Foreground Detection via Deep Variation Transformation

Yongxin Ge, Xinyu Ren, Chenqiu Zhao

***Abstract*—Previous approaches to foreground detection generally analyze the variation of pixel observations. In this paper, we focus on transforming the variation into another space where the entry of variation is easily classified, and a novel foreground detection method called Deep Pixel Variation Transformation Learning (DPVTL) is proposed. In particular, the pixel variation is represented by the sequence of pixel observations, which is used as the input of fully convolutional network(FCN) to find a transformation of pixels' variation. Then, the FCN is trained to learn the pattern of pixel variations for the transformation, followed by a liner classifier for labeling the pixels as foreground or background. Benefited from the ingenious utilization of deep learning network leading by our clear cognition of essence about foreground detection problem, the proposed approach adaptively generates superior performances in diverse nature scenes. Comprehensive experiments in several standard benchmarks demonstrate the superiority of the proposed approach compared with state-of-the-art methods including both deep learning and traditional methods.**

***Index Terms*—foreground detection, Feature Transformation, Deep Learning,**

I. Introduction

Foreground detection as a fundamental problem in computer vision [1] has been discussed over decades with the increasing number of cameras, which is widely used in the applications of video processing [2]. Typically, it is recognized as a binary classification task that assigns each pixel in the video stream with a label, for either belonging to the background or foreground scene. Traditional foreground detection algorithms focus on analyzing the pixel variation, establishing background models with statistical methods such as GMM [3] [4] and KDE [5]. However, due to the unpredictability and rapidity of the pixels' variation in natural scenes, the variation becomes so unordered which is hard to be analyzed for foreground detection. Therefore, the foreground detection is still a challenging problem in complex natural scenes.

In the diversely natural scenes, it is possible that moving objects produce the similar or even the same observations of pixels to that of background. As shown in the Fig. 1, the observation C is closely related to the observations which belong to the background. However, it is actually produced by moving objects and should be classified as foreground. Unfortunately, in most cases, it is highly possible that the observation C will be falsely classified due to the similarity with their counterparts of background. In this work, we focus on learning the pattern of pixels' variation and transforming the variations into a new space where the observations are easier to classify, rather than learning a classifier for individual pixels. As shown in the bottom part of the Fig. 1, the pattern of fragment consists of observations A-D can be learned by the network and transformed into another fragment where these observations are easily and correctly classified as foreground. Based on this, the Deep Variation Transformation model for foreground detection is proposed.

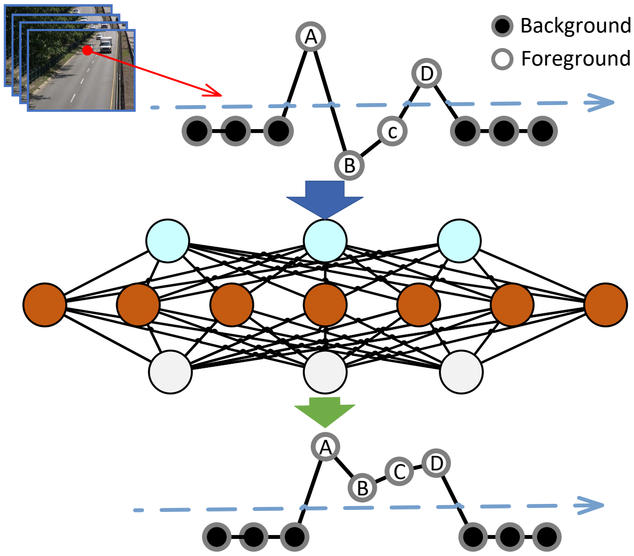

Fig. 1. The demonstration of deep variation transformation. Due to the complexity of natural scenes, the original pixels' variation is hard to classify correctly. After transforming by deep learning network, the pixels in variation become easy to be classified as foreground and background correctly.

In the DVTL model, the sequence of pixel observations is used to represent the variation and input into the network for learning, which encode both the intensity distribution and sequential information. Then, a Fully Convolutional Network (FCN) [6] is applied to learn patterns of the pixel variation and find a transformation which guarantees the linear separability of pixels in the transformed variation. In particular, we take advantage of the strong learning ability of the deep neural network to learn an end-to-end representation of the pixel variation in a new space where they can be easily classified to background and foreground. Benefited from the innovative framework of variation transformation, proposed approach works well in diversely complex scenes adaptively.

The rest part of this paper is organized as follows: In Section II, we give a brief discussion of the early and recent relevant works. The details of the variation transformation are presented in Section III. The architecture of network is introduced in Section IV, followed by experiments and result comparison in Section V and conclude the paper in Section VI.

II. Related Work

Over the last few decades, a huge number of foreground detection methods have been proposed, which are broadly categorized into pixel-based, region-based and learning-based methods.

A. pixel-based methods

Pixel-based methods usually assume the independence between neighboring pixels, and utilize the low-level features, such as color or gradients for background subtraction.

In particular, the Gaussian Mixture Model (GMM) proposed by Stauffer et al. [3] is the most popular approach among pixels-based methods [7]. It utilizes a mixture of weighted Gaussians to model the probability distribution of each pixel in time sequence. Pixels are considered as background if there exists a Gaussian function includes their values with sufficient evidence. Zivkovic et al. [8] improve the GMM method with the utilization of the recursive equation to automatically update the parameters and adjust the needed number of components of mixture for each pixel. Kim et al. [9] present the codebook method, which records the sampling background values to codewords for each pixel position. The incoming pixels are compared with these codewords to see if their distances lie within a certain bound. In addition, a non-parametric background model is proposed by Elgammal et al. [5]. They assume that each background pixel is drawn from a probability distribution function, which is estimated with Kernel Density Estimation(KDE). Another non-parametric method is proposed by Barnich et al. [2], called the Visual Background Extractor (ViBe). The background model of ViBe consists of pixel samples from the video stream. Each pixel in the current frame is compared with sampling pixels from the corresponding background model and labelled as the foreground when there exist sufficient samples with a distance to itself within a certain range. To adaptively update the parameters, Hofmann et al. [10] improve the ViBe by presenting an adaptive threshold, which depended on the pixel position and a background dynamics metric.

