Background Subtraction via Deep Variation Transformation

Yongxin Ge, Xinyu Ren, Chenqiu Zhao

***Abstract*—One of the main challenges of foreground detection comes from the diversity and complexity of real-world scenes. While most of previous works in this field were proposed by designing an artificial model, we tend to present a self-adaption solution, by combining the fully convolutional networks (FCNs), which is named as the Deep Pixels Variation Learning model (DPVL). Our main job is to find a new representation of pixels historical observations in a new feature space. In this paper, videos are divided into fixed length image stacks, which are later transformed into the pixel observation matrixes as the input of a FCN for learning the variation of pixels. We conduct our training and prediction processes on stack level. The architecture of the FCN is devised from the semantic segmentation problem. Experiment shows that our model has a strong learning ability to the patterns of those permutations of pixels variation. We compare our method with some state-of-the-art methods, and the results show that the proposed approach outperform its**

***Index Terms*—Background Subtraction, Feature Transformation, Deep Learning,**

I. Introduction

Background subtraction or foreground object detection, as a fundamental problem in computer vision, has been much discussed with the increasing number of outdoor cameras over the last few decades. It is widely used as the pre-processing step of video processing, which can help us efficiently mark the region of interest, e.g. vehicles and humans, thus saving us huge amount of computing resources. Typically, Background subtraction can be viewed as a binary classification that assigns each pixel in a video sequence with a label, for either belonging to the background or foreground scene.

With a huge number of works, existed algorithms have already achieved well performances in the scenes of low diversity or complexity, such as the indoor scenes. However, background Subtraction is still unsolved because of the diversity in background scenes and the changes originated from the camera itself. Scene variations can be in many forms such as, camera jitter, dynamic background, bad weather, illumination changes, intermittent object motion. This is principally because the major methods, in most cases, try to find a universal solution by generating some universal background models for later linear classification; for instance, the earlier Frame Difference Methods, trying to get a fixed image as background to classify each pixels. The fundamental problem lies in that linear classifier might not be powerful enough for the job, due to the complexity and the diversity of natural scenes.

In order to present a better and universal solution, we proposed the Deep Variation Transformation Learning (DVTL) model for background subtraction in diversely natural scenes. In our method, the main job is to transform the sequence of pixels into another sequence which is easy for classification.

For *Sq*, it is a sequence of pixel variation, which is hard to classify which entry is foreground or background. But in *Sq'*, it is more easy to classified, since the output of the network is the label of each entry in sequence.

In this paper, pixel observation matrixes are proposed to describe the historical variation of pixels. And a random initialized FCN network is trained to learn the pixel variation and generate a new representation in a new feature space. We take advantage of the strong learning ability of FCN to learn a new representation of the pixel observation matrixes in a new space where the pixels can be easily classified to background and foreground.

II. Related Work

[1] Over the last few decades, Background subtraction has been well studied. Meanwhile, a huge number of methods were proposed. These methods can be broadly categorized into pixels-based, region-based, frame-based and Deep Learning.

A. pixels-based methods

The most widely used algorithms in Background subtraction are pixels-based methods. And one of the famous method is Gaussian Mixture Model (GMM), in which a GMM is used to model the history over time of pixels intensity values. It is assumed that pixels are independent from their neighbors. Incoming pixels are labeled as background if there exists a Gaussian in the GMM, where the distance between its mean and the pixel lies within a certain bound. For learning the parameters, that maximize the likelihood, the authors proposed an online method that approximates the Expectation Maximization (EM) algorithm.

In XXX , Mingliang Chen et al. propose a background subtraction algorithm using hierarchical superpixels segmentation, spanning trees and optical flow. Their Background model combine the GMM with constrains of temporal and spatial from optical flow and superpixels.

