, and the dominant appearance index over the video block is denoted by j_{max} (*i.e.*, the mode of $\mathcal{H}_t^i$).

TCP Map: To create a spatial map of the TCP measure c_t for any given frame f_t, the TCP measure is computed for all video blocks b_t^i, and we assign the value of c_t^i to a patch that is temporally located at the middle of the selected video block. The output c_t is a map with the same size as the binary map m_t which contains TCP measure values for each patch in the frame f_t. Finally, the TCP maps are extracted for the entire video footage. We denote the TCP map for frame f_t as $c_t = \{c_t^i\}_{i=1}^{I}$, where I is the number of patches in the frame.

Method	Ped1 (frame level)		Ped1 (pixel level)		Ped2 (frame level)	
	ERR	AUC	ERR	AUC	ERR	AUC
MPPCA [12]	40%	59.0%	81%	20.5%	30%	69.3%
Social force(SF) [21]	31%	67.5%	79%	19.7%	42%	55.6%
SF+MPPCA [18]	32%	68.8%	71%	21.3%	36%	61.3%
SR [5]	19%	—	54%	45.3%	—	—
MDT [18]	25%	81.8%	58%	44.1%	25%	82.9%
LSA [36]	16%	92.7%	—	—	—	—
Detection at 150fps [17]	15%	91.8%	43%	63.8%	—	—
AMDN (early fusion) [43]	22%	84.9%	47.1%	57.8%	24 %	81.5%
AMDN (late fusion) [43]	18%	89.1%	43.6%	62.1%	19 %	87.3%
AMDN (double fusion) [43]	16%	92.1%	40.1%	67.2%	17 %	**90.8%**
SL-HOF+FC [40]	18%	87.45%	**35%**	64.35%	19%	81.04%
Spatiotemporal Autoencoder [4]	12.5%	89.9%	—	—	**12%**	87.4%
Sparse Dictionaries with Saliency [44]	—	84.1%	—	—	—	80.9%
Compact Feature Sets [14]	21.15%	82%	39.7%	57%	19.2%	84%
Feng et al. [10]	15.1%	92.5%	64.9%	**69.9%**	—	—
Turchini et al. [39]	24%	78.1%	37%	62.2%	19%	80.7%
TCP (Proposed Method)	**8%**	**95.7%**	40.8%	64.5%	18%	88.4%

Table 1. Comparison with state-of-the-art on UCSD dataset: reported ERR (Equal Error Rate) and AUC (Area Under Curve). The values of previous methods are reported from [43].

Method	AUC
Optical-Flow [21]	0.84
SFM [21]	0.96
Del Giorno et al.[6]	0.910
Marsden et al. [19]	0.929
Singh and Mohan [37]	0.952
Sparse Reconstruction [5]	0.976
Commotion [24]	**0.988**
Yu et al. [44]	0.972
Cem et al. [7]	0.964
TCP (proposed method)	**0.988**

Table 2. Results on UMN dataset. The values of previous methods are reported from [24].

Up-sampling TCP Maps: Since the frame will pass through several convolution and pooling layers in the network, the final TCP map is smaller than the original video frame. To localize the exact region there is a need to produce a map of the same size as the input frame. For this reason, the TCP value in the map is assigned to all pixels in the corresponding patch of the frame on the up-sampled TCP map.