Unfortunately, the pixel-based methods ignore the spatial-temporal information due to their assumption of independence between pixels. But there is, in fact, a strong coherence in image sequences that contains abundant hidden clues for the background model. To address this shortcoming, we introduce a framework of variation transformation learning, where pixel's historical observations are embedded in a piece of pixel patch and sorted in chronological order as a whole for the training of our DPVTL model. Bringing together the historical observations ensures the data integrity and preserves the temporal coherence of our training data. On the other hand, the application of neural network guarantees the strong learning ability of our model. Thus, the proposed approach is more capable of learning the patterns of pixel variation and exploiting the spatial-temporal context, compared to those classifiers based on the assumption of pixel independence.

B. region-based approach

Region-based approaches are usually performed at block-level resolution, in order to exploit the spatial context between neighboring pixels.

Varadarajan et al. [11] proposed a region-based GMM model, which is derived from expectation-maximization (EM) theory with the consideration of neighboring pixels. In addition, Chen et al. [12] combine the GMM with constrains of temporal and spatial information from the optical flow and hierarchical superpixels. Moreover, Sheikh et al. [13] introduce a framework based on the Markov Random Field modeling with Maximum A Posteriori probability (MAP-MRF) estimation, which incorporate the pixel location into background and foreground KDEs for the detection based on spatial context. Similarly, in [14], original images are divided into overlapping blocks. Each block is sequentially processed by an adaptive multi-stage classifier, which consists of a likelihood evaluation, an illumination invariant measure and a temporal correlation check. Hence, Izadi et al. [15] present a robust region-based approach, which generates a pair of foreground maps based on gradient and color respectively. Any foreground region that does not exist in the first foreground map could be recovered from the other one.

In contrast, we accept the assumption that neighboring blocks of background pixels should follow similar variations over time, and combine the pixel variation with its spatial neighbors to revise our prediction. Due to the application of deep learning, the proposed approach is more powerful in capturing the structural background variation and achieves significant improvement compared to its competitors.

C. Machine Learning based Methods

The last category of background subtraction methods applies traditional machine learning and deep learning for the foreground detection.

Traditional machine learning methods are commonly involved with support vector machines (SVM) [16] and Bayesian methods [17]. For example, Han et al. [16] integrate gradient, color, and Haar-like features to address the spatiotemporal variations for each pixel. Their background model is obtained for each feature in a kernel density framework and a SVM is employed for classification.

Recent years, deep learning methods begin to flourish in several computer vision fields. A novel approach for foreground detection with the use of the Convolutional Neural Network(CNN) is proposed by Wang et al. [18]. They utilize a CNN with a cascade network architecture for segmentation in foreground detection, which performs excellently with sufficient training data. Braham et al. [19] employ a scene-specific CNN, which is trained with corresponding image patches from the background image, video frames and groundtruth. In particular, the background image is obtained by temporal median filtering, and the groundtruth can be replaced with segmentation results from other foreground detection methods. A similar approach is presented by M. Babaee et al. [20]. Their background images combine the segmentation mask from

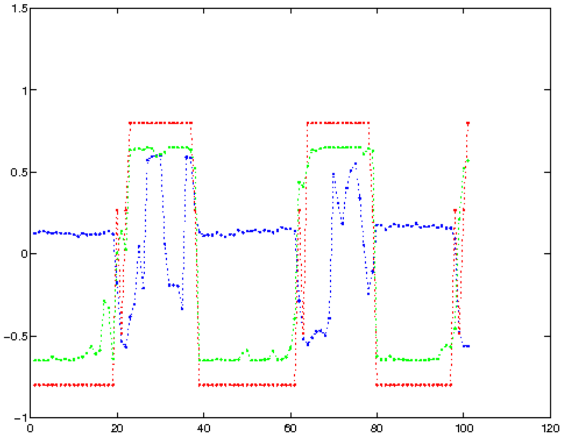

Fig. 2. Comparison of the original pixel variation, groundtruth and transformed variation. They are represented by the blue line, red line and green line respectively.

SuBSENSE [21] algorithm and the output of Flux Tensor algorithm [22], which is able to adaptively update the parameters used in the background model. They also utilize spatial-median filtering as the post processing of the network predictions. In [23], a fully convolutional network with the skip architecture is proposed. The authors also utilize a temporal approach to sample training images from the given video, thus providing the background model with limited temporal information.

Some breakthroughs and progress have been achieved by applying those methods, especially deep learning based approaches with optimized network architectures. However, due to the diversity of nature scenes, pixel variations can be very complex, leading to difficulty in analyzing specific pixel variation patterns. Unlike existing ones, our method focus on creating a transformation, which guarantees the linear separability of pixel variations in the mapping space, rather than building classifiers for individual observations. Besides, our network is trained to learn patterns of pixel variations, which promise a better performance in time series modeling on temporal coherence.

III. VARIATION TRANSFORMATION

In this section, we will elaborate on the proposed approach and how does the variation transformation work in foreground detection.

Foreground detection is essentially a binary classification in time sequence, where pixels in a video stream are classified as two categories: foreground (cars, people or animals) and background (roads, trees or other backdrops). In most cases, if we print out the historical observations of a single pixel, it is not hard to see that they usually keep their stability when belonging to the background. However, there are some exceptions: historical observations vary regularly when belonging to some dynamic background scenes(e.g. waves, swaying tree leaves). Generally, pixels from the background are sharing some common patterns of variation.

To give an intuitive understanding of the pixel variation, we plot the historical observations and the corresponding groundtruth of a single pixel in Fig. 2. As we can see, there are several lines in three colors: blue, red and green. The blue line consists of original observations, representing a piece of pixel variation. Accordingly, the red line is made of the groundtruth data, corresponding to the observations from the blue one. And the green line consists of the outputs of the proposed approach which represent the transformed variation.