Kim et al used a codebook to record the sampling background values at each pixel, which can be seemed as a compressed representation of background model. This allows them efficient in memory and speed compared with other background modeling techniques. The final foreground is detected by a distance measurement in a cylindrical color model. In XXX, Zhi Zeng et al. proposed an equal-qualification updating strategy to replace the maximum-negative-run-length-based filtering strategy. Their experiments show that, the proposed method outperforms well, despite using only color information. Elgammal et al. introduced a probabilistic non-parametric method to model the background. It is assumed that each background pixel is drawn from a PDF. The PDF for each pixel is estimated with Kernel Density Estimation (KDE).

B. region-based approach

region-based approach assumpt that the neighboring pixels have a similar variation as the pixel itself.

In xxx, Sriram Varadarajan et al. propose a region-based MoG to takes in which the updated mixtures represent the scene distribution in a neighborhood region. In (PCA), classification is done by comparing a stack in current frame to its reconstruction from PCA coefficients and declaring it as background if the reconstruction is close. recently region based method is presented in XXX which used the statistical circular shift moments (SCSM) in image regions for change detection. subspace learning method in xxx, is used to compress the background into the eigenbackground. For each video, the mean and the covariance matrix are calculated. After a PCA of the covariance matrix, a projection matrix is set up with M eigenvectors. Then, incoming images are compared with their projection onto the eigenvectors. Foreground labels are assigned to pixels with large distances, after calculating the distances between the image and the projection and comparing them with the corresponding threshold value.

Marghes et al. used a mixed method that combines a reconstructive method (PCA) with a discriminative one (LDA) to robustly model the background. single Gaussians are employed for foreground modeling. By computing flux tensors, which depict variations of optical flow within a local 3D spatio-temporal volume, blob motion is detected. With the combination of the different information from blob motion, foreground models and background models, moving and static foreground objects can be spotted. Also, by applying edge matching, static foreground objects can be classified as ghosts or intermittent motions. Frame-based background modeling via Principal Component Analysis (PCA) and low-rank/sparse decomposition approaches is a popular alternative to pixel-level modeling xx. These approaches are however not ideal for surveillance applications as most rely on batch or offline processing or suffer from scaling problems.

In XXX, ss et al. addressed scaling problems by reformulating principal component analysis for 2D images. Meanwhile their method takes much lower memory consumption and computational cost than others. Some online approaches have also been proposed recently, but they are still very computationally expensive.

C. Deep learning methods

novel approach for background subtraction with the use of CNN was proposed by Braham and Droogenbroeck. They used a fixed background model, which was generated from a temporal median operation over N video frames. In XXX, Yi Wang et al. tried a CNN architecture combined with a Cascade model for segmentation in Background subtraction. Given 200 labelled images as training set, their model performed excellently in dataset2014. Braham et al. present an Deep learning-based method. with the help of CNN, they generated a fixed background model from a temporal median operation over N video frames. Then, a scene-specific CNN is trained with corresponding image matrixes from the background image. In xxx, M. Babaee et al. combine the segmentation mask from SuBSENSE algorithm and the output of Flux Tensor algorithm, which can dynamically change the parameters used in the background model based on the motion changes in the video frames. They also used spatial-median filtering as the post processing of the network outputs.

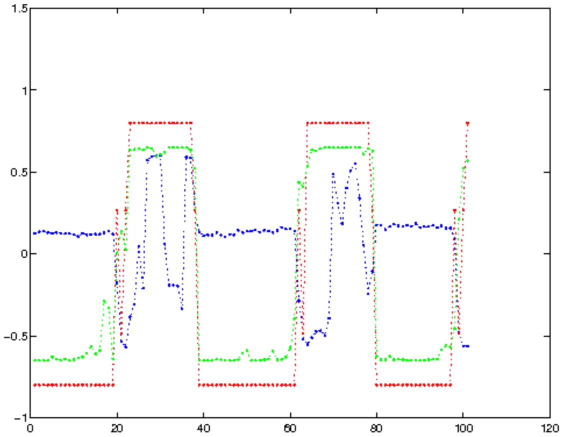

Fig. 1. It is hard to separate the foreground pixels from the background precisely in a time sequence.