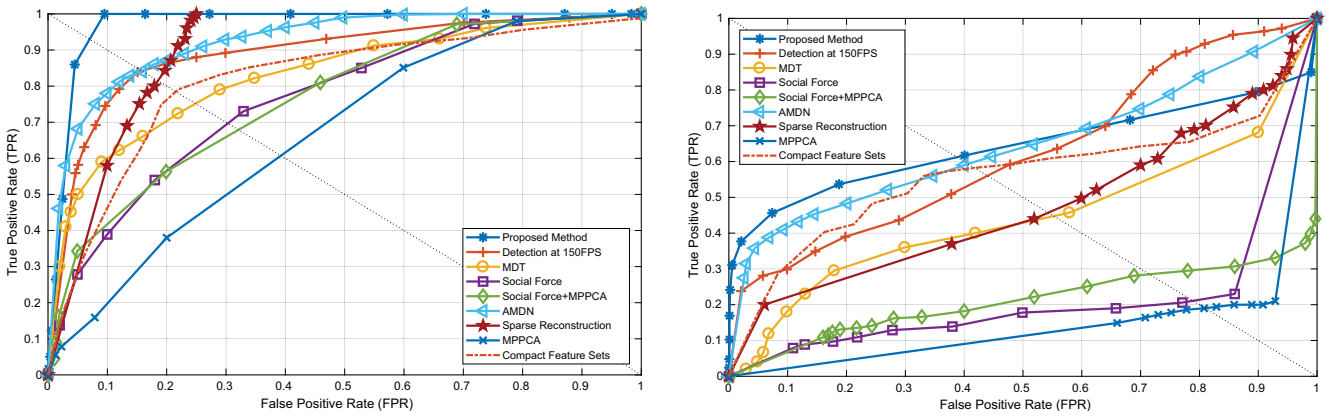
4. Fusion with optical-flow Maps

Since the Up-sampled TCP map can only detect the coarse region of abnormality, we propose to fuse optical-flow with the TCP maps in order to have a more accurate localization. The optical-flow [2] is extracted from each two consecutive frames. However the TCP map is computed for each L frames. To be able to fuse the optical-flow with the corresponding extracted TCP map, an aligned optical-flow map is constructed. Suppose that f_t and f_{t+1} are two consecutive frames from video $\mathbf{v} = \{f_t\}_{t=1}^{T}$, optical-flow map of_t, with the same resolution of an input frame, represents the optical-flow values transition between the two frames. optical-flow values are extracted for entire video footage $\mathbf{v}$ and stacked as optical-flow sequences $\{d_t\}_{t=1}^{T-1}$. Finally, similar to the overlapped video block extraction protocol, overlapped optical-flow maps are computed. If the length of a video block p_t^i is $L+1$, then the corresponding optical-flow map d_t^i is the sum of all optical-flow values over the corresponding i-th region as $d_t^i = \sum_{l=1}^{L} d_t^i(l)$. The optical-flow map for entire frame f_t is described as $d_t = \{d_t^i\}_{i=1}^{I}$.

Feature Fusion: The extracted optical-flow maps and the computed TCP maps for each video frame are fused together with importance factors α and β to create motion segment map: $mseg_t = \alpha d_t + \beta c_t$, $mseg = \{mseg_t\}_{t=1}^{T}$, where, $\{mseg\}$ is the extracted motion segments along the entire video $\mathbf{v}$. The importance factors indicates the influence of each fused map in the final segment motion map, we simply select 0.5 for both α and β.

5. Experimental Results

In this section, we evaluate our method over two well-known crowd abnormality datasets and compare our results with state of the art. The evaluation has been performed with both a *pixel-level* and a *frame-level* protocol, under



(a) ROC curve of frame level (anomaly detection)

(b) ROC curve of pixel level (anomaly localization)

Figure 3. Frame level and Pixel level comparison ROC curves of Ped1 (UCSD dataset).

standard setup. The rest of this section is dedicated to describing the evaluation datasets, the experimental setup and the reporting the results quantitatively and qualitatively.