Previous methods have a good performance when background and foreground observations are remarkably different and the backgrounds are normally keeping static. While in fact, backgrounds can be rather turbulent and dynamic due to the complexity and diversity of nature scenes. Especially when illumination and camouflages are involved, background and foreground observations are easily confused and mixed up.

For example, in Fig. 2, it is noticeable that some foreground observations in blue line share the same or similarly value with background ones. In that case, those popular methods will inevitably yield some sticky moments when separating the pixels in the blue line. Statistical methods, for instance, are no longer valid, because they only focus on establishing a statistical model for the background, while having little or no concern for the temporal coherence of these observations. In other words, despite knowing that pixels from the background share some common patterns of variation in a temporal sequence, they still let the order information of sequential images all go to waste.

In this paper, a novel framework of pixel variation transformation is proposed.

Different from existing methods, we decide to shift our attention from single pixel classification to variation transformation. More specifically, a FCN is employed to learn patterns of pixel variations and mapping them into a new space where they are close to the groundtruth data. The transformed variation is presented by the green line in Fig. 2. The benefits are evident and clearly seen. It is hard to distinguish a foreground observation when its value is similar to the background ones. Classification on the transformed variation, however, is much more convenient and intuitive. With the strong learning ability of the FCN, we are able to learn the diverse variation patterns of background and foreground over a long period of time. And the variation patterns, in turn, contribute to the variation transformation where the value of each observation is adjusted according to its compatriot in time sequence.

IV. FOREGROUND DETECTION VIA DEEP VARIATION TRANSFORMATION

In this section, we will introduce the details of proposed approaches. We explain the production of the pixel patch, which is a specific form of the pixel variation, and the architecture of our network.

The flow chart of the proposed approach is illustrated in Fig. 3. Firstly, we sample the input and ground truth images to generalize the pixel variations, and reshape them into patches and feed them into the network with its spatial neighbors.

After reassembling the patches into the complete output frame, it is post-processed, yielding the final segmentation of the respective video frame.

Given different videos, their numbers of frames are generally different. In order to keep each piece of variation the same length, image stacks are sampled from the given videos before the pixel variations are extracted. Let denote the frames sequence as:

$$\mathcal{I} = \{I_1, I_2, \cdots, I_T\}, \tag{1}$$

And image stacks can be defined as follows:

$$\mathcal{I}^d = \mathcal{D}(\mathcal{I}) = \{I_1, I_{[2\cdot\frac{T}{N}]}, \cdots, I_{[(N-1)\cdot\frac{T}{N}]}, \} \tag{2}$$

where I_t is the frame t of the given video. T and N represent respectively the totall number of the video frames and the number of image stacks. For each video, we can produce multiple image stacks and choose one of them for the training. In addition, we can get compact pixel variations, which contain more temporal information than the continuous pixel sequences, by using equal interval sampling.

After the sampling process, a large number of observation sequences, or pixel variations, can be extracted from the chosen stack at each pixel position. Each of the pixel variation is a piece of temporal information which containing the changing patterns of background pixels. And each of them is of fixed length $\cdot\frac{T}{N}$. It is notable that observation sequences are extracted from individual pixel positions, which promises that sufficient training data can be obtained with only one image stack.

Our task is to provide an end-to-end transformation of pixel variations, based on the strong learning ability of FCNs. However, vectors are not appropriate for the network training and learning. Besides, we also hope pixel observations can be interacted with their further compatriots in temporal sequence. Thus we reshape the variations into pixel patches as the input of our network. A sample of the pixel patch is like this:

$$S_{x,y}(m,n) = \mathcal{C}(\mathcal{I}^d, R) = \mathcal{I}_{m\times R+n}(x,y), \tag{3}$$

where the pixel patch is denoted as $S_{x,y}(m,n)$, at the pixel positionthe pixel position (x,y). And the parameter R is the column and row of pixel patches. the column and row of the pixel patch. Observations are put into the $R \times R$ patch from top to bottom and left to right. To put it another way, the pixel patch is just a specific form of the pixel variation. Since the pixel sequences are extracted from each pixel position, a large number of pixel patches can be extracted from only one single image stack, which promises sufficient training data for our network.

It is generally accepted that neighboring background pixels share a similar temporal distribution. In order to benefit from the spatial context, we concatenate the pixel patch with 2 neighboring pixel patches, which are randomly selected in the 8-connected neighborhood of each pixel position. The input patch of our network is defined as:

$$S_{x,y}(m,n) = \mathcal{C}o(S_{x,y}(m,n), S_{x',y'}(m,n), S_{x'',y''}(m,n)), \tag{4}$$

where the $\mathcal{C}o$ represents the concatenation process on the third dimension. And neighboring patches are at pixel position (x',y') and (x'',y'') respectively.

Correspondingly, the groundtruth patches are obtained in a similar way as the input ones, except the concatenation process, as shown in the Fig. 3. Both of them are fed into the network for training to find a end-to-end transformation at each pixel position. However, the size of feature maps will shrink quickly during forward computation. In preprocessing, we borrow the idea from Image Semantic Segmentation [6], padding the input patches before the training. After the forward computing, observations in input patches are mapped into a new space where they can be easily classified by thresholding. The transformed pixel patch is defined as:

$$\begin{aligned} S'(m,n) &= f_{\theta_N}(f_{\theta_{N-1}}(\cdots f_{\theta_1}(Pad(S(m,n)))\cdots)) \\ &= f^n_{\prod_{n=1}^N \theta_n}(S(m,n)), \end{aligned} \tag{5}$$

where, $S'(m,n)$ is the prediction of the network, and the forward computation at each layer of the network is represented by f_{θ_N}. The symmetric padding process is denoted as Pad, where we have lost none of the pixel information to make the output $S'(m,n)$ the same size as the pixel patch $S(m,n)$. Another advantage of symmetric padding is that the order information still remains.

