MotionRec: A Unified Deep Framework for Moving Object Recognition

Murari Mandal
murarimandal.cv@gmail.com

Lav Kush Kumar
lavkushkumarmnit@gmail.com

Mahipal Singh Saran
mahipalsaran007@gmail.com

Santosh Kumar Vipparthi
skvipparthi@mnit.ac.in

Vision Intelligence Lab, Malaviya National Institute of Technology Jaipur, India

Abstract

In this paper we present a novel deep learning framework to perform online moving object recognition (MOR) in streaming videos. The existing methods for moving object detection (MOD) only computes class-agnostic pixel-wise binary segmentation of video frames. On the other hand, the object detection techniques do not differentiate between static and moving objects. To the best of our knowledge, this is a first attempt for simultaneous localization and classification of moving objects in a video, i.e. MOR in a single-stage deep learning framework. We achieve this by labelling axis-aligned bounding boxes for moving objects which requires less computational resources than producing pixel-level estimates. In the proposed MotionRec, both temporal and spatial features are learned using past history and current frames respectively. First, the background is estimated with a temporal depth reductionist (TDR) block. Then the estimated background, current frame and temporal median of recent observations are assimilated to encode spatiotemporal motion saliency. Moreover, feature pyramids are generated from these motion saliency maps to perform regression and classification at multiple levels of feature abstractions. MotionRec works online at inference as it requires only few past frames for MOR. Moreover, it doesn't require predefined target initialization from user. We also annotated axis-aligned bounding boxes (42,614 objects (14,814 cars and 27,800 person) in 24,923 video frames of CDnet 2014 dataset) due to lack of available benchmark datasets for MOR. The performance is observed qualitatively and quantitatively in terms of mAP over a defined unseen test set. Experiments show that the proposed MotionRec significantly improves over strong baselines with RetinaNet architectures for MOR.

1. Introduction

Moving object recognition (MOR) corresponds to the localization and classification of moving objects in videos. Discriminating moving objects from static objects and background in videos is an essential task for many computer vision applications. MOR has widespread applications in intelligent visual surveillance, intrusion detection, anomaly detection and monitoring, industrial sites monitoring, detection-based tracking, autonomous vehicles, etc. However, numerous real-world scenarios such as dynamic background changes, illumination variations, shadows, challenging environmental conditions such as rainfall, haze, etc. make recognition of relevant moving objects a challenging task.

Two closely related tasks to MOR are moving object detection (MOD) or change detection [1-24] and object detection [25-54]. In MOD, the relevant motion information is identified through a class-agnostic segmentation of video frames into foreground and background regions. However, these methods do not categorize the foreground regions into their respective object classes. In object detection, the object instances are both localized and categorized into their respective classes (from a given set of classes i.e. cars, people, trucks, dogs, etc.). However, they work only on static images and do not take into account the temporal behavior over time to differentiate between moving and nonmoving objects.

For several applications, it is important to detect and classify only the moving objects. This is a different task in comparison to both object detection and MOD. Another important requirement for real-world applications is to detect and classify the moving objects online, i.e., while video is streaming. More specifically, the motion detector must not make use of future frames to analyze the current position and category of the moving objects. It must also be free from the requirement of target initialization from user as done for visual object tracking. Figure 1 demonstrates the difference between object detection, MOD and MOR.

In this paper, our goal is to identify both locations and corresponding categories of moving objects with a single-stage convolutional network. We also aim at retaining the property of online inference by using only the recent

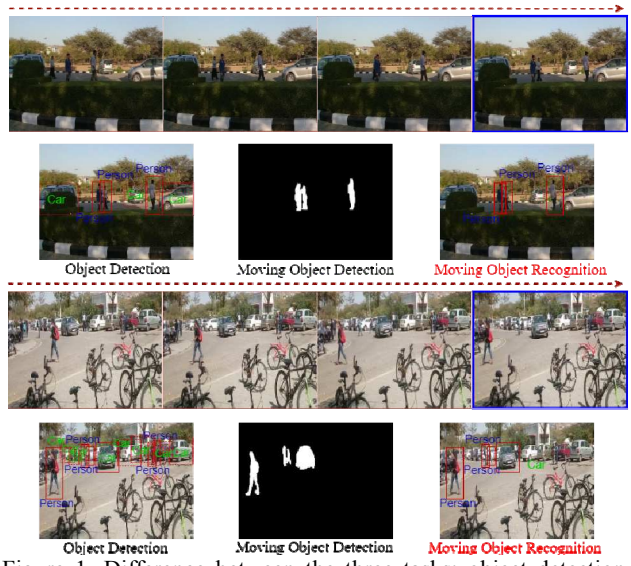

Figure 1. Difference between the three tasks: object detection, moving object detection (MOD) and MOR is depicted. The generic object detector detects all (moving and non-moving) object instances. Whereas, in MOD, only the pixel-wise changes are identified in class-agnostic manner. The MOR both detects and classifies the moving objects in the video. The same is shown in this figure.

history frames in live streaming videos. To achieve these goals, we design an online end-to-end one stage convolutional network MotionRec for MOR. In MOR, we first estimate the background representation from past history frames with a temporal depth reductionist (TDR) block. TDR estimates the background through a sequence of mean multi-spatial receptive feature (MMSR) modules. Each MMSR module selectively determines the probable salient background representation. Moreover, we produce contrasting features through assimilation of estimated background, temporal median and contemporary features from current frame. Multi-scale features are generated for these encoded maps. Moreover, in order to reinforce more accurate semantic features of the salient objects, multi-scale features are also extracted parallelly from the current frame. The temporal and spatial encodings from the two abovementioned parallelly extracted features are fused at matching scales to further generate the multi-level feature pyramids. These feature pyramids are used in the regression and classification blocks to perform MOR.

