JOINT BACKGROUND RECONSTRUCTION AND FOREGROUND SEGMENTATION VIA A TWO-STAGE CONVOLUTIONAL NEURAL NETWORK

Xu Zhao[1,2], *Yingying Chen*[1,2], *Ming Tang*[*,1,2] *and Jinqiao Wang*[1,2]

[1]National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing, China, 100190
[2]University of Chinese Academy of Sciences, Beijing, China, 100190
{xu.zhao, yingying.chen, tangm, jqwang}@nlpr.ia.ac.cn

ABSTRACT

Foreground segmentation in video sequences is a classic topic in computer vision. Due to the lack of semantic and prior knowledge, it is difficult for existing methods to deal with sophisticated scenes well. Therefore, in this paper, we propose an end-to-end two-stage deep convolutional neural network (CNN) framework for foreground segmentation in video sequences. In the first stage, a convolutional encoder-decoder sub-network is employed to reconstruct the background images and encode rich prior knowledge of background scenes. In the second stage, the reconstructed background and current frame are input into a multi-channel fully-convolutional sub-network (MCFCN) for accurate foreground segmentation. In the two-stage CNN, the reconstruction loss and segmentation loss are jointly optimized. The background images and foreground objects are output simultaneously in an end-to-end way. Moreover, by incorporating the prior semantic knowledge of foreground and background in the pre-training process, our method could restrain the background noise and keep the integrity of foreground objects at the same time. Experiments on CDNet 2014 show that our method outperforms the state-of-the-art by 4.9%.

Index Terms— Foreground segmentation, background modeling, convolutional neural network

1. INTRODUCTION

Foreground segmentation in video sequences is classic topic in computer vision. It is the fundamental step for many high-level computer vision tasks, such as activity recognition, object tracking and automated anomaly detection in video surveillance. In the last two decades, many algorithms have been proposed to solve this problem [1, 2, 3, 4, 5, 6, 7, 8, 9].

Statistically modeling background is a prevalent strategy to extract foreground for its robustness and efficiency. Some typical methods, such as GMM [4], KDE [5], and ViBe [3] assume the independence among pixels and model the variation of every pixel over time. Several other works [6, 1, 7] attempt to employ more discriminative hand-crafted features. Nevertheless, all existing methods perform poorly in scenes where the night light, heavy shadows, or camouflaged foreground are involved, because they all do not consider the semantic explanation of the scenes. And usually, post-processing has to be employed by them to smooth the foreground edges and reduce the noises on the foreground.

In general, the existing background modeling methods are considered unsupervised. The fact, however, is that there still exist many parameters in them to control the initialization, classification, and update rate of background model, etc., and unfortunately, all these parameters have to be set to adapt to different characteristics of videos. In order to achieve high performance on the whole benchmark, a variety of experiential rules for setting the parameters have to be designed according to characteristics of the whole benchmark. This implies that the background model is actually *tuned with the test data*. Therefore, it could be argued that the existing background modeling methods might not be considered as unsupervised. It would be more reasonable to exploit some labeled data to train the model, and others to test it.

Therefore, in this paper, we propose a supervised end-to-end two-stage deep convolutional neural network to segment the foreground regions, utilizing semantic information learned by the network. An overview of the network is shown in Fig. 1. It consists of a background reconstruction stage and a foreground segmentation stage. Our network is designed based on the deep fully-convolutional network (FCN). FCNs can extract the semantic features of the input. And through supervised learning, it shows good performance in pixel-wise labeling tasks, such as semantic segmentation [10]. However, the original FCN takes a single image as input, so that it lacks the analysis on the foreground motion in its segmentation results, which is necessary to distinguish moving objects from non-moving objects in the foreground segmentation task. To combat this, we restore the background image first, and then

concatenate it with the input image as multi-channel input for segmentation network. Consequently, the motion information of the foreground is naturally implied in the image pair of the input image and its reconstructed background.