For the loss function, we choose the Sigmoid Cross Entropy (SCE), which are helpful to address the learning slowdown. The formulation is as follows:

$$\begin{aligned} \mathcal{L}(S'(x,y), S^{gt}(x,y)) = \sum_{x=1}^{m}\sum_{y=1}^{n} S^{gt}(x,y)log(Sig(S'(x,y))) \\ + (1 - S^{gt}(x,y))log(1 - Sig(S'(x,y))), \end{aligned} \tag{6}$$

$$Sig(x) = \frac{1}{1+e^{-x}}, \tag{7}$$

where $S^{gt}(x,y)$ denotes the groundtruth patch which is given by the corresponding groundtruth stack, and Sig represents the sigmoid function. The SCE is calculated between the transformed patches and the corresponding groundtruth patches. Boundaries of moving objects and pixels that out of the region of interest are not taken into account in the cost function. Finally, we get the transformed variation through the FCN, which are a lot easier for classification.

As shown in the Fig. 3, the prediction maps can be reconstructed from the prediction patches. Then a global threshold T_r is set for each transformed observation in order to map them to $\{0,1\}$. The threshold function is given by

$$g(x,y) = \begin{cases} 1, & x < y \\ 0, & otherwise \end{cases}, \tag{8}$$

$$\mathcal{M}(x,y) = g(T_f, \mathcal{C}^{-1}(S'(x,y), R)), \tag{9}$$

where the M denotes the final foreground map given by our DPVTL model. After the thresholding calculations, our experiment results show that a random initialized FCNs, trained end-to-end on feature learning can achieve the state-of-the-art without further machinery.

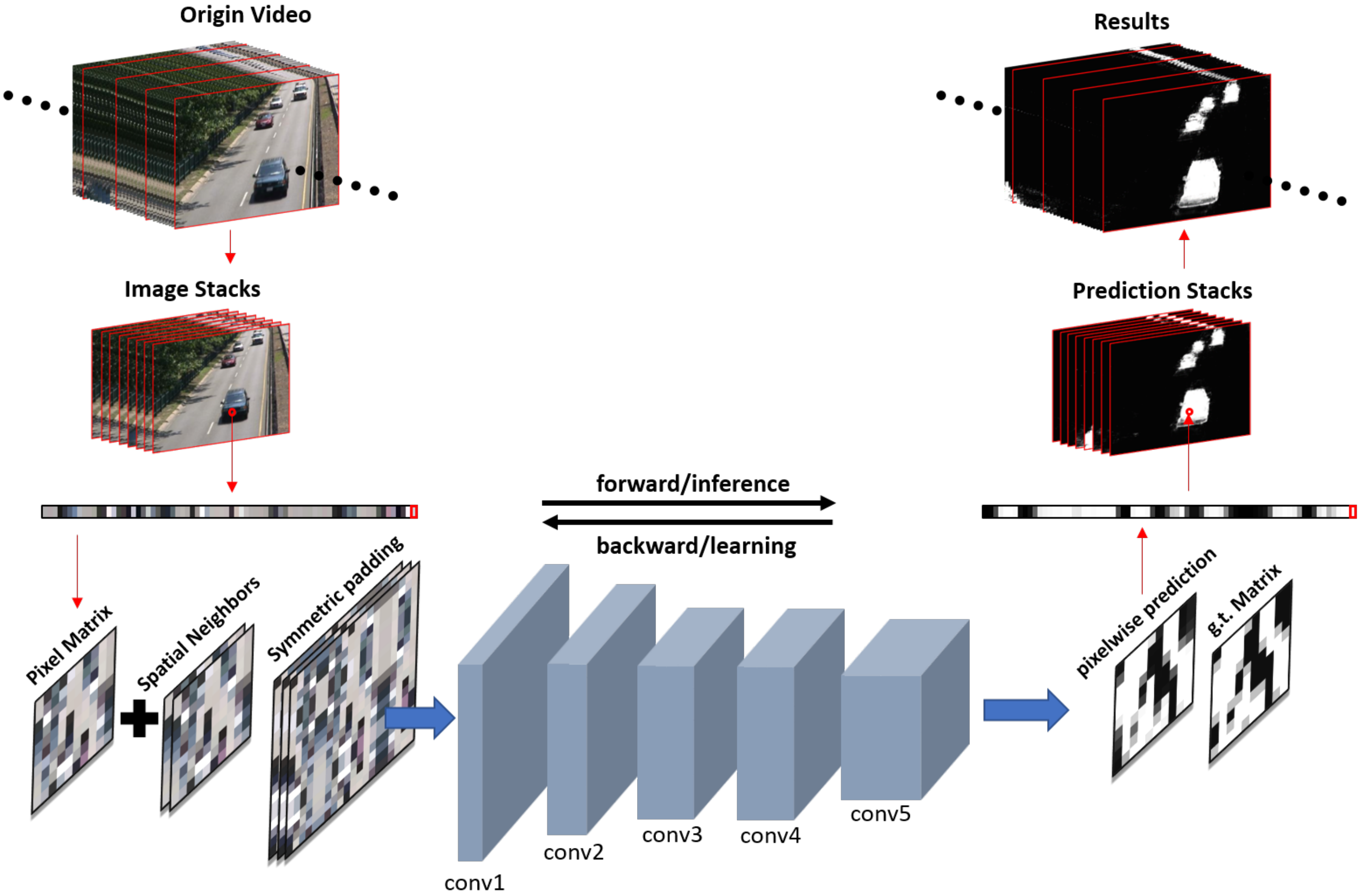

Fig. 3. Original videos break down into pixel patches for the training of the FCN, which Fully convolutional network produces an efficient machine for end-to-end transformation learning.

Here we make a detailed introduction to the deep learning model we used and explain why we choose it.

Different with CNNs, FCNs utilize convolutional layers with 1×1 kernels to take the place of fully connected layers, which provide several benefits. First and foremost, the size of output is adjustable in FCNs, which allows an end-to-end mapping of input patches and prediction patches at each pixel position. We hope to determine the label of an observation through the comparison of its compatriot in time sequence. There is one more point I ought to touch on, that since the feature maps are no longer needed to be converted into vectors, spatial information can be retained. The last but not the least, FCNs have been used in sematic segmentation and researchers found FCNs have a strong learning ability which won't lose to the traditional ones. Meanwhile, it's also a highly efficient computation model.