III. VARIATION TRANSFORMATION

In this section, we will elaborate on the motivation of the proposed approach and how does the variation transformation works in background subtraction.

Background subtraction is essentially an binary classification in time sequence, where pixels in a video stream are separated into 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 varies 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 variations, we plot the historical observations and corresponding groundtruth of a single pixel in Figure.1, with the X axis showing their range of variation and the Y axis representing the time. As we can see, there are several lines in three colors: blue, red and green. The blue one is called the pixel variation curve, which containing 100 original observations. Accordingly, the red one, the groundtruth curve is made of a piece of groundtruth data corresponding to the observations. The green one, which we call the transformed variation curve, is made of the outputs of the proposed approach. In practice, background subtraction is to produce a prediction curve, which is as near as possible to the groundtruth carve, by given the pixel variation carve.

Under ideal conditions like indoor videos, previous methods perform quite effectively, when distributions of 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 Figure.1, it is noticeable that some foreground observations in blue line share the similarly value with background ones. In that case, those popular solutions 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. Unfortunately, most previous methods, as far as we know, are not capable to take advantage of the temporal coherence of pixel variation. In other words, despite knowing that pixels from the background share some common patterns of variation in a temporal sequence, we still let the order information of sequential images all go to waste. To alleviate this, order information of pixels must be taken in consideration. More concretely, we must find an efficient method to model the patterns of background pixels variations.

In this paper, a method of background subtraction based on variation transformation are proposed. Pixel observations are no longer considered independent of each other but regarded as a whole, which we call the pixel variation. Consequently, the classification of pixels can be viewed as a transformation of pixel variations, from the observation sequences to the prediction sequences. In the specific implementation, we trained a FCN to learn a transformation for the pixel variations by mapping them into a new space where it is close to the groundtruth, just like the green line in Figure.1. After thresholding, we can easily get the labels of each observation. The benefits from variation transformation are evident and clearly seen. It is hard to distinguish a foreground observation when its value are similar to the background ones. However, classification on the transformed variation is much convenient and intuitive, due to the advantage of temporal information. With the aid of deep learning method, the proposed approach can be effectively implemented.

And the definition of region searching method $G(x, y)$ is shown as follows:

$$G(x, y) = \operatorname{argmin} \|I_t(m, n) - I_b(x, y)\|_1 \quad m, n \in x, y \pm R, \tag{1}$$

where x and y is the location of pixels. And I_t is the current frame, where t is the time index.

IV. Background Subtraction via Deep Variation Transformation

In this section, we introduce the proposed approach that consists of a novel FCN network for background subtraction. We explain the details of the procedures of pixel matrixes, which is a specific form of pixel variation, and the architecture of our network.

The complete system is illustrated in Figure 1. Firstly, we temporally sample the input and ground truth images to generalize the pixel variations, and reshape them into fixed-size matrixes and fed them into the network. After reassembling the matrixes into the complete output frame, it is post-processed, yielding the final segmentation of the respective video frame.

Given different videos, the number of frames are generally different. However, the size of our input is fixed, which means a sampling processing is required to keep the length of pixel variations invariant. Therefore, image stacks are sampled form the given videos before the pixel variations are extracted. The given video and image stacks can be defined as follows:

$$Given\ Video = \{I_1, I_2, I_3, \ldots, I_L\}, \tag{2}$$

$$Stack_{(x)} = \{I_x, I_{(x+p)}, I_{(x+2p)}, \ldots, I_L\}, \quad p{*}l = L, \quad 1 \leq x \leq p, \tag{3}$$

Where I_t represent the frame t of the given video. And p is an integer number depended on the video frames length. For each video, we can produce multiple image stacks and choose one of them for the training. In addition, thanks to the temporal sampling, we can get compact pixel historical observations containing more temporal information than the continuous pixel sequences. This works well when it comes to some situation where the moving objects keep stationary for a long time.