Our novel framework offers several advantages: (i) the proposed MotionRec solely relies on the recent video frame observations and operates online without any external updates, (ii) it produces labeled bounding boxes at approximately 5 frames per second and does not require bounding box initialization from user, (iii) since, labelling axis-aligned bounding boxes for moving objects require less computational resources in comparison to producing pixel-level estimates, our approach is well suited for training application specific moving object recognition models on customized datasets.

We also developed a new set of annotations with labelled axis-aligned bounding boxes for MOR. To the best of the authors' knowledge, no such benchmark dataset (with annotations) is available in the literature. We annotated 42,614 objects (14,814 cars and 27,800 person) in 24,923 video frames from videos collected from CDnet 2014 [55] for experimental analysis. We present qualitative and quantitative results of MotionRec over a defined unseen test set. Moreover, in order to analyze the importance of different blocks in MotionRec, we conduct multiple experiments for ablation analysis. From experimental results analysis, it is observed that the proposed MotionRec shows significant performance improvement over strong baselines based on RetinaNet detectors repurposed for MOR. The detailed literature on the existing methodologies for MOD and object detection are discussed in the following section.

2. Related Work

An extensive body of literature is available on MOD and object detection. In this section, we briefly discuss the most representative techniques for these two applications.

Moving Object Detection (MOD). The objective of a MOD method is to segment a video frame into foreground and background regions corresponding to object motion. Traditional approaches for MOD have used background subtraction methods to model the background behavior and identify the foreground region using various thresholding techniques. In past two decades, the parametric statistical models [1-3] have been widely adopted in background subtraction. However, most of the modern works [4-8] in background subtraction are inspired by the nonparametric background modelling approaches [9, 10]. The authors in [10] used a combination of three strategies for background maintenance. Moreover, an adaptive mechanism to update the pixel-wise decision thresholds and update rate was introduced in [5]. St-Charles et al. [6, 7] proposed to use spatiotemporal feature descriptors and adaptive feedback mechanism for background subtraction. Several other works [4, 11-15] have also been proposed to further improve the performance.

Various CNN based techniques have also been presented for MOD. Many attempts in this domain leverage off-the-shelf pre-trained CNNs and integrate them with hand-crafted background modelling techniques for temporal feature encoding [16-19]. To learn the local changes, certain methods [16, 20-22] have divided the frames and background model into patches and trained the model with concatenated input. Chen at al. [23] designed an attention ConvLSTM to model pixel-wise changes over

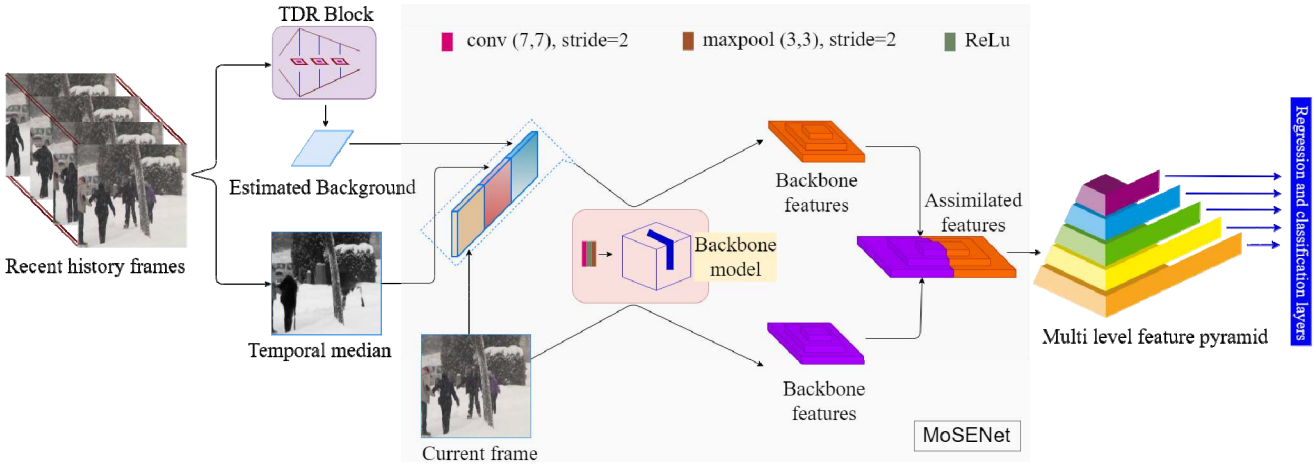

Figure 2. Schematic illustration of the proposed MotionRec framework.

time. Moreover, a conditional Generative Adversarial Network (cGAN) was proposed to learn the motion features for MOD in [24].