Datasets and Experimental Setup: In order to evaluate our method two standard datasets: UCSD Anomaly Detection Dataset [18] and UMN SocialForce [21]. The **UCSD dataset** is split into two subsets *Ped1* and *Ped2*. *Ped1* contains 34/16 training/test sequences with frame resolution 238×158. Video sequences consist of 3,400 abnormal frame samples and 5,500 normal frames. *Ped2* includes 16/12 training/test video samples, with about 1,600 abnormal frames and 350 normal samples. This subset is captured from different scenes than *Ped1*, and the frames resolution is 360×240. This dataset is challenging due to different camera view points, low resolution, different types of moving objects across the scene, presence of one or more anomalies in the frames. The **UMN dataset** contains 11 video sequences in 3 different scenes, and 7700 frames in total. The resolution is 320×240. All sequences start with a normal scene and end with abnormality section.

In our experiments to initialize the weights of $hconv6$ an ITQ is applied on the train set of UCSD pedestrian dataset with a 7-bits binary code representation, which addresses 128 different appearance classes. Video frames are fed to the BFCN sequentially to extract binary bit maps. All video frames are resized to 460×350, then BFCN for any given frame returns a binary bit map with resolution 8×5, which splits the frame into a 40-region grid. The length of video block extracted from a binary map is fixed to $L = 14$ with 13 frames overlapping. The TCP measure is normalized over the entire video block sequence, then a threshold $th < 0.1$ is applied for detecting and subtracting the background region in the video.

Optical-flow feature maps are extracted to fuse with our computed features on the TCP measure maps. The fusion importance factor set to 0.5 equally for both feature sets. These motion segment maps are used to evaluate the performance of our method on detection and localization of anomalous motions during video frames.

5.1. Quantitative Evaluation

The evaluation is performed with two different levels: *frame level* for anomaly detection, and *pixel level* for anomaly localization. We evaluate our method on UCSD abnormality crowd dataset under the original setup [15].
Frame Level Anomaly Detection: This experiment aims at evaluating the performance of anomaly detection along the video clip. The criterion to detect a frame as abnormal is based on checking if the frame contains at least one abnormal patch. To evaluate the performances the detected frame is compared to ground truth frame label regardless of the location of the anomalous event. The procedure is applied over range of thresholds to build the ROC curve. We compare our method with state-of-the-art in detection performance on UCSD ped1 and ped2 datasets. The result is shown in Table 1, beside the ROC curves on Fig. 3.

The proposed method is also evaluated on UMN dataset. Fig. 4 shows the computed TCP for each frame illustrated as "detection signal" (green). We compared TCP with commotion measure (blue). The overall TCP value for a frame is computed from the sum of TCP measures over the patches in a frame and normalized in $[0, 1]$ as an abnormality indicator. In Fig. 4, the horizontal axis represents the time(s), the vertical axis shows the "abnormality indicator", and the light blue bars indicate the ground truth labels for abnormal frames.
Pixel Level Anomaly Localization: The goal of the pixel level evaluation is to measure the accuracy of anomalous event localization. Following [15], detected abnormal pixels are compared to pixel level groundtruth. A true positive