Specifically, in the background reconstruction stage, we adopt a deep convolutional encoder-decoder network, which establishes the background model according to training images and outputs the refined reconstructed background. Such a background model encodes rich prior knowledge of background scenes and can handle various background changes and generate clean background images. In the segmentation stage, we adopt a multi-channel fully-convolutional network (MCFCN), which is a variant of FCN. It processes a multi-channel input, which is a six-channel concatenation of the reconstructed background and the current image. The image pair of reconstructed background and current image naturally contains motion information of the foreground. Therefore, besides utilizing the semantic information, the MCFCN can distinguish the moving objects more accurately than the FCN that has single-image input. The two sub-networks are jointly trained for the final optimization, so that two tasks of background reconstruction and foreground segmentation sub-networks benefit from each other in the process of iteration.

The contribution of this paper is three-fold. First, we propose a new two-stage neural network architecture for foreground segmentation, which encodes rich background knowledge and combines semantic features with motion information. Second, with the reconstructed background and current frame as input together, we adopt a multi-channel fully-convolutional network to segment the foreground objects, Third, the end-to-end network is jointly optimized with a multi-task loss including reconstruction loss and segmentation loss. The performances of the two sub-networks are mutually promoted by joint optimization.

2. RELATED WORK

Background Reconstruction. Traditional foreground detection approaches often reconstruct the background by PCA [11] or RPCA [12, 13]. Oliver *et al.* [11] proposed a Principal Component Analysis (PCA) based method that builds a vectorization representation with a set of images, and decomposes the representation via PCA to restore the background. In the last decade, Robust PCA (RPCA) based methods are proposed to decompose the foreground and background. Candes *et al.* [12] proposed a convex optimization to address the RPCA-PCP problem. Recently, Rodriguez and Wohlberg [13] proposed an incremental PCP algorithm that consumes low memory and calculation.

Convolutional Autoencoder. Denoising autoencoders [14] take corrupted image as input and then output the original image. But the original denoising autoencoders [14] can only undo damages that are localized and low-level. In [15] Pathak

et al. presented Context Encoders, a convolutional encoder-decoder network that can generate the contents of an arbitrary missing image region conditioned on its surroundings. In Context Encoders, the missing image regions can be regarded as noises. Thus, Context Encoders share the same characteristics with denoising autoencoders. Regarding the foreground as noises, we use a similar convolutional encoder-decoder framework for background reconstruction as Context Encoders and get a satisfactory reconstruction result.

Semantic Segmentation. FCN [10] has shown good performance in semantic segmentation. Based on FCN, Deeplab [16] architecture uses atrous convolutions to produce a more dense feature map and uses CRF [17] to get a better localization especially along the edge of objects. We adopt almost the same network architecture as the VGG-16 based DeepLab-LargeFOV in the foreground segmentation sub-network, except some modifications like that the first layer is changed to take a 6-channel input, which is the concatenation of the reconstructed background image and the current input frame.

For paired-image inputs, siamese networks [18] has been widely used. Siamese networks is a two-stream network architecture that has two same branches. Without using siamese networks, Dosovitskiy *et al.* [19] and Khan *et al.* [20] adopted one-stream multi-channel networks to process two input images in optical flow prediction and weakly supervised change detection. The one-stream multi-channel network is more efficient than the siamese networks. In the second stage, we adopt the one-stream multi-channel network architecture to locate the foreground regions in the image, inputting the current frame image and background image.

3. OVERVIEW

An overview of the proposed two-stage network is shown in Fig. 1. The first-stage sub-network is a convolutional encoder-decoder network for encoding rich background knowledge and background reconstruction. Next, the second-stage sub-network is an MCFCN to segment the foreground objects by inputting the current frame and its reconstructed background image. Finally, the whole network is jointly optimized with a multi-task loss.