Object Detection. The modern state-of-the-art detectors can be categorized into two-stage and one-stage methods. In the two-stage methods, as pioneered by Uijlings et al. [25], first stage generates a sparse set of possible locations for the objects and perform sampling based on the image structure and the second stage classifies the candidate proposals into foreground classes / background. R-CNN by Girshick et al. [26] significantly improved the accuracy by introducing the use of CNNs in the second stage classification. Further improvements over R-CNN were presented both in terms of speed [27, 28] and accuracy [29, 30]. Ren et al. [31] proposed Faster R-CNN by introducing region proposal networks (RPN) for integrated proposal generation and trained the detector with a single CNN. Several extensions and variants of this framework [32-39] have been proposed since for improved performance. He et al. proposed Mask R-CNN [40] to generate instance segmentation masks of the detected object along with the spatial coordinates.

Sermanet et al. [41] designed one of the first one-stage detector OverFeat based on deep networks. Subsequently, more popular YOLO [42, 43] and SSD [44, 45] have ushered in considerable interest in one-stage detectors. These detectors have shown remarkable speed although trail the two-stage methods for accuracy. The seminal work RetinaNet [46] improved the accuracy of one-stage detectors while maintaining the speed by designing a focal loss function. Huang et al. [47] demonstrated comparative analysis of different object detectors for speed/accuracy trade-offs. The concept of multi-scale feature representation [38] have been successfully used in the recent detectors [48-51] to achieve comparable or even better accuracy in comparison to the two-stage detectors. A more detailed comparative analysis of object detection techniques can be found in [52, 53].

In [56], the authors define MOR as classifying video frames into moving or nonmoving classes based on object movements. However, in this paper we take a broader definition of MOR to further compute both spatial coordinates and class labels of moving object in each video frame. To the best of our knowledge, this is a first attempt to develop an end-to-end convolutional network for online MOR.

3. MotionRec

The proposed MotionRec consists of temporal depth reductionist (TDR) block, motion saliency estimation (MoSENet) network, regression and classification blocks. The overall architecture of MotionRec is shown in Figure 2. We discuss the functionality of these constituent blocks in the following subsections.

3.1. Temporal Depth Reductionist Block

In MotonRec, we first estimate the background from past history frames. Thereafter, we identify the motion information by comparing estimated background with the current frame in MoSENet block. For background estimation, we designed TDR block which is completely trainable as a part of the end-to-end MotonRec framework as shown in Figure 2. We only use grayscale images which are sufficient to represent temporal motion encodings. In TDR, the background is learned through a sequence of mean multi-spatial receptive feature (MMSR) modules.

Each MMSR module captures the average response from receptive fields of size 1x1, 3x3 and 5x5. The inspiration behind using multiple filter sizes comes from theoretical propositions and corresponding experimental success of algorithms presented in [5-7, 10-12, 57-59]. In [5, 10, 12], pixel-based background model estimation

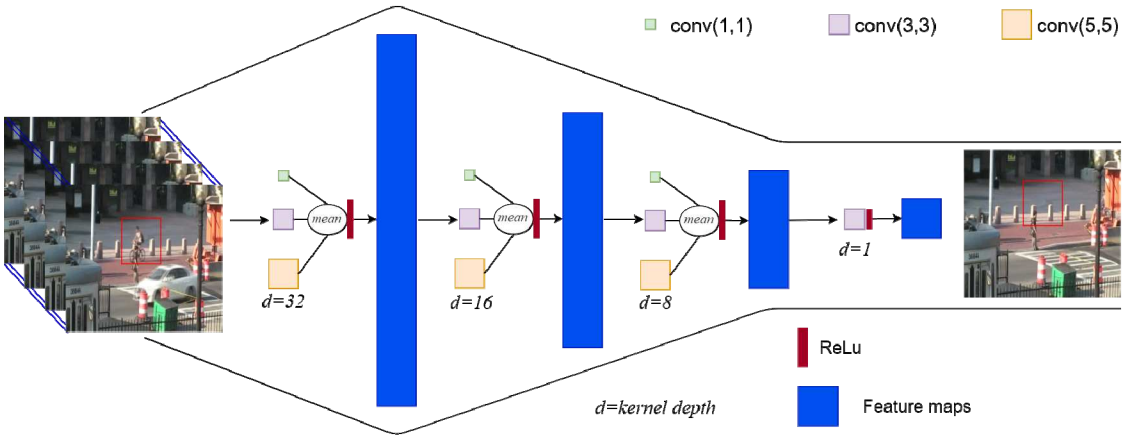

Figure 3. Temporal depth reductionist block for background estimation from recent temporal history. All convolutions (conv) are applied with stride (1, 1).

strategies were proposed. In [11, 57-59], local patterns extracted from 3x3 region provided the discriminative texture features to capture background statistics. Furthermore, methods in [6, 7] relied upon both the features extracted from 5x5 region and pixel-level intensities for robust background characterization. Thus, to mimic similar features through learnable parameters, we proposed MMSR to incorporate responses from three different levels of receptive fields. Moreover, by taking average of thee three responses, we ensure adaptability in the network to handle different change scenarios. These MMSR blocks with decreasing feature depths selectively determines the probable salient background representation. Finally, through these reductionist stages, we estimate a single depth background map. The detailed TDR block architecture is depicted in Figure 3.