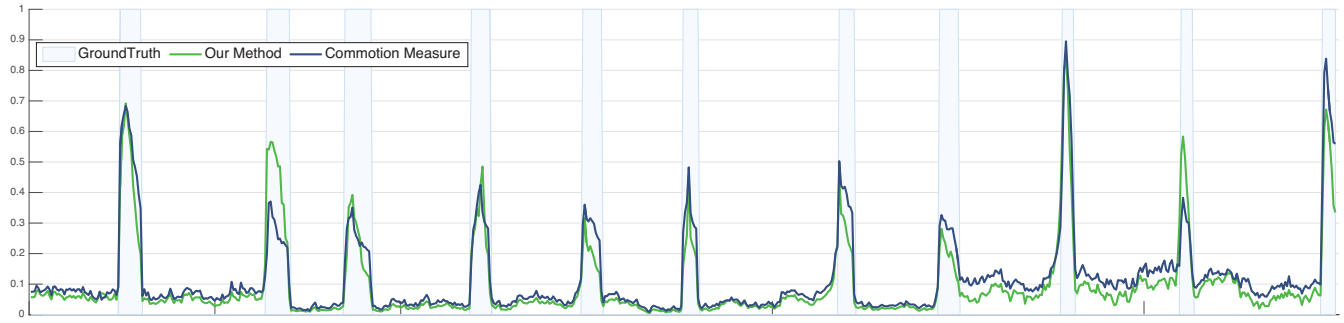


Figure 4. Frame-level anomaly detection results on UMN dataset: Our method compares to Commotion Measure [24]. The green and blue signals respectively show the computed TCP by our approach and Commotion Measure over frames of 11 video sequences. The light blue bars indicate the ground truth abnormal frames of each sequence. All the sequences start with normal frames and ends with abnormality.

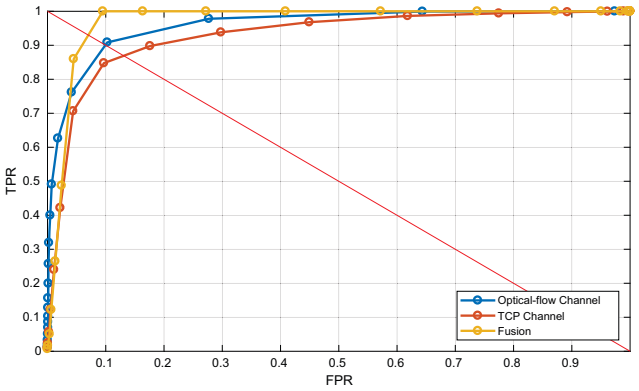


Figure 5. Comparison of different Streams in the proposed method on Ped1 frame-level evaluation.

prediction should cover at least 40% of true abnormal pixels over groundtruth, otherwise counted as a false positive detection.Fig. 3 shows the ROC curves of the localization accuracy over USDC Ped1 and Ped2. We compare our method with state of the art in accuracy for localization. Result is presented in Table 1.

In our experiments we observed that in most of the cases the proposed method hit the abnormality correctly in terms of detection and localization. Only in some cases our measure achieved slightly lower accuracy in anomaly localization and the anomaly detection performance in compare with the state of the art methods. Note that the proposed method is not taking advantage of any kind of learning in comparison with the others. The proposed method can be effectively exploited to detect and localize anomaly with no additional learning costs. Qualitative results on Ped1 and Ped2 are shown in Fig. 6. The figure shows we could successfully detect different abnormality sources (like cars, bicycles and skateboards) even in the case in which the object can not be recognized by visual appearance alone (*e.g.*, the skateboard). The last row in Fig. 6 shows the confusion cases, which not detect the abnormal object (the car) and detect normal as abnormal (the pedestrian). Most of the errors (e.g., miss-detections) are due to the fact that the abnormal object is very small or partially occluded (e.g., the skateboard in the rightmost image) and/or has a "normal" motion (i.e., a car moves the same speed of normally moving pedestrians in the scene).

5.2. Components Analysis

Analysis of the Importance of two Streams: The evaluation of the TCP-only version is performed on ped1 and in two levels: frame-level for anomaly detection, and pixel-level for anomaly localization. In the both cases we unchanged the same experimental setup reviewed in Sec. 5.

In the frame-level evaluation, the TCP-only version obtains 93.6%(AUC), which is slightly lower than 95.7% of the fused version. In pixel-level evaluation, however, the performance of the TCP-only version dropped 9.3% with respect to the fused version. This result is still significantly above most of the methods in Tab.1, but this clearly shows the importance of the optical-flow stream for abnormality localization. This is probably due to refining the abnormal segments leveraging the fine motion segments created by the optical-flow map. Hence, fusing appearance and motion can refine the detected area, which leads to a better localization accuracy. Fig. 5 shows ROCs for the three different states TCP-only, motion-only, and fusion. We simply select equal weights for optical-flow and TCP.