3.1. Encoder-decoder Network for Background Reconstruction

Inspired by the network in [15] used for inpainting, we adopt an encoder-decoder network to reconstruct the background image. The encoder contains a set of convolutions, and represents the input image as a latent feature vector. The decoder restores the background image from the feature vector. Generally, in the encoder-decoder networks [21, 15, 14], the latent feature vector distils the most useful information from the input image. In our encoder-decoder network, the vector represents the context information and encodes the rich knowledge
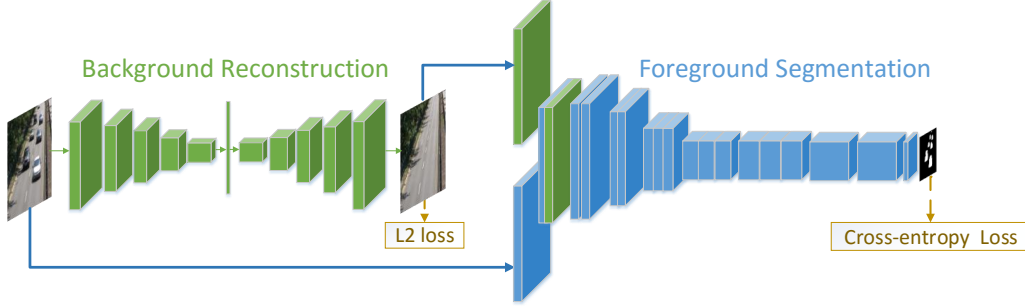
Fig. 1. An overview of the two-stage deep neural network for foreground segmentation. The current frame is input into the encoder-decoder network to reconstruct the background. Then the reconstructed background image is concentrated to the current frame and fed into the following fully convolutional network to segment the foreground.

of background scenes. Then the decoder restores the clean background image from the latent feature vector.

Our encoder-decoder network is designed following the pattern as DCGAN [22] and Context Encoders [15]. The layers of the encoder-decoder network are all convolutional layers. In the encoder part, the number of nodes in each layer is reduced layer by layer. As in [22] and [15], we use strided convolution layers with the stride of 2 instead of using spatial pooling layers to downsample the layer's input. We make the convolution layers learn a pooling operation without limited to "max" or "mean" operation [22], which allows the network to learn customized downsampling function.

We choose the L2 loss as the reconstruction loss, and the loss function of the encoder-decoder network can be expressed as follows:

$$L_{rec} = \sum_{i,j,c} \|B_{ijc} - B_{ijc}^*\|_2, \qquad (1)$$

where B_{ijc} is the value of the reconstruction background image on the i-th row, the j-th colume and c-th color channel. And B^* is the ground truth background image.

The ground truth background images for each training frames are manually generated according to their label maps. After training, the encoder-decoder network can separate the background from the input image and restore a clean background image. This is a good stepping-stone for the following segmentation step.

3.2. MCFCN for Foreground Segmentation

The second stage of our network is a multi-channel fully-convolutional network (MCFCN). Its input is the concatenation of the reconstructed background image and the current frame. The network can learn semantic knowledge of the foreground and background. Therefore, it could handle various changes better, like the night light, shadows and camouflaged foreground objects. And by inputting the image pair of

the background image and current frame, it can easily catch motion information, thus inferring the moving object easily. Besides, according to our experiment results, the network segments rather refined foregrounds without any post processing such as CRF when adding the background image as input.

As mentioned in [19, 20] and proven by our initial experiment, a multi-channel network has comparable performance to the traditionally siamese network [18] for paired images. However, the siamese network consumes as almost twice the memory as an MCFCN. Therefore, we take the 6-channel input by concatenating the reconstructed background image to the current frame, and then adopt MCFCN to segment the foreground regions from the input.

The MCFCN derives from the VGG-16 based DeepLab-LargeFOV network [16], and we make two modifications. The one is that original first layer with 3-channel input is substituted with a new layer taking 6-channel input. The other is to change the arous sampling rate in the "fc6" layer from 12 to 6. This leads to a smaller receptive field, which brings benefit to segment small objects.