Let's define a convolutional kernel as $\kappa_{x,h,w}$ where the parameters *h, w, x* represents height, width and kernel depth respectively. The past temporal history stack is denoted as P_T having height, width and number of frames as *H*, *W* and *T* respectively. An MMSR block denoted Ψ can be defined through Eq. (1)

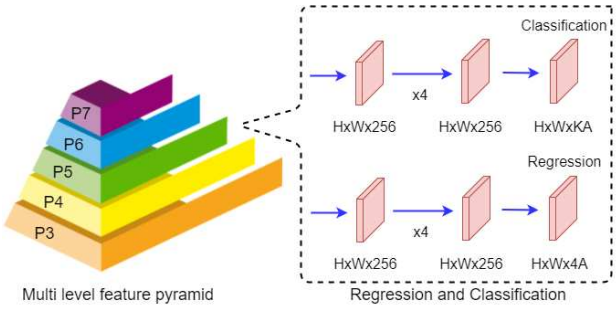

Figure 4. The regression and classification subnets at each pyramid level. P3 to P7 are the feature levels used for the final prediction. H×W is the height and width of feature maps. *A: anchors, K: object classes*.

$$\Psi_x(z) = \frac{1}{3}\sum_{i=1}^{3} \Re(\kappa_{x,2i-1,2i-1} \otimes z) \quad (1)$$

where $\otimes$ denotes the convolution operation, stride = (1, 1) and $\Re(\cdot)$ is the rectified linear unit (ReLu) activation function.

We compute the final TDR feature response using these MMSR blocks as given in Eq. (2) and Eq. (3).

$$TDR = \zeta_1(\Psi_8(\Psi_{16}(\Psi_{32}(P_T)))) \quad (2)$$

$$\zeta_1(z) = \Re(\kappa_{1,3,3} \otimes z) \quad (3)$$

Thus, we reduce the feature map depth by using 32, 16, 8 and 1 kernel depths respectively. The final response is single depth estimated background map.

3.2. Motion Saliency Estimation Network

In MoSENet block, we assimilate estimated background with a temporal median (M_T) and current frame (I). The pixel-wise temporal median of recent observations fortifies the background confidence by supplementing TDR response with statistical estimates. This enhances robustness of background model for different real-world scenarios such a dynamic background changes, bad weather, shadows, etc. These assimilated features are computed using Eq. (4)

$$AsFeat = [TDR, M_T, I] \quad (4)$$

Since AsFeat contains contrasting features i.e. background maps and current frame features. It provides crucial encoding to construct coarse motion saliency maps. We then extract base features from AsFeat for higher level feature abstractions. The base features are extracted from three layers (C_3, C_4, C_5) of ResNet residual stage as in [46]. Moreover, in order to delineate semantically

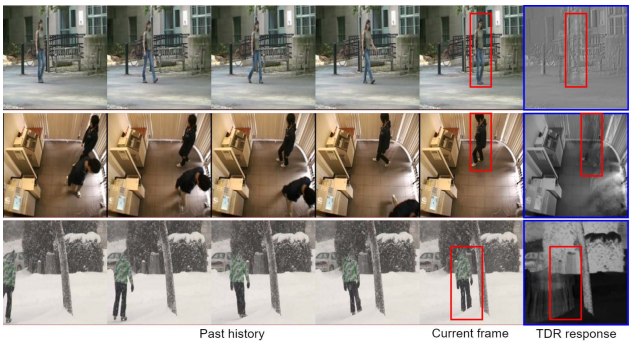

Figure 5. Visualization of the TDR background estimation block. The last column represents the background feature map estimated by TDR block from the past history frames for each row.

accurate shape representation for object categorization, we propose to incorporate certain reinforcements by parallelly extracting ResNet features for the current frame as well. The feature maps at same scales are combined for both temporal and spatial saliency aware feature representation. The MoSENet feature response is computed using Eq. (5).

$$MoSENet = [resnet(AsFeat), resnet(I)] \quad (5)$$

where $resnet(\cdot)$ returns the base features from ResNet residual stages.

3.3. Regression and Classification

The features from MoSENet are used to construct multi-level feature pyramids. We follow FPN [38] to detect different sizes of objects at different levels of feature maps. More specifically, we generate five levels of feature maps: P3, P4, P5, P6, P7. P3, P4 and P5 are computed using the backbones feature maps C3, C4 and C5 followed by a 1×1 convolutional layer with the lateral connections as given in [38]. Similarly, P6 and P7 are computed by applying convolution with the stride=2 on P5 and P6 respectively. At each pyramid level, two subnets are connected to perform bounding box regression and object classification as depicted in Figure 4. At each level of pyramidal feature map, we set 9 anchors. The detection and classification layers follow similar configurations as in [49].