Binary Quantization Layer vs. Clustering: The Binary Quantization Layer ($hconv6$) is a key novelty of our method, which ensures the CNN will work both in the plug-and-play fashion as well as -possibly- being trained end-to-end. In order to evaluate the proposed binary quantization layer, the $hconv6$ removed from the network and a k-means clustering ($k = 2^7$) is performed on the $pool5$ layer of FCN in an offline fashion. Then, the TCP measure is computed on the codebook generated by clustering instead of the binary codes. We evaluated this on UCSD (ped1), obtaining

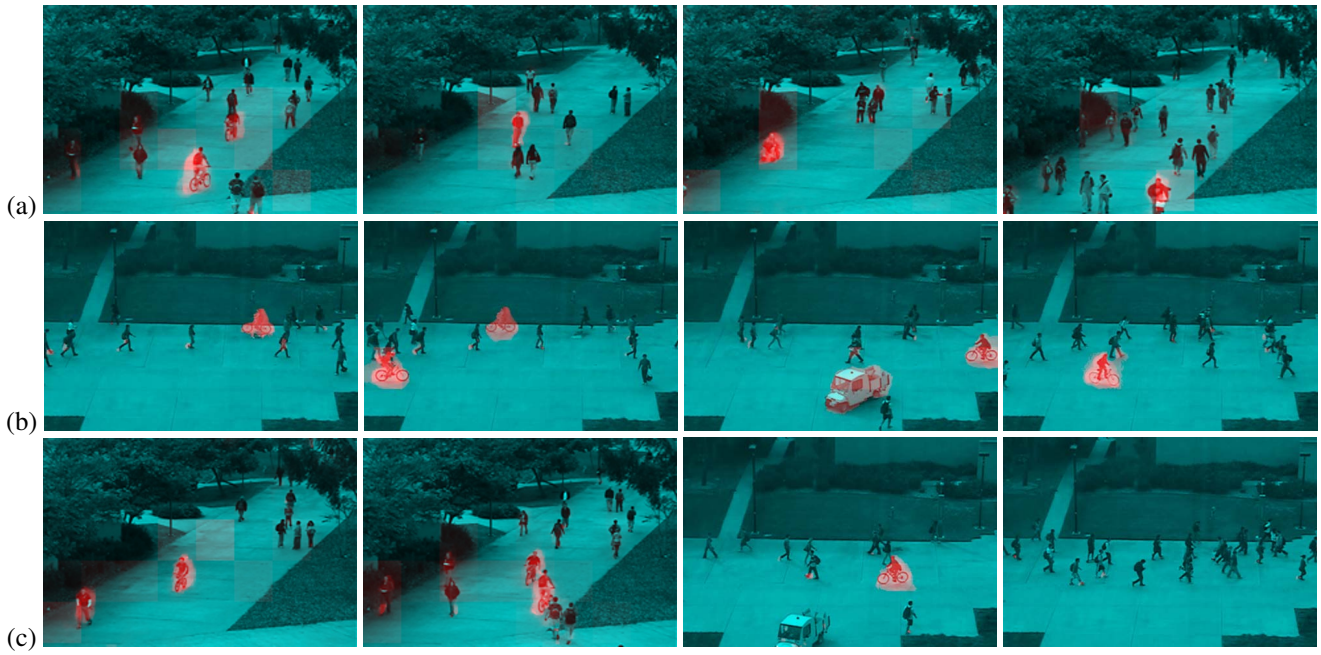


Figure 6. Sample results of anomaly localization on UCSD: (a) selected from Ped1, (b) Ped2, and (c) confusion cases from Ped1 and Ped2

78.4% (17.3% less than our best result on frame-level).