For an input image, given its ground truth of foreground segmentation map L and the output probability map P from the softmax layer of the network, the segmentation loss is calculated as,

$$L_{seg} = \sum_{i,j,k} \log P_{i,j,k} \mathbf{1}[k = L_{i,j}] \mathbf{1}[L_{i,j} \neq l^{ign}], \quad (2)$$

where $\mathbf{1}$ is the indicator function. $P_{i,j,k}$ is the probability for class k of the element on the i-th row and the j-th column. And $L_{i,j} \in \{0, 1, -1\}$ is the ground truth label , for which 1 stands for the foreground, 0 stands for the background and -1 stands for the ignored regions ($l^{ign} = -1$).

3.3. Joint Optimization

Multi-task Loss. Previous work on object detection [23] indicates that multi-task learning usually leads to an improvement on performance in tasks that relate to each other. In our

network, the two stages can be integrated into an end-to-end architecture and trained jointly. Combining Eq. 1 and Eq. 2, the multi-task loss can be represented as:

$$L = L_{rec} + \lambda \times L_{seg}, \qquad (3)$$

where λ is a weight parameter.

In this way, the loss computed at the segmentation stage can be back-propagated to the encoder-decoder network. The two losses (L_{rec} and L_{seg}) lead to a more refined background reconstruction and segmentation maps.

Training Details. We train the two-stage network in three steps. In the first step, we train the encoder-decoder sub-network using single loss L_{reg}. In the second step, we fix the parameters of the encoder-decoder sub-network, and train the MCFCN using single loss L_{seg}. Finally, we train the two sub-networks jointly with the parameters of two pre-trained sub-networks. The three-step training assures a faster convergence to the optimal value than directly optimizing the entire network in one single step.

The three steps all adopt the mini-batch stochastic gradient descent (SGD) method to optimize the network. The mini-batch sizes for the three steps are 4, 2 and 1 respectively. And the learning rate is set to 10^{-4}, 10^{-3} and 10^{-5} respectively. The first step trains the sub-network with 20k iterations. The second and the third step take 6000 and 3000 iterations respectively. The MCFCN is initialized with the weights of pre-trained DeepLab model [16]. And the other layers of the whole network is initialized from a zero-centered Normal distribution with standard deviation 0.01.

3.4. Implementation Details

Framework Details. We use the modified CAFFE [24] code released with DeepLab v2 [25, 16] to implement the whole neural network. The input image (I) is resized to 128×128 (I_1) and 961×961 (I_2), and I_1 is fed into the first sub-network to get the reconstructed background image (B) with the size of 128×128. Then B is resized to size 961×961 and fed into the second sub-network with I_2 together. The resize operation between the two sub-networks is implemented by a neural network layer that is initialized with bilinear interpolation's weights. After initialization, this layer's parameters do not update in the training process. Our initial experiment shows the parameter λ in Eq. 3 has little impact on the performance. So, λ is set to 1 in this paper. On a computer with Intel i7 CPU with 4.00GHz and a single NVIDIA GTX Titan X GPU, our algorithm runs at five fps on test videos.

4. EXPERIMENTS

4.1. Dataset

We conduct experiments on the public ChangeDetection benchmark 2014 (CDNet 2014) [26] to evaluate the performance of our algorithm. CDNet 2014 dataset has 53 video
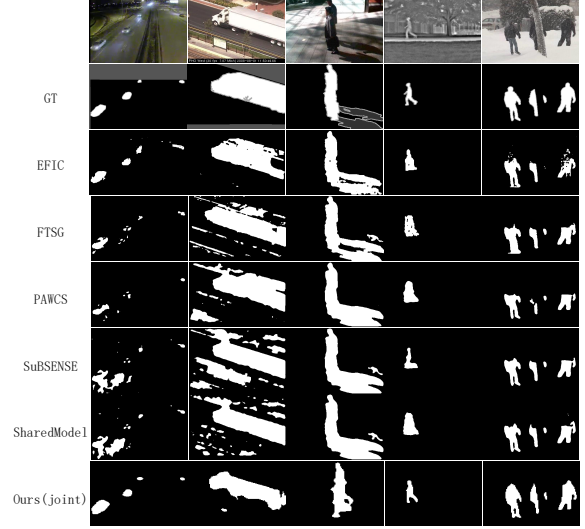