3.4. Visualization

We show the visualizations of TDR block and the pyramid layers for analyzing the proposed MOR network behavior. The TDR response for 3 different sample videos is visually represented in Figure 5. Here, we can see that the background is robustly estimated from recent temporal history. Moreover, the visualizations for 4 pyramid levels are depicted in Figure 6. It is clear from Figure 6, that the MOR features are quite accurately being localized through these shallow and deep layers.

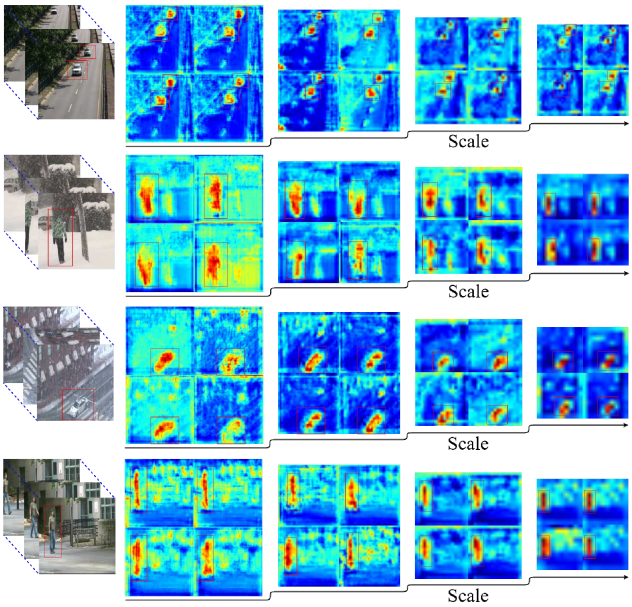

Figure 6. Visualization of pyramid levels P3, P4, P5 and P6. The relevant motion saliencies of moving objects are highlighted using red boxes. We do not show visualization of P7 due to very small size.

3.5. Network Configurations

MotionRec takes two tensors of shape 608x608xT (past temporal history) and 608x608x3 (current frame) as input and returns the spatial coordinates with class labels for moving object instances. While training MotionRec, we use the ResNet50 backbone pretrained over the ImageNet dataset. For regression and classification, smooth L1 and focal loss functions are used respectively. The training loss is the sum of above mentioned two losses. The loss gradients are backpropagated through TDR blocks as well.

Training and Inference. MotionRec forms a single-stage fully connected network which ensures online operability and fast speed. The entire framework is implemented in Keras with Tensorflow backend. Training is performed with batch size=1 over Nvidia Titan Xp GPU system. We use adam optimizer with initial learning rate set to $1x10^{-5}$. Unless otherwise specified, all models are trained for approximately 500k iterations. We only use horizontal image flipping for data augmentation.

Similar to training, inference involves simply giving current frame and recent T temporal history frames as input to the network. Only few past frames (T=*10/20/30*) are required, enabling online moving object recognition. Top 1000 prediction scores per pyramid level are considered after thresholding detector confidence at 0.05. The final detections are collected by combining top predictions from all levels and non-maximum suppression with a threshold of 0.5.

Videos	#Frames	# Car	# Person	# Objects
blizzard	1,000	2,496	0	2,496
skating	1,964	306	4,392	4,698
snowfall	1,585	1,587	0	1,587
wetsnow	2,126	1,556	1,594	3,150
highway	1,473	4,818	0	4,818
pedestrian	572	0	657	657
PETS2006	1,180	0	2,872	2,872
fountain01	313	418	0	418
fountain02	175	240	0	240
fall	1,466	990	1,786	2,776
tramstop	2,535	1,031	4,356	5,387
backdoor	667	0	809	809
busstation	895	0	1,662	1,662
copymachine	2,182	0	3,150	3,150
cubicle	3,053	0	3,798	3,798
peopleinshade	531	0	615	615
Train Data	*21,717*	*13,442*	*25,385*	*38,827*
sofa	1,809	0	1,975	1,975
parking	706	624	440	1,064
bungalows	691	748	0	748
Test Data	*3,206*	*1,372*	*2,415*	*3,787*

Table 1. Summary description of the dataset used in our experiments for training and evaluation.