Based on above-mentioned factors, FCN is designed as the alternative network architecture in this paper. The structure of our FCN for foreground detection is shown in Fig. 3. The proposed FCN contains 6 convolutional layers, and the last convolutional layer has a filter size of 1×1. We use the Rectified Linear Unit (ReLU) as the activation function after each convolutional layer and the Sigmoid function after the last fully connected layer. We do not use any other tricks in our network training and the experiment results prove that the proposed approach is feasible and very effective.

V. Experiments

In this section, we ran comprehensive experiments to evaluate the performance of the proposed approach on the CDnet 2014 benchmark [24] and CAMO-UOW [25]. The CDnet is the largest dataset for foreground detection so far as we are aware, containing 11 categories with several complexly challenging scenes, such as Dynamic Background, Camera Jitter, Shadow, Night Videos, PTZ and so on. The CAMO-UOW is a challenging benchmark for camouflaged foreground detection, which contains 10 high resolution videos. For each video, one or two persons appear in the scene with the clothes in the similar color as the background.

The proposed approach is compared with several existing traditional state-of-the-art foreground detection algorithms, including the IUTIS-5 [26], the SuBSENSE [21], the WeSamBE [27], sharable GMM the SharedModel [28], word-dictionaries-based method the PAWCS [29], the SemanticBGS [30], the AAPSA [31], etc. Moreover, two deep learning based algorithms are also compared with the proposed approach, which include DeepBS [20], and DBMF [23]. All the results of compared algorithms are provided by authors.

During the comparison, the F-measure(Fm) has been used for evaluation. The Fm is a general international standard in foreground detection which measures the segmentation accuracy by considering both the recall and the precision. The definition of Fm is shown as follows:

$$Fm = \frac{2 \times precision \times recall}{precision + recall} = \frac{2TP}{2TP + FN + FP}, \quad (10)$$

where TP, FP, and FN are true positives, false positives, and false negatives respectively, computed in pixels of all test frames for each video.

The quantitative and qualitative comparisons are shown in Table I and Fig. 4 respectively. Due to the paper length, several typical videos are selected for the qualitative comparisons as well as the discussion. In the dynamic background scene, the video ”canoe” is a typically challenging video which includes a large area of water rippling. The main challenge comes from the dynamic background, in which it is so hard to describe the background by a single image. In this condition, since the traditional foreground detection method such as the SharedModel and the WeSamBE do not have enough ability to describe the complex dynamically background, they are fail to detect the people on the boat, as shown in the Fig. 4. Besides, the detected moving objects of the SharedModel are not accurate in the boundary due to the utilization of texture features. In contrast, benefited from the strong learning ability of Deep Learning network, the DeepBS successfully detected the people. Unfortunately, since the DeepBS ignores the fact that a single background is not enough to describe the dynamic background, even the deep learning based algorithm is suffering from the detection of the boat shape. In contrast, the proposed approach performed superior than others in this scene, since the essence of foreground detection is considered as a binary classification of pixels' observation in time sequence. Based on this insight, the FCN network focuses on learning the patterns of the pixel variation rather than a static background image, and proposed approach achieves promising performance in the canoe video.

As for the case of shadow scene, ”peopleInShade” is a typical example with prevalent hard and soft shadows. In the traditional approaches, these shadow regions are usually segmented as foreground since it is also moving with the objects. Therefore, traditional methods like the PAWCS, the WeSamBE and the SharedModel falsely segments part of shadows as moving objects. In addition, the foreground provided by the IUTIS-5 is incomplete on the part of the pedestrian's body due to the interference of shades, which can be owing to the severe dependency of texture features. Whereas, the DeepBS performs well in this video benefited from the utilization of CNN. However, the shape of pedestrians are slightly deformed as the result of their patch-wise processing of CNN. In contrast, derived from the fact that our DPVL focus on learning the pattern of pixels' variation in the shadow regions, proposed approach successfully segments the shadow part as the background and achieves the highest performance in the category of shadow scene.

In the video ”corridor” among the Thermal scene, there is no color information since the videos are obtained through a Thermal camera. Moreover, moving objects in these videos are exceedingly fuzzy and indistinct, which is the main challenge of this category. The WeSamBE, the SharedModel and the PAWCS successfully detect the target objects, owing to a stable background in this indoor video. However, they fail to remove the reflections since their modeling ability has already reached a limit under the extreme condition of thermal map. The DeepBS, by contrast, succeeds in eliminating most of the reflections. Meanwhile, moving objects are also clearly divided from the background thanks to the strong modeling ability of CNN. However, due to the dependency of the edge feature, a small object were missed in the detection result. Fortunately, the proposed approach focus on the pattern of pixels' variation, which should be theoretically effective even in the observation without the color information. Consequently, our DPVL performed much better than compared algorithms, with the situation that most parts of the shadow are removed and the segmentation results are more accurate.

The quantitative evaluation of the proposed approach on CDnet 2014 is shown in the Table I. It can be inferred that the proposed approach significantly outperformed all of the compared state-of-the-art algorithms in most of complex scenes and achieved 11.37% gain in FM over the second one on the whole dataset. Moreover, in order to compare the proposed approach with the DBMF, which is also based on deep learning and only publish their results in specific videos. The proposed approach has also ran in these videos and the results are shown in Table 2. Again, the proposed approach has noticeably better performance than the DBMF and some other classical background approaches.

As shown in the Table I and Table II, previous deep learning based methods like the DeepBS and the DBMF achieve well performance. From our own perspective, that good performance should attribute to the stronger modeling ability and learning adaptation of CNN and FCN. However, the proposed approach focused on the pixels variation in temporal sequence rather than low-level static features such as color, edges and textures, which give us the ability to avoid the shortcomings of the background models. Consequently, the proposed approach still gets considerably better results, which over 17.85% in FM metrics compared with the DeepBS and over 9.50% compared with the DBMF.