6. Discussion

The underlying idea of the proposed approach is to capture the crowd dynamics, by exploiting the temporal variations of CNN features. The CNN network is specifically used to narrow down the semantic gap between low-level pixels and high-level concepts in a crowd scene. The proposed approach provides the means to inject such semantics into model, while maintaining the method complexity in a manageable level. The proposed BFCN is composed of a fully convolutional neural network followed by a binary quantization layer (BQL). The weights of the former network are borrowed from an already pre-trained network and the weights of the BQL layer are obtained through an external hashing module and "plugged" into the network as an additional convolutional layer. The last layer of the network (BQL) provides the means to quantize the $pool5$ feature maps into 7-bit binary maps. The training of ITQ is done only once in an off-line fashion and is used for all the experiments without any further fine-tuning. The plug-and-play nature of our proposed architecture enables our method to work across multiple datasets without specific retraining.

The key idea behind this work is that the consecutive frames should have similar binary patterns, unless they undergo a large semantic change (e.g., abnormal object/motion). The role of TCP measure is to capture such changes across time by computing the irregularity over histogram of binary codes. The proposed TCP measure defines abnormal events as *irregular* events deviated from the normal ones, and the abnormality is measured as the uniformity of the histogram of binary codes. Hence, a flat histogram of binary patterns implies more inconsistency in visual patterns so increases the chance of abnormality. Such particular formulation allows to deal with the context-dependent abnormal events. These characteristics make our method unique in the panorama of the measure-based methods for abnormality detection.

7. Conclusions

In this work, we employed a Fully Convolutional Network as a pre-trained model and plugged an effective binary quantization layer as the final layer to the net. Our method provides both spatial consistency as well as low dimensional semantic embedding. We then introduced a simple yet effective unsupervised measure to capture temporal CNN patterns in video frames. We showed that combining this simple measure with traditional optical-flow provides us with the complementary information of both appearance and motion patterns. The qualitative and quantitative results on the challenging datasets show that our method is comparable to the state-of-the-art methods. As future work, we will study plugging a TCP measure layer and fine-tuning this layer with back-propagation. Moreover, exploring the use of [31, 32] as an alternative to Binary Fully Convolutional Net (BFCN) for end-to-end training of abnormality detection would be a potential direction.

References

[1] S. Amraee, A. Vafaei, K. Jamshidi, and P. Adibi. Anomaly detection and localization in crowded scenes using connected component analysis. *Multimedia Tools and Applications*, 2017.

[2] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *ECCV*, 2004.

[3] R. Chaker, Z. Al Aghbari, and I. N. Junejo. Social network model for crowd anomaly detection and localization. *Pattern Recognition*, 2017.

[4] Y. S. Chong and Y. H. Tay. Abnormal event detection in videos using spatiotemporal autoencoder. In *International Symposium on Neural Networks*, 2017.

[5] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. In *CVPR*, 2011.

[6] A. Del Giorno, J. A. Bagnell, and M. Hebert. A discriminative framework for anomaly detection in large videos. In *ECCV*, 2016.

[7] C. Direkoglu, M. Sah, and N. E. O'Connor. Abnormal crowd behavior detection using novel optical flow-based features. In *AVSS*, 2017.

[8] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *arXiv preprint arXiv:1310.1531*, 2013.

[9] R. Emonet, J. Varadarajan, and J.-M. Odobez. Multi-camera open space human activity discovery for anomaly detection. In *AVSS*, 2011.

[10] Y. Feng, Y. Yuan, and X. Lu. Learning deep event models for crowd anomaly detection. *Neurocomputing*, 2017.

[11] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*, 2011.

[12] J. Kim and K. Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *CVPR*, 2009.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[14] R. Leyva, V. Sanchez, and C.-T. Li. Video anomaly detection with compact feature sets for online performance. *TIP*, 2017.

[15] W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly detection and localization in crowded scenes. *PAMI*, 2014.

[16] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[17] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in matlab. In *ICCV*, 2013.

[18] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, 2010.

[19] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor. Holistic features for real-time crowd behaviour anomaly detection. In *ICIP*, 2016.

[20] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor. Resnetcrowd: A residual deep learning architecture for crowd counting, violent behaviour detection and crowd density level classification. In *AVSS*, 2017.