In order to make use of the temporal information of pixel historical observations, we regard the sampled observations from a single pixel as a whole, namely the pixel variation. After the temporal sampling, a large number of observation sequences, or pixel variations, are extracted from the chosen stack, which is shown as follows:

$$Sq_{(m)} = \{P_1^m, P_2^m, P_3^m, \ldots, P_n^m\}, \tag{4}$$

Where P_t^m denotes the numerical value of pixel m at frame t in the chosen stack. 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 n. It is notable that the quantity of observation sequences is exactly the resolution of the given video, which means an abundance of training data can be obtained with just one image stack.

Our intention 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 hope the pixel observations can be interacted with their further compatriots in temporal sequence.

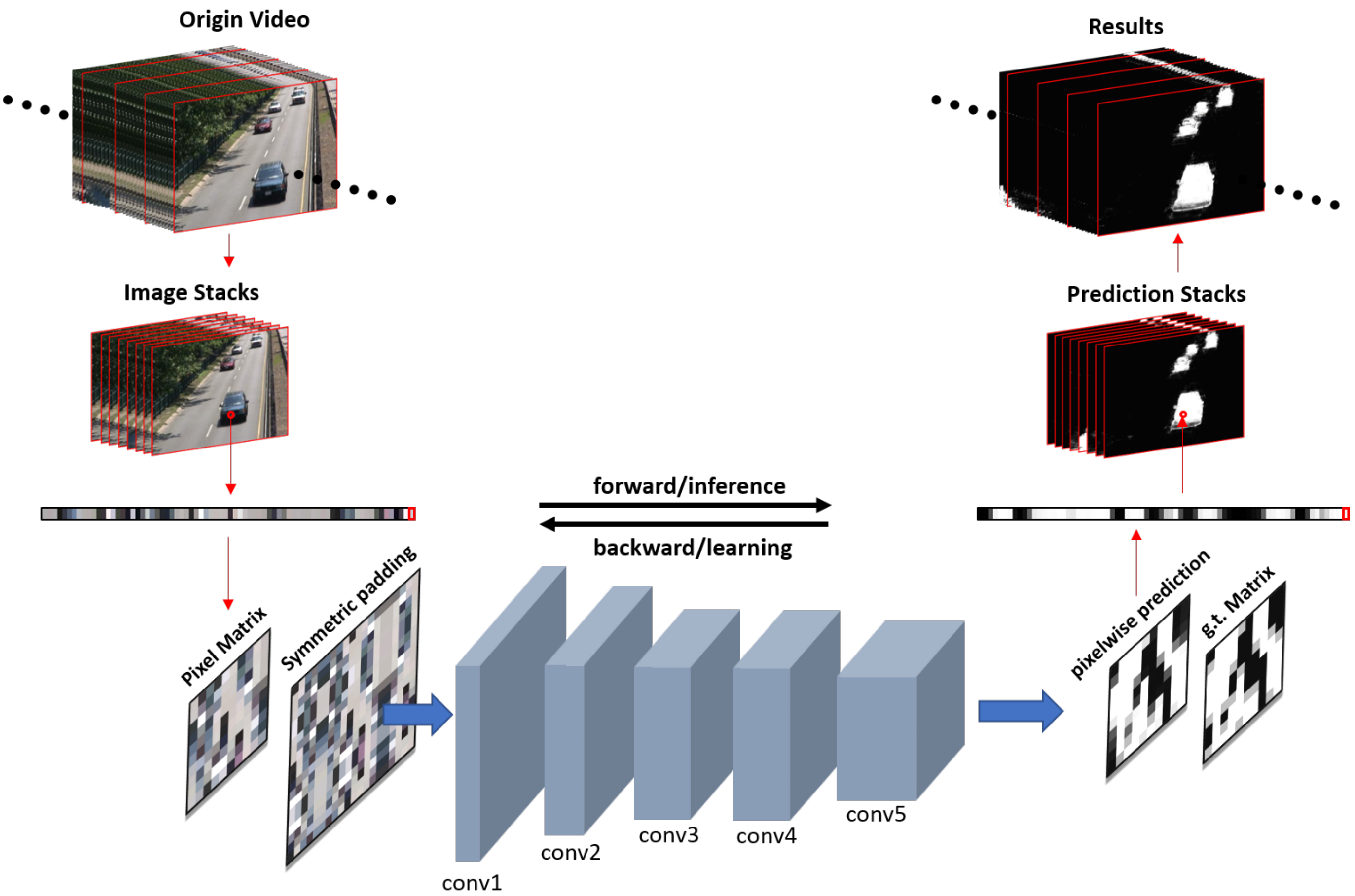

Fig. 2. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

Thus we reshape the variations into pixel matrixes as the input of our network. A sample of pixel matrix H is like this:

$$H = P_{i+j*d} = \begin{bmatrix} P_1 & P_2 & \dots & P_d \\ \vdots & & & \vdots \\ P_{1+(d-1)d} & \dots & \dots & P_D \end{bmatrix}, \quad d^2 = D, \tag{5}$$

Pixel observations are put into the $d * d$ matrix according to the order of top to bottom and left to right. The parameter i and j represent the column and row respectively. And the parameter D is the total frame length of a variation. To put it from another way, pixel matrix is just a specific form of the pixel variation. Although the pixel matrix and the observation sequence belong to different forms of the pixel variation, they are formed of the same components, and share the common temporal information from the pixel variation. Since the pixel sequences are extracted from individual pixels, a large number of pixel matrixes may be extracted from only a single image stack.

In the previous steps, videos are broken down into pixel matrixes which containing abundant background information. Next, the groundtruth matrixes are obtained in the same way. Both of them are put into the network for the training. However, the size of the input will decreased after the network computing. In order to make output the same size as input, we borrowed ideas from Image semantic segmentation, which is doing padding to the input before the training. After the forward computing, variations are transformed in a new space where they can be easily classified by thresholding. The new representation of input pixel variation is defined as:

$$H_p = \mathcal{L}(E_x(H)), \tag{6}$$

Where, H_p denotes the output of network, which we call the prediction matrix, the forward computation of network is represented by L, and E_x denotes the symmetric padding of pixel matrix where we have lost none of the pixel information to make the output H_p the same size as the pixel matrix H. 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:

$$\begin{aligned} \ell_{H_p,H_{gt}} &= \sum_{x\in X, y\in Y} H_{gt}(x,y) log(Sigmoid(H_p(x,y))) \\ &+ (1 - H_{gt}(x,y)) log(1 - Sigmoid(H_p(x,y))) \\ Sigmoid(x) &= \frac{1}{1+e^{-x}}\ , \end{aligned} \tag{7}$$

Where H_{gt} denotes the groundtruth matrix which is given by the corresponding GT stack. The SCE is calculated between the transformed matrixes and the corresponding groundtruth matrixes. Boundaries of moving objects and pixels that out of the region of interest are ignored in the cost function. Finally, we get the transformed variation through the FCN, which are easier for classification. We globally threshold the values for each observation in order to map them to $\{0, 1\}$. The threshold function is given by

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

$$M(x, y) = g(H_p, r), \tag{9}$$

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. And the major contribution is that we demonstrate the effectiveness of temporal information in background subtraction.

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

Different with CNNs, fully convolution neural networks (FCNs) utilize convolutional layers with $1 * 1$ kernels to take the place of fully connected layers, which largely resolves these above-mentioned problems. First and foremost, the size of outputs are adjustable in FCNs, which allows an end-to-end mapping of pixel observation sequences and network outputs on the time sequence. We hope to determine the label of a pixel 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 need 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 wont lost to the traditional ones. Meanwhile, its also a high 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 background modeling is shown in Fig.1. The proposed FCN contains 5 convolutional layers, 2 pool layers and a convolutional layer which have a filter size of $1 * 1$. We use the Rectified Linear Unit (ReLU) as 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 and CAMO-UOW. The CDnet is the largest dataset for background subtraction 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 another challenging benchmark 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 background subtraction algorithms, including the IUTIS-5, the SuBSENSE, the WeSamBE, sharable GMM, the SharedModel, word-dictionaries-based method and PAWCS, the SemanticBGS, the AAPSA, etc. Moreover, two deep learning based algorithms are also compared with the proposed approach, which include DeepBS, and DBMF. 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 background subtraction which measures the segmentation accuracy by considering both the recall and the precision. The definition of Fm is shown as follows:

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

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.3 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 background subtraction method such as the SharedModel and the WeSamBE do not have the enough ability to describe the complex dynamically background, they are fail to detect the people on the boat, as shown in the Fig. X. Besides, the detected moving objects of the SharedModel are not accurate in 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 background subtraction 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 UBSS is incomplete on the part of the pedestrians body due to the interference of shades, which can be owing to the

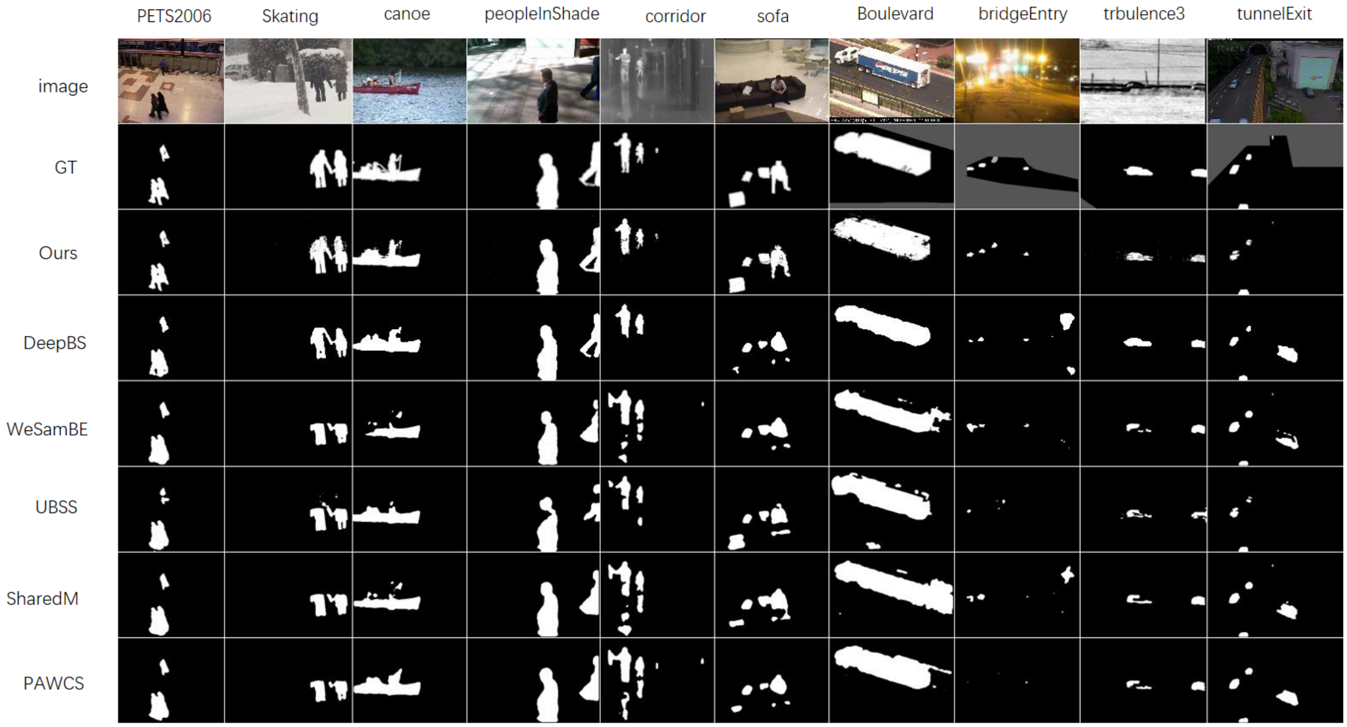

Fig. 3. The qualitative evaluation of the proposed method. All the results is followed in the CDnet 2014.