The evaluation of the proposed approach in CAMO-UOW dataset is shown in the Table III. Unlike the CDnet dataset, the videos of CAMO-UOW dataset are specially proposed for moving objects with camouflage, which is the main challenge of this dataset. As shown in the Table III proposed approach achieves better performance compared to its competitions, with an average F-measure of 0.97, compared to values between 0.77 and 0.94 for the other methods. Therefore, it is fair to say that the proposed approach performs better compared with their peers.

In this dataset, target objects have similar color and textures with the background, which brings a lot of difficulties and obstacles to traditional methods. However, our FCN is a powerful Neural Network model which is good at capturing

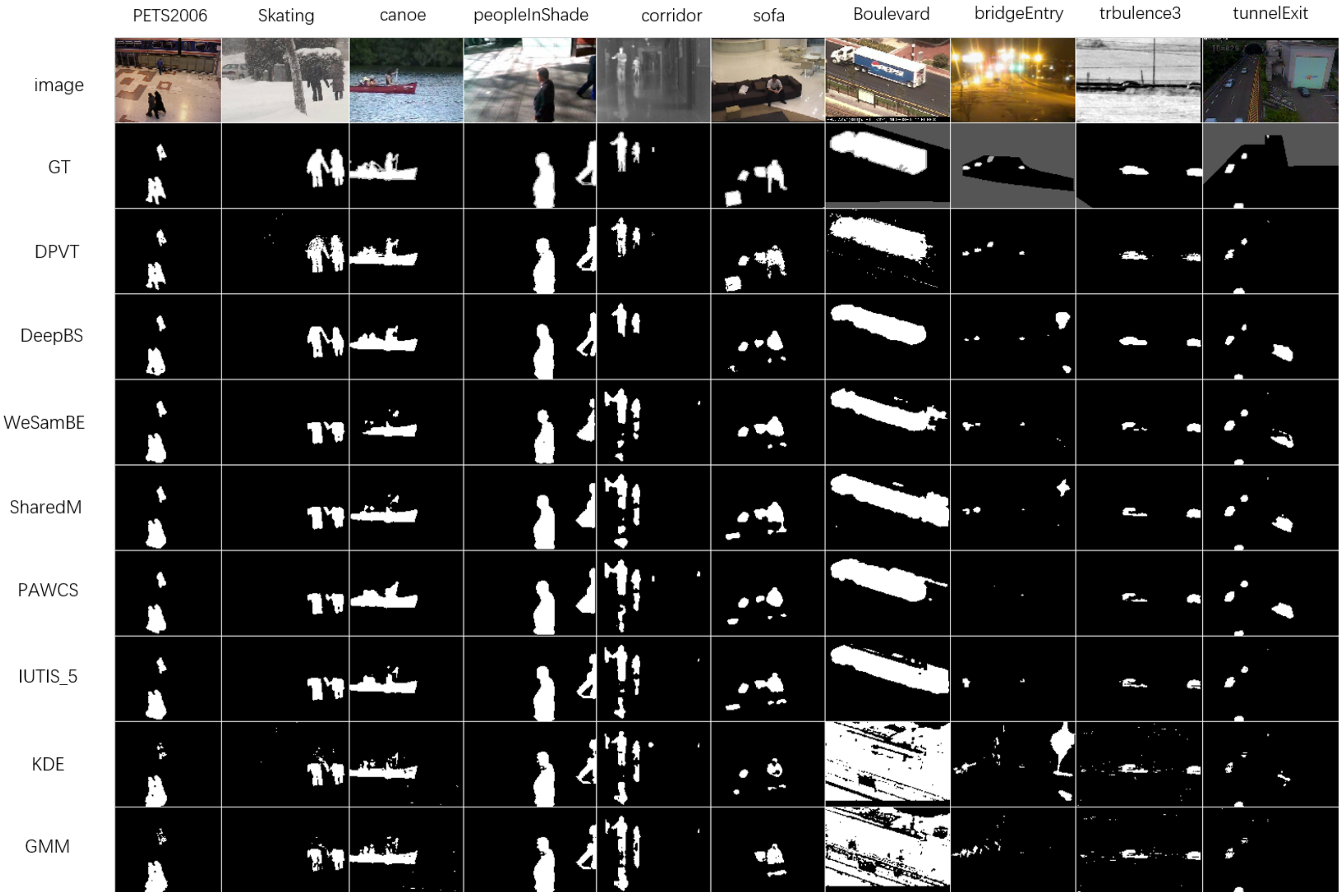

Fig. 4. The qualitative evaluation of the proposed method. All the results is followed in the CDnet 2014.

TABLE I
THE PERFORMANCE COMPARISON OF THE PROPOSED APPROACH AND SOME STATE-OF-THE-ART ALGORITHMS ON THE VIDEO SEQUENCES FROM DIFFERENT CATEGORIES IN CDNET 2014. FOR EACH VIDEO, 100 FRAMES WERE TAKEN FOR THE TRAINING OF OUR FCN.