Method\mAP50	Depth	Sofa	Par.	Bun.	Overall
T_RetinaNet (resnet50)	10	50.5	49.0	68.9	56.1
	20	72.1	47.2	77.8	65.7
	30	72.0	47.2	88.7	69.3
T_RetinaNet (mobilenet_v2)	10	0.16	0.11	9.12	3.13
	20	10.98	0.42	10.10	7.16
	30	35.09	0.30	42.57	26.00
MotionRecV1	10	56.6	67.2	69.0	64.3
	20	**82.2**	39.4	79.7	67.1
	30	80.5	**69.5**	89.0	**79.7**
MotionRecV2	10	75.2	43.6	69.2	62.7
	20	79.0	49.8	**90.3**	73.0
	30	70.3	61.2	84.6	72.0

Table 2. MOR performance comparison of the proposed MotionRec models (V1 and V2) with the baseline results of temporal RetinaNet (T_RetinaNet) models. T_RetinaNet models are trained by forwarding sequence of frames (10/20/30) in the input layer. Best results are highlighted in **bold**. Depth: temporal history depth. All results are computed in terms of mean AP with 50% IoU. *Depth: temporal history depth, Par.: Parking, Bun.: Bungalows*.

4. Experiments

In this section we first discuss about the generated data annotations and evaluation metrics for MOR. We then proceed with quantitative and qualitative analysis of the results with baseline model comparisons to demonstrate the strengths of our MotionRec method. Moreover, in order to analyze the contribution of different modules in MotionRec, we perform multiple ablation experiments.

4.1. Dataset, Evaluation Metrics and Baseline Models

Dataset. Due to lack of available benchmark datasets with labelled bounding boxes for MOR, we created a new set of ground truths by annotating 42,614 objects (14,814 cars and 27,800 person) in 24,923 video frames from CDnet 2014 [55]. We selected 16 video sequences having 21,717 frames and 38,827 objects (13,442 cars and 25,385 person) for training. For testing, 3 video sequences with 3,206 frame and 3,787 objects (1,372 cars and 2,415 person) were chosen. We created axis-aligned bounding box annotations for moving object instances in all the frames. Since, there has been no previous attempt to perform such task, we defined our own train and test divisions for qualitative and quantitative evaluation. The complete details about the created dataset are given in Table 1.

Evaluation metrics. Since the results are computed as spatial coordinates and class labels of moving object instances in every frame. Therefore, to measure MOR performance, we use the standard average precision (AP) [25-53] metrics. The average precision metric AP50 counts a predicted object instance as true positive if it has at least 50% intersection-over-union (IoU) with the corresponding ground truth object instance. Performance is measured across two classes: car and person.

Baseline models. Since this is a first attempt for MOR in videos, we designed two baselines for comparative analysis. We adapted the RetinaNet object detector to take stack of multiple frames (temporal history) as input and produce the detection estimates for the current frame. Thus, the network is repurposed for MOR in videos. We further created two different models based on resnet50 and mobilenet_v2 backbone respectively. We denote these models as T_RetinaNet (resnet50) and T_RetinaNet (mobilenet_v2) respectively. Since RetinaNet based methods have achieved very high performance for the task of generic object detection, these models serve as very strong baselines for comparative analysis of our work.

4.2. Quantitative Results

In addition to the proposed MotionRec (also denoted as MotionRecV1), we designed MotionRecV2 by removing the parallel feature extraction (from current frame) layer from MotionRec. Thus, we could also evaluate the effect of directly using AsFeat without the reinforcements of base features extracted from the current frame. We present the quantitative results of the proposed MotionRecV1, MotionRecV2 and the baseline models in Table 2.

From Table 2, it is evident that ehe proposed methods outperform the T_RetinaNet models by a good margin in

Method	Depth	FPS	# Parameters	Model Size
MotionRecV1	10	5.2	65.43 M	263.7 MB
	20	3.6	65.44 M	
	30	2.0	65.45 M	
MotionRecV2	10	5.6	36.34 M	146.4 MB
	20	3.4	36.35 M	
	30	2.0	36.36 M	

Table 3. Inference speed, number of parameters and inference model size comparison between MotionRecV1 and MotionRecV2 at three different temporal history depth.

each of the test video sequences. The baseline T_RetinaNet with mobilenet_v2 backbone performed very poorly. Whereas, the resnet50 based model yielded reasonable results. More specifically, the best performing MotionRec achieves 10.1%, 20.5% and 1.6% better mAP50 over best performing T_RetinaNet for sofa, parking and bungalow videos respectively. Overall, the proposed MotionRecV1 outperforms baseline T_RetinaNet (resnet50) by 8.2%, 1.4% and 10.4% in terms of mAP50 for depths 10, 20 and 30 respectively. Similarly, MotionRecV2 improves upon the baseline T_RetinaNet (resnet50) by 6.6%, 7.3% and 2.7% mAP50 for depths 10, 20 and 30 respectively. The mAP at different IOUs for MotionRecV1 is further analyzed through Figure 7. The IoU vs mAP graph for MOR across different temporal depths for each test video is depicted in Figure 7 (a), Figure 7 (b) and Figure 7 (c) respectively. The mAP is computed at IoU thresholds 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 and 0.8. We also show the overall average performance through graph in Figure 7 (d). It is clear that if we could lower the IoU threshold, we can recognize higher number of moving objects. However, it might also increase the number of false detections. The decision for the same can be taken according to the demands of real-world applications.