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 matrix-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, the 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 have already reached a limit under the extreme condition of thermal map. The DeepBS, by contrast, succeeds in eliminating most of the reflections. Meanwhile, the moving objects are also clearly divided from the background thanks to the strong modeling ability of CNN. However, due to the dependency of 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 shadow are removed and the segmentation results are more accurate.

The quantitative evaluation of 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 6% gain in FM over the second one on the whole dataset. Moreover, in order to compare proposed approach with the DBMF, which is also based on deep learning and only publish their results in several special vides. The proposed approach has also ran in these video 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 gives us the ability to avoid the shortcomings of the background models. Consequently, the proposed approach still get considerably better results, which over 10.46% in FM metrics compared with the DeepBS and over 6.38% compared with the DBMF. The evaluation of 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 the moving objects with camouflage, which is the main challenge of this dataset. As shown in the Table III proposed approach achieves better

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.

Videos	baseline	dyna.bg	cam.jitter	int.obj.m	shadow	thermal	bad.weat	low f.rate	night vid.	PTZ	turbul.	overall
DeepBS	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	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	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	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	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	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	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	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	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	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	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	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	0.7848	0.7352	0.7010	0.5431	0.7212	0.4788	0.6826	0.5312	0.4265	0.2470	0.4578	0.5735
Ours	**0.9668**	0.8639	0.8446	**0.9181**	**0.9390**	**0.9268**	**0.9265**	0.7338	**0.7306**	**0.5845**	0.8761	**0.8504**

TABLE II
THE PERFORMANCE COMPARISON OF THE PROPOSED APPROACH AND SOME CLASSICAL METHODS AND DEEP-BASED METHOD DBMF .

Methods	highway	office	Pedestrians	PETS2006	Fall	sofa	overall
GMM	0.5788	0.2338	0.5202	0.6011	0.8026	0.5225	0.5432
CodeBook	0.8356	0.5939	0.7293	0.7808	0.3921	0.8149	0.6911
ViBe	0.7535	0.6676	0.8367	0.6668	0.6829	0.4298	0.6729
PBAS	0.8071	0.6839	0.7902	0.7280	0.3420	0.5768	0.6547
P2M	0.9160	0.3849	0.9121	0.7322	0.5819	0.4352	0.6604
DBMF	0.9412	0.9236	0.8394	0.9059	0.8203	0.8645	0.8824
ours	**0.9775**	**0.9565**	**0.9652**	**0.9681**	**0.9157**	**0.8943**	**0.9462**

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	FCI	LBA-SOM	PBAS	SuBSENSE	ML-BGS	DECOLOR	COROLA	FWFC	Ours
Video 1	0.79	0.88	0.8	0.9	0.89	0.89	0.92	0.8	**0.94**	**0.94**
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**	**0.94**
Video 4	0.89	0.9	0.76	0.93	0.78	0.88	0.95	0.87	0.94	**0.97**
Video 5	0.84	0.86	0.82	0.83	0.82	0.8	0.82	0.75	0.91	**0.97**
Video 6	0.93	0.87	0.77	0.95	0.92	0.95	**0.97**	0.72	0.94	0.96
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	**0.98**
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**

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 proposed approach performs better compared with their peers.

In this dataset, target objects have the 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 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 tasela K80 GPU and all images are keep their original resolution. For each video in CDnet 2014, 100 training frames are extracted to produce the image matrix. 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 the videos in CAMO-UOW have fewer frames, we reduce the number of training frames to 49. 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 100, a learning rate $= 110^3$ over 20 epochs. The last threshold R is set to 0.6.

VI. CONCLUSION

In this paper, we proposed the Moreover, we captured different superpixels under multiple scales and integrating these superpixels to improve the accuracy of proposed approach.

REFERENCES

[1] T. Bouwmans, “Traditional and recent approaches in background modeling for foreground detection: An overview,” *Computer Science Review*, vol. 1112, pp. 31 – 66, 2014.