Videos	baseline	dyna.bg	cam.jitter	int.obj.m	shadow	thermal	bad.weat	low f.rate	night vid.	PTZ	turbul.	overall
DeepBS [20]	0.9580	0.8761	0.8990	0.6097	0.9304	0.7583	0.8647	0.5900	0.6359	0.3306	0.8993	0.7458
IUTIS-5 [26]	0.9567	0.8902	0.8332	0.7296	0.9084	0.8303	0.8289	**0.7911**	0.5132	0.4703	0.8507	0.7717
FTSG [22]	0.9330	0.8792	0.7513	0.7891	0.8832	0.7768	0.8228	0.6259	0.5130	0.3241	0.7127	0.7283
AAPSA [31]	0.9183	0.6706	0.7207	0.5098	0.7953	0.7030	0.7742	0.4942	0.4161	0.3302	0.4643	0.6179
CwisarDH [32]	0.9145	0.8274	0.7886	0.5753	0.8581	0.7866	0.6837	0.6406	0.3735	0.3218	0.7227	0.6812
PAWCS [29]	0.9397	0.8938	0.8137	0.7764	0.8934	0.8324	0.8059	0.6433	0.4171	0.4450	0.7667	0.7403
SuBSENSE [21]	0.9503	0.8177	0.8152	0.6569	0.8986	0.8171	0.8594	0.6594	0.4918	0.3894	0.8423	0.7408
SemanticBGS [30]	0.9604	**0.9489**	0.8388	0.7878	0.9244	0.8219	0.8260	0.7888	0.5014	0.5673	0.6921	0.7892
MBS [33]	0.9287	0.7915	0.8367	0.7568	0.8262	0.8194	0.7980	0.6350	0.5158	0.5520	0.5858	0.7288
WeSamBE [34]	0.9413	0.7440	0.7976	0.7392	0.8999	0.7962	0.8608	0.6602	0.5929	0.3844	0.7737	0.7446
ShareM [35]	0.9522	0.8222	0.8141	0.6727	0.8898	0.8319	0.8480	0.7286	0.5419	0.3860	0.7339	0.7474
GMM [8]	0.8245	0.633	0.5969	0.5207	0.7370	0.6621	0.7380	0.5373	0.4097	0.1522	0.4663	0.5707
RMoG [36]	0.7848	0.7352	0.7010	0.5431	0.7212	0.4788	0.6826	0.5312	0.4265	0.2470	0.4578	0.5735
DPVT	**0.9811**	0.9329	**0.9014**	**0.9595**	**0.9467**	**0.9479**	**0.8780**	0.7818	**0.7737**	**0.5957**	**0.9034**	**0.8789**

the non-linearities of the manifold of pixel variations.

All these experiments of the proposed method were implemented in matlab and ran on the computer with Nvidia tasla K80 GPU and all images keep their original resolution. For each video in CDnet 2014, 100 training frames are extracted to produce the image patch. It should be noted that the 100 frames only accounts for less than 10% of total Groundtruth in CDnet 2014. In contrast, 90% of data were used as training samples in the DBMF, which suggest that the proposed approach achieves well performance with limited training frames. Considering that videos in CAMO-UOW have fewer frames, we reduce the number of training frames to 64 for each video. During the experience, the training set and testing set are completely separated. More specifically, our FCN network is random initialized. We train the network with mini-batches of size 200, a learning rate $\alpha = 1 \times 10^3$ over 20 epochs. The last threshold R is set to 0.6.

TABLE II
THE PERFORMANCE COMPARISON OF THE PROPOSED APPROACH AND SOME CLASSICAL METHODS AND DEEP-BASED METHOD DBMF. FOR EACH VIDEO, 64 FRAMES WERE TAKEN FOR THE TRAINING OF OUR FCN.

Methods	highway	office	Pedestrians	PETS2006	Fall	sofa	overall
GMM [3]	0.5788	0.2338	0.5202	0.6011	0.8026	0.5225	0.5432
CodeBook [37]	0.8356	0.5939	0.7293	0.7808	0.3921	0.8149	0.6911
ViBe [2]	0.7535	0.6676	0.8367	0.6668	0.6829	0.4298	0.6729
PBAS [10]	0.8071	0.6839	0.7902	0.7280	0.3420	0.5768	0.6547
P2M [38]	0.9160	0.3849	0.9121	0.7322	0.5819	0.4352	0.6604
DBMF [23]	0.9412	0.9236	0.8394	0.9059	0.8203	0.8645	0.8824
DPVT	**0.9888**	**0.9819**	**0.9728**	**0.9808**	**0.9394**	**0.9333**	**0.9662**

TABLE III
THE PERFORMANCE COMPARISON OF THE PROPOSED APPROACH AND SOME STATE-OF-THE-ART ALGORITHMS ON THE VIDEO SEQUENCES FROM DIFFERENT CATEGORIES IN CAMO-UOW.

Methods	MOG2 [39]	FCI [40]	LBA-SOM [41]	PBAS	SuBSENSE	ML-BGS [42]	DECOLOR [43]	COROLA [44]	FWFC [45]	DPVT
Video 1	0.79	0.88	0.8	0.9	0.89	0.89	0.92	0.8	0.94	**0.96**
Video 2	0.82	0.79	0.8	0.82	0.88	0.8	0.83	0.58	0.96	**0.98**
Video 3	0.88	0.86	0.85	0.91	0.9	0.8	0.9	0.82	0.94	**0.95**
Video 4	0.89	0.9	0.76	0.93	0.78	0.88	0.95	0.87	0.94	**0.98**
Video 5	0.84	0.86	0.82	0.83	0.82	0.8	0.82	0.75	0.91	**0.98**
Video 6	0.93	0.87	0.77	0.95	0.92	0.95	0.97	0.72	0.94	**0.98**
Video 7	0.76	0.83	0.88	0.91	0.87	0.79	0.91	0.83	0.96	**0.99**
Video 8	0.83	0.87	0.85	0.87	0.93	0.86	0.86	0.68	**0.96**	**0.96**
Video 9	0.89	0.9	0.87	0.84	0.92	0.87	0.86	0.78	0.88	**0.99**
Video 10	0.89	0.86	0.89	0.91	0.92	0.9	0.94	0.85	0.96	**0.97**
average	0.85	0.86	0.83	0.89	0.88	0.85	0.90	0.77	0.94	**0.97**

VI. CONCLUSION

In this paper, we proposed a novel foreground detection approach based on deep learning and the variation transformation learning. The DPVTL model includes a FCN network and a novel variation transformation learning framework, which allows us to efficiently combine the temporal coherence and distribution information of pixels over a long period of time. Comparison with other traditional and deep learning methods shows that the DPVTL has good properties on several challenging scenes.

REFERENCES

[1] B. T, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Comput. Sci. Rev.*, vol. 1112, pp. 31 – 66, 2014.

[2] O. Barnich and M. Van Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," *IEEE Trans. on Image. Process.*, vol. 20, no. 6, pp. 1709–1724, June 2011.