We also tabulate the inference speed, compute and memory requirements of the proposed models in Table 3. MotionRecV2 reduces the model size and number of trainable parameters as shown in Table 3. It consists of almost half the number of parameters to be trained as compared to MotionRecV1. However, both these models have similar inference speed for corresponding temporal depths. This is due to computational independency between the two parallel feature extraction layers in MotionRecV1. Both extract base features from two independent ResNet50 models. Therefore, removing one parallel layer in MotionRecV2 reduces number of trainable parameters, model size but doesn't substantially affect the inference speed in GPU environment. Since, MotionRecV1 achieves better mAP as compared to MotionRecV2. Overall, MotionRecV1 outperforms

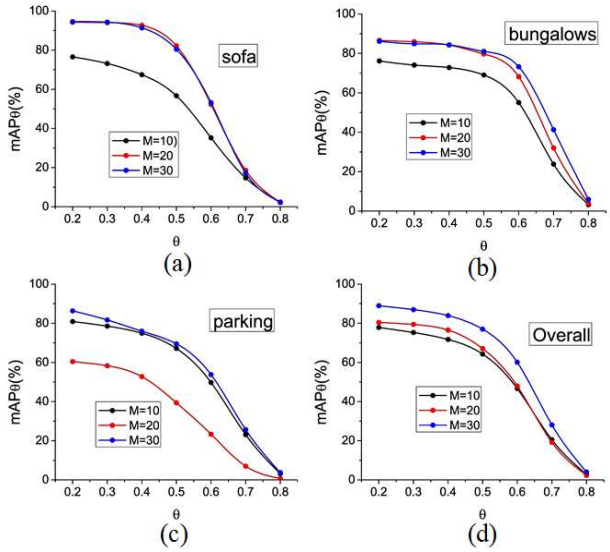

Figure 7. Moving object detection mAPθ across different IoU thresholds θ over (a) sofa, (b) bungalows, (c) parking videos and (d) average across all video sequences. M is the temporal depth of input layer

d=10\mAP	sofa	parking	bungalow	overall
MotionRec_RM	32.87	0.02	32.47	21.78
MotionRec_RI	19.17	1.87	18.98	13.34

Table 4. Ablation experiments with temporal depth=10 for different variants of MotionRecV1. *MotionRec_RM: Removal of temporal median (M_T) in AsFeat, MotionRec_RI: Removal of current frame (I) in AsFeat.*

MotionRecV2 when all the performance measures are taken into consideration.

4.3. Ablation Studies

We conduct multiple ablation experiments to analyze the contribution of different modules in MotionRecV1. In the previous section, we have already discussed a variant of the proposed method (MotionRecV2) by dropping the parallel features in the network. Similarly, the effect of temporal depths can be inferred from Table 2 and Table 3. We can see that as we increase the *temporal depth*, the performance also increases in terms of mAP but the network efficiency decreases in terms of compute and memory parameters.

We further quantify the influence of the following components in AsFeat block: temporal median (M_T) and current frame (I). The AsFeat block is a significant part of MoSENet which generates the contrasting features to learn the salient motion information in the subsequent layers. As given in Table 4, Removal of M_T led to approximately 42% decrease in overall mAP. Similarly, removing current frame I decreased the mAP by approximately 50%. These ablation experiments further provide evidence in support