Fig. 2. Visual comparison of foreground segmentation results. From top to bottom, the sequences are "fluidHighway" from *Night Videos*, "boulevard" from *Camera Jitter*, "peopleInShade" from *shadow*, "park" from *Thermal* and "skating" from *Bad Weather*.

sequences belonging to 11 diverse categories. All video sequences are captured in real scenes. Since ground truth is required in the training step of our network, we build up the training and testing data set by dividing the frames having ground truth. Specifically, suppose there are n frames having ground truth, numbered as $1, \ldots, n$, the frames numbered as $[1, \lfloor \frac{n}{2} \rfloor]$ are used as training samples and $[\lfloor \frac{n}{2} \rfloor + 1, n]$ are used for testing. We select the F-Measure to evaluate the performance, the same as CDNet 2014.

We train our network on each video's training frames separately. For some videos, all their training frames have no foreground regions. In this case, we randomly paste 1 or 2 objects as the foreground, which are cut from the object regions of PASCAL VOC 2012 [27] .
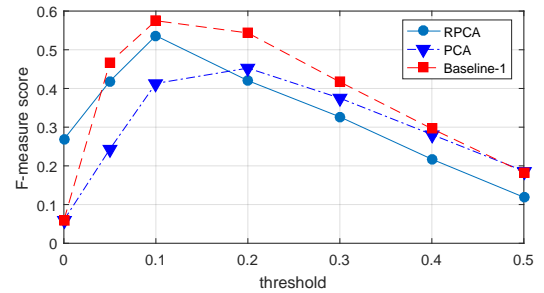

Fig. 3. F-measure scores of three methods (PCA, RPCA, Baseline-1) varing with different thresholds.

Table 1. F-measures for subset of ChangeDetection benchmark 2014 [26].PTZ: Pan/Tilt/Zoom; BW: Bad Weather; Ba: Baseline; CJ: Camera Jitter; DB: Dynamic Background; IOM: Intermittent Object Motion; LF: Low Framerate; NV: Night Videos; Sh: Shadow; Th: Thermal; Tu: Turbulence; Overall is the average F-measure of 11 categories. C-EFIC: [6], FTSG: [7], PAWCS: [2], SuBSENSE: [1], SharedModel: [8], GMM: [4], KDE: [5],ViBe [3].

Approach	PTZ	BW	Ba	CJ	DB	IOM	LF	NV	Sh	Th	Tu	Overall
KDE	0.0512	0.7522	0.9101	0.5748	0.6226	0.3911	0.5097	0.4057	0.8157	0.7573	0.5004	0.5719
GMM	0.1937	0.7819	0.7845	0.5920	0.6650	0.4841	0.5008	0.4003	0.8047	0.6377	0.5364	0.5801
ViBe	0.0666	0.7734	0.8793	0.4476	0.6357	0.4726	0.3286	0.4019	0.8241	0.5512	0.6058	0.5443
C-EFIC	0.6140	0.7731	0.9382	0.8216	0.5531	0.5868	0.6984	0.6502	0.8808	0.8097	0.7230	0.7317
FTSG	0.3923	0.8672	0.9321	0.7066	0.8750	0.8423	0.7133	0.5347	0.8876	0.7791	0.7922	0.7566
PAWCS	0.5728	0.7937	0.9384	0.7880	0.8995	0.7756	0.7531	0.4193	0.9016	0.7975	0.7583	0.7634
SuBSENSE	0.4165	0.8665	0.9527	0.7995	0.8043	0.7259	0.6859	0.5127	0.9037	0.7820	0.8600	0.7554
SharedModel	0.4567	0.8072	0.9558	0.8034	0.8047	0.6834	0.7936	0.4794	0.8907	0.7889	0.8493	0.7557
Baseline-1	0.2815	0.7505	0.8681	0.7587	0.2369	0.6289	0.5780	0.3961	0.7450	0.7651	0.2993	0.5731
Baseline-2	0.5150	0.7228	0.9048	0.8706	0.6796	0.7779	0.7267	0.8042	0.8954	0.8373	0.4242	0.7417
Ours	0.4493	0.8004	0.9630	0.8699	0.7405	0.8734	0.8075	0.6851	0.9216	0.8536	0.6929	0.7870
Ours(joint)	0.5168	0.8550	0.9680	0.8988	0.7716	0.9066	0.7491	0.7695	0.9286	0.8586	0.7143	0.8124