[3] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognit.*, vol. 2, 1999, pp. –252 Vol. 2.

[4] D.-S. Lee, "Effective gaussian mixture learning for video background subtraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 827–832, May 2005.

[5] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," *Eccv*, vol. 1843, pp. 751–767, 2000.

[6] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, April 2017.

[7] K. Goyal and J. Singhai, "Review of background subtraction methods using gaussian mixture model for video surveillance systems," *Artificial Intelligence Review*, vol. 50, no. 2, pp. 241–259, Aug 2018.

[8] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. Int. Conf. on Pattern Recognit. (ICPR)*, vol. 2, 2004.

[9] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, pp. 172–185, 2005.

[10] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Computer Vision and Pattern Recognition Workshops*, 2012, pp. 38–43.

[11] S. Varadarajan, P. Miller, and H. Zhou, "Region-based mixture of gaussians modelling for foreground detection in dynamic scenes," *Pattern Recognit.*, vol. 48, no. 11, pp. 3488 – 3503, 2015.

[12] M. Chen, X. Wei, Q. Yang, Q. Li, G. Wang, and M. H. Yang, "Spatiotemporal gmm for background subtraction with superpixel hierarchy," *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. PP, no. 99, pp. 1–1, 2018.

[13] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection." *IEEE Trans Pattern Anal Mach Intell*, vol. 27, no. 11, pp. 1778–1792, 2005.

[14] V. Reddy, C. Sanderson, and B. C. Lovell, "Robust foreground object segmentation via adaptive region-based background modelling," in *International Conference on Pattern Recognition*, 2010, pp. 3939–3942.

[15] M. Izadi and P. Saeedi, "Robust region-based background subtraction and shadow removing using color and gradient information," pp. 1–5, 01 2008.

[16] B. Han and L. Davis, "Density-based multifeature background subtraction with support vector machine," *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 34, no. 5, pp. 1017–1023, May 2012.

[17] X. Zhang, Z. Liu, H. Li, X. Zhao, and P. Zhang, "Statistical background subtraction based on imbalanced learning," in *2014 IEEE International Conference on Multimedia and Expo (ICME)*, vol. 00, July 2014, pp. 1–6.

[18] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," 09 2016.

[19] M. Braham and M. Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," pp. 1–4, 05 2016.

[20] "A deep convolutional neural network for video sequence background subtraction," *Pattern Recognition*, vol. 76, pp. 635 – 649, 2018.

[21] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin, "Subsense: A universal change detection method with local adaptive sensitivity," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 359–373, Jan 2015.

[22] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Static and moving object detection using flux tensor with split gaussian models," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2014, pp. 420–424.

[23] L. Yang, J. Li, Y. Luo, Y. Zhao, H. Cheng, and J. Li, "Deep background modeling using fully convolutional network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 254–262, Jan 2018.

[24] Y. Wang, P. M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "Cdnet 2014: An expanded change detection benchmark dataset," in

2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, June 2014, pp. 393–400.

[25] S. Li, D. Florencio, W. Li, Y. Zhao, and C. Cook, "A fusion framework for camouflaged moving foreground detection in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3918–3930, Aug 2018.

[26] S. Bianco, G. Ciocca, and R. Schettini, "Combination of video change detection algorithms by genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 6, pp. 914–928, Dec 2017.

[27] S. Jiang and X. Lu, "Wesambe: A weight-sample-based method for background subtraction," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2017.

[28] Y. Chen, J. Wang, and H. Lu, "Learning sharable models for robust background subtraction," in *2015 IEEE International Conference on Multimedia and Expo (ICME)*, June 2015, pp. 1–6.

[29] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, Jan 2015, pp. 990–997.

[30] M. Braham, S. Pierard, and M. V. Droogenbroeck, "Semantic background subtraction," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sept 2017, pp. 4552–4556.

[31] G. Ramłrez-Alonso and M. I. Chacn-Murguła, "Auto-adaptive parallel som architecture with a modular analysis for dynamic object segmentation in videos," *Neurocomputing*, vol. 175, pp. 990 – 1000, 2016.

[32] M. D. Gregorio and M. Giordano, "Change detection with weightless neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2014, pp. 409–413.

[33] H. Sajid and S.-C. S. Cheung, "Universal multimode background subtraction," *Submitted to Image Processing, IEEE Transactions on*, 2015.

[34] S. Jiang and X. Lu, "Wesambe: A weight-sample-based method for background subtraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. PP, no. 99, pp. 1–1, 2017.

[35] Y. Chen, J. Wang, and H. Lu, "Learning sharable models for robust background subtraction," in *Multimedia and Expo (ICME), 2015 IEEE International Conference on*, June 2015, pp. 1–6.

[36] S. Varadarajan, P. Miller, and H. Zhou, "Spatial mixture of Gaussians for dynamic background modelling," in *Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2013, pp. 63–68.

[37] M. Wu and X. Peng, "Spatio-temporal context for codebook-based dynamic background subtraction," *AEU - International Journal of Electronics and Communications*, vol. 64, no. 8, pp. 739 – 747, 2010.

[38] L. Yang, H. Cheng, J. Su, and X. Li, "Pixel-to-model distance for robust background reconstruction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 903–916, May 2016.

[39] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773 – 780, 2006.

[40] F. E. Baf, T. Bouwmans, and B. Vachon, "Fuzzy integral for moving object detection," in *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, June 2008, pp. 1729–1736.

[41] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1168–1177, July 2008.

[42] J. Yao and J. M. Odobez, "Multi-layer background subtraction based on color and texture," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.

[43] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 597–610, March 2013.

[44] M. Shakeri and H. Zhang, "Corola: A sequential solution to moving object detection using low-rank approximation," *Computer Vision and Image Understanding*, vol. 146, pp. 27 – 39, 2016.

[45] S. Li, D. Florencio, W. Li, Y. Zhao, and C. Cook, "A fusion framework for camouflaged moving foreground detection in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3918–3930, Aug 2018.