[21] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *CVPR*, 2009.

[22] H. Mousavi, S. Mohammadi, A. Perina, R. Chellali, and V. Murino. Analyzing tracklets for the detection of abnormal crowd behavior. In *WACV*, 2015.

[23] H. Mousavi, M. Nabi, H. K. Galoogahi, A. Perina, and V. Murino. Abnormality detection with improved histogram of oriented tracklets. In *ICIAP*, 2015.

[24] H. Mousavi, M. Nabi, H. Kiani, A. Perina, and V. Murino. Crowd motion monitoring using tracklet-based commotion measure. In *ICIP*, 2015.

[25] M. Nabi, A. Bue, and V. Murino. Temporal poselets for collective activity detection and recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013.

[26] H. Rabiee, J. Haddadnia, H. Mousavi, M. Nabi, V. Murino, and N. Sebe. Crowd behavior representation: an attribute-based approach. *SpringerPlus*, 2016.

[27] H. Rabiee, J. Haddadnia, H. Mousavi, M. Nabi, V. Murino, and N. Sebe. Emotion-based crowd representation for abnormality detection. *arXiv preprint arXiv:1607.07646*, 2016.

[28] R. Raghavendra, M. Cristani, A. Del Bue, E. Sangineto, and V. Murino. Anomaly detection in crowded scenes: A novel framework based on swarm optimization and social force modeling. In *Modeling, Simulation and Visual Analysis of Crowds*. 2013.

[29] M. Ravanbakhsh, H. Mousavi, M. Nabi, M. Rastegari, and C. Regazzoni. Cnn-aware binary map for general semantic segmentation. In *ICIP*, 2016.

[30] M. Ravanbakhsh, H. Mousavi, M. Rastegari, V. Murino, and L. S. Davis. Action recognition with image based cnn features. *arXiv preprint arXiv:1512.03980*, 2015.

[31] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. S. Regazzoni, and N. Sebe. Abnormal event detection in videos using generative adversarial nets. In *ICIP*, 2017.

[32] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. Sebe. Training adversarial discriminators for cross-channel abnormal event detection in crowds. *arXiv preprint arXiv:1706.07680*, 2017.

[33] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPRW*, 2014.

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

[35] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette. Fully convolutional neural network for fast anomaly detection in crowded scenes. *arXiv preprint arXiv:1609.00866*, 2016.

[36] V. Saligrama and Z. Chen. Video anomaly detection based on local statistical aggregates. In *CVPR*, 2012.

[37] D. Singh and C. K. Mohan. Graph formulation of video activities for abnormal activity recognition. *Pattern Recognition*, 2017.

[38] A. A. Sodemann, M. P. Ross, and B. J. Borghetti. A review of anomaly detection in automated surveillance. *TSMC, Part C*, 2012.

[39] F. Turchini, L. Seidenari, and A. Del Bimbo. Convex polytope ensembles for spatio-temporal anomaly detection. *ICIAP*, 2017.

[40] S. Wang, E. Zhu, J. Yin, and F. Porikli. Anomaly detection in crowded scenes by sl-hof descriptor and foreground classification. In *ICPR*, 2016.

[41] T. Xiang and S. Gong. Video behavior profiling for anomaly detection. *PAMI*, 2008.

[42] G. Xiong, J. Cheng, X. Wu, Y.-L. Chen, Y. Ou, and Y. Xu. An energy model approach to people counting for abnormal crowd behavior detection. *Neurocomputing*, 2012.

[43] D. Xu, Y. Yan, E. Ricci, and N. Sebe. Detecting anomalous events in videos by learning deep representations of appearance and motion. *CVIU*, 2016.

[44] Y. Yu, W. Shen, H. Huang, and Z. Zhang. Abnormal event detection in crowded scenes using two sparse dictionaries with saliency. *Journal of Electronic Imaging*, 2017.