4.2. Experiments on Background Reconstruction

To verify the performance of background reconstruction, we design the first baseline (named Baseline-1) by appending a threshold classifier to the reconstruction network.

We compare the Baseline-1's foreground segmentation results with original PCA-based method (PCA for short) [11] and an incremental PCP-based RPCA method (RPCA for short) [13] on CDNet 2014. The three methods all use the threshold-based classifier to segment the foreground regions. We change the threshold from 0 to 0.5 to find the best value for each method. Fig. 3 shows how the F-measure varies with the threshold. It is clear that our method has better performance than other methods. The reason is that the background reconstruction of our method is more accurate than others.

In spite of the good performance on background reconstruction, a threshold-based classifier only achieves an F-measure less than 0.6. This classifier can not handle cases like the night light, shadows and camouflaged foreground well. We need a better classifier to segment the foreground regions with the reconstructed background image and input frame.

4.3. Experiments on Foreground Segmentation

To evaluate how much the reconstructed background image benefits the segmentation result, of the foreground segmentation sub-network, we design the second baseline (Baseline-2), which segments the foreground using a 3-channel FCN sharing the same structure with our multi-channel FCN at the second stage. It takes the current image as input and produces the foreground segmentation result. We train the Baseline-2 in the same way as our foreground segmentation sub-network. We test both our two-stage network and Baseline-2 on the CDNet 2014 dataset [26] and compare their performance.

In Tabel 1, it is clear that the two-stage network perform-

s better in most scenes than Baseline-2. Thus, we conclude that the segmentation network can infer the foreground regions more accurately by adding background image as input.

4.4. Evaluation on ChangeDetection benchmark

We compare the results of our two-stage network with the state-of-the-art methods and several classic methods on CD-Net 2014 in Table 1. We download the foreground segmentation results of some method [6, 7, 2, 1, 8, 4, 5, 3] from CDNet[1] and calculate the F-measure on our testing data. Besides, the result of ViBe is obtained with BGSLibrary [28]. In Table 1, our method with joint optimization achieves the best performance in 5 out of the 11 categories. The two-stage network improves the state-of-the-art by 2.36%. And after joint optimization, its has another 2.54% gain.

In scenes of *Camera Jitter*, *Intermittent Object Motion*, *Night Video* and *Thermal*, our method outperforms the others by a large margin. There are lots of camouflaged foreground regions in the cases of *Night Video*, *Intermittent Object Motion* and *Thermal*. Combining semantic knowledge and motion information, our network can recognize the camouflaged foreground objects. And the good performance on *Shadow*, *Camera Jitter* and *Night Video* categories demonstrates the effectiveness of our method in dealing with illumination changes, various background changes, and shadows. Fig. 2 shows some visual comparisons of foreground segmentation results. The segmentation maps of our method have less false detections in the background regions and more refined boundaries in the foreground region. In the scene of *Shadow*, compared to other methods, our method removes almost all the shadow. For *Bad Weather*, the camouflaged foreground regions are recognized correctly by our method.

---

[1]http://changedetection.net/

5. CONCLUSION

We design an end-to-end two-stage deep CNN architecture for joint background reconstruction and foreground segmentation. An encoder-decoder network is used to reconstruct the background image, and an MCFCN is introduced to segment the foreground with the concatenation of the reconstructed background image and the current frame. At last, the two sub-networks are jointly optimized with a multi-task loss. Experimental results on CDnet 2014 demonstrate the superiority of the proposed approach.

6. REFERENCES

[1] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau, and Robert Bergevin, "Subsense: A universal change detection method with local adaptive sensitivity," *IEEE TIP*, vol. 24, no. 1, pp. 359–373, 2015.

[2] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau, and Robert Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *WACV*. IEEE, 2015, pp. 990–997.

[3] Olivier Barnich and Marc Van Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," *IEEE TIP*, vol. 20, no. 6, pp. 1709–1724, 2011.

[4] Chris Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *CVPR*, vol. 2, pp. 2246, 1999.

[5] Ahmed Elgammal, David Harwood, and Larry Davis, "Non-parametric model for background subtraction," in *ECCV*. Springer, 2000, pp. 751–767.

[6] Gianni Allebosch, David Van Hamme, Francis Deboeverie, Peter Veelaert, and Wilfried Philips, *C-EFIC: Color and Edge Based Foreground Background Segmentation with Interior Classification*, VISIGRAPP, 2016.

[7] Rui Wang, Filiz Bunyak, Guna Seetharaman, and Kannappan Palaniappan, "Static and moving object detection using flux tensor with split gaussian models," in *CVPRW*, 2014, pp. 414–418.

[8] Yingying Chen, Jinqiao Wang, and Hanqing Lu, "Learning sharable models for robust background subtraction," in *ICME*. IEEE, 2015, pp. 1–6.

[9] Yingying Chen, Jinqiao Wang, Min Xu, Xiangjian He, and Hanqing Lu, "A unified model sharing framework for moving object detection," *SP*, vol. 124, no. C, pp. 72–80, 2016.

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431–3440.

[11] Nuria Oliver, Barbara Rosario, and Alex Pentland, "A bayesian computer vision system for modeling human interactions," in *ICVS*. Springer, 1999, pp. 255–272.

[12] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright, "Robust principal component analysis?," *JACM*, vol. 58, no. 3, pp. 11, 2011.

[13] Paul Rodriguez and Brendt Wohlberg, "Incremental principal component pursuit for video background modeling," *JMIV*, vol. 55, no. 1, pp. 1–18, 2016.

[14] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, "Extracting and composing robust features with denoising autoencoders," in *ICML*. ACM, 2008, pp. 1096–1103.

[15] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros, "Context encoders: Feature learning by inpainting," *arXiv preprint arXiv:1604.07379*, 2016.

[16] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv preprint arXiv:1606.00915*, 2016.

[17] Vladlen Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *NIPS*, 2011.

[18] Sumit Chopra, Raia Hadsell, and Yann Lecun, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR*, 2005, pp. 539–546 vol. 1.

[19] Alexey Dosovitskiy, Philipp Fischery, Eddy Ilg, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, Thomas Brox, et al., "Flownet: Learning optical flow with convolutional networks," in *ICCV*. IEEE, 2015, pp. 2758–2766.

[20] Salman H Khan, Xuming He, Mohammed Bennamoun, Fatih Porikli, Ferdous Sohel, and Roberto Togneri, "Weakly supervised change detection in a pair of images," *arXiv preprint arXiv:1606.02009*, 2016.

[21] Geoffrey E Hinton and Ruslan R Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[22] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *Computer Science*, 2015.

[23] Ross Girshick, "Fast r-cnn," in *ICCV*, 2015, pp. 1440–1448.

[24] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[25] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in *ICLR*, 2015.

[26] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar, "Cdnet 2014: an expanded change detection benchmark dataset," in *CVPRW*, 2014, pp. 387–394.

[27] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes challenge: A retrospective," *IJCV*, vol. 111, no. 1, pp. 98–136, 2015.

[28] Andrews Sobral, "Bgslibrary: An opencv c++ background subtraction library," in *WVC*, 2013.