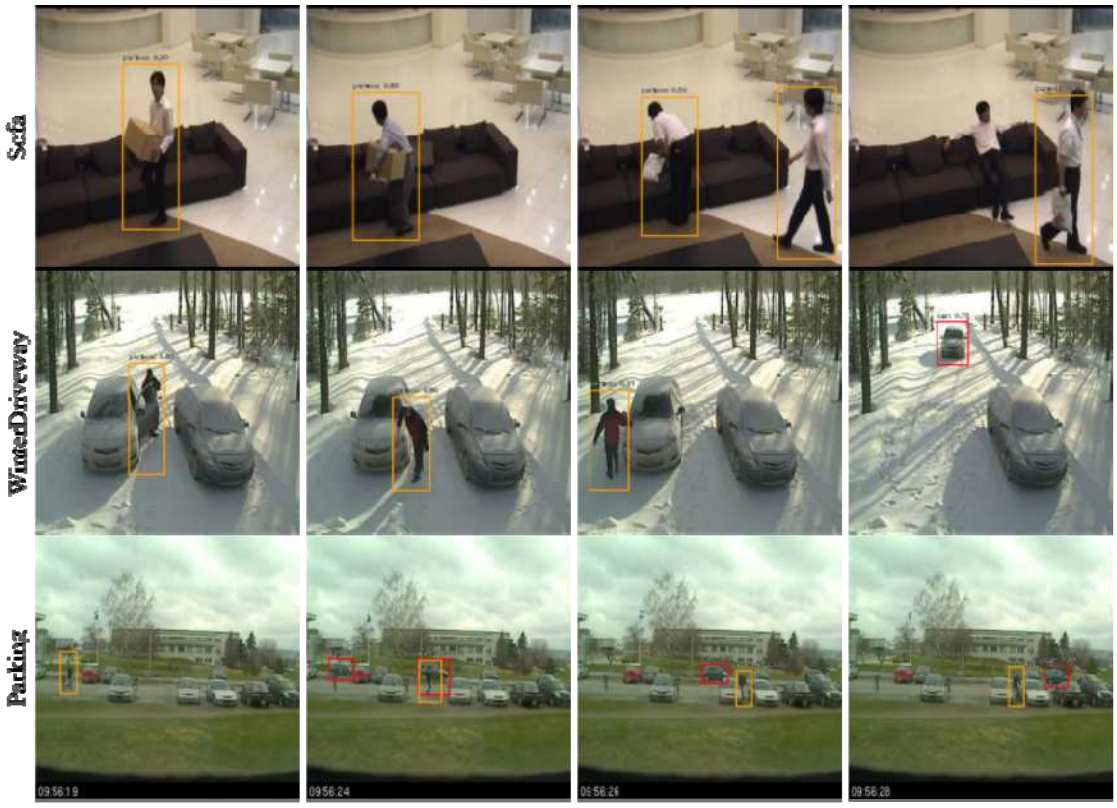

Figure 8. Qualitative results of our method for unseen video sequences sofa, winterdriveway and parking from CDnet 2014 dataset.

of our original model designs. All the experiments were conducted for temporal depth=10 and the same can be generalized to other depths as well.

4.4. Qualitative Results

We show the qualitative results of our approach on both indoor and outdoor scenarios in Figure 8. The MotionRec produces accurate MOR bounding boxes even in diverse object movements scenarios. For example, in *parking* video, from rows of parked cars, people are walking and cars are passing by. Two different classes (car and person) of objects cross-over at various points in time. They also overlap each other at multiple instances of time. All these scenarios are handled quite well. In *sofa* video, people are moving around and stopping in between at various point of time. Sometimes, two persons are crossing over and one of them sits down for few seconds. All these diverse movements are quite accurately detected. Similarly, in *winterDriveway*, both static and moving objects (car and person) are present. The MotionRec is able to distinguish between static and moving objects in all these scenarios and the same can be seen in Figure 8. Our model quite accurately recognizes partially occluded moving objects or when two objects cross each other as shown in row-3 (column-2) of Figure 8.

5. Conclusion

A novel deep learning framework MotionRec for online moving object recognition (MOR) is presented in this paper. We also generated a new set of MOR labels by annotating 42,614 objects in 24,923 video frames from CDnet 2014. The axis-aligned bounding boxes were used for moving objects which requires less computational resources than producing pixel-level estimates. To the best of the authors' knowledge, this is a first attempt to design single-stage framework for online MOR. The proposed method significantly outperforms two strong baselines of temporal RetinaNet. Our work shows that with a robust CNN design and limited amount of training data, we are able to obtain surprisingly accurate results. Further improvements in architectural design with additional labelled data have the potential to improve performance both in speed and accuracy.

Acknowledgements

The work was supported by the DST-SERB project #SERB/F/9507/2017.The authors would like to thank the members of Vision Intelligence Lab for their valuable support.

References

[1] Z. Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. In Proc. IEEE Int. Conf. Pattern Recognit., 2:28-31, 2004.

[2] S. Varadarajan, P. Miller, and H. Zhou. Spatial mixture of Gaussians for dynamic background modelling. In Proc. IEEE Int. Conf. Adv. Video Signal Based-Surveill., pages 63-68, 2013.

[3] C. Stauffer, and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2:246-252, 1999.

[4] S. Jiang, and X. Lu. WeSamBE: A weight-sample-based method for background subtraction. IEEE Trans. Circuits Syst. Video Technol., 28(9): 2105-2115, 2018.

[5] M. Hofmann, P. Tiefenbacher, and G. Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, pages 38-43, 2012.

[6] P. L. St-Charles, and G. A. Bilodeau. Improving background subtraction using local binary similarity patterns. In Proc. IEEE Winter Conf. Appl. Comput. Vis., pages 509-515, 2014.

[7] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin. SuBSENSE: A universal change detection method with local adaptive sensitivity. IEEE Trans. Image Process., 24(1): 359-373, 2015.

[8] S. Bianco, G. Ciocca, and R. Schettini. Combination of video change detection algorithms by genetic programming. IEEE Trans. Evolutionary Computation, 21(6): 914-928, 2017.

[9] H. Wang, and D. Suter. A consensus-based method for tracking: Modelling background scenario and foreground appearance. Pattern Recognit., 40: 1091-1105, 2007.

[10] O. Barnich, and M. V. Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. IEEE Trans. Image Process., 20(6): 1709-1724, 2011.

[11] M. Mandal, M. Chaudhary, S. K. Vipparthi, S. Murala, A. B. Gonde, and S. K. Nagar. ANTIC: ANTithetic Isomeric Cluster Patterns for Medical Image Retrieval and Change Detection. IET Comput. Vis., 13(1): 31-43, 2019.

[12] M. Mandal, P. Saxena, S. K. Vipparthi, and S. Murala. CANDID: Robust Change Dynamics and Deterministic Update Policy for Dynamic Background Subtraction. In Proc. IEEE Int. Conf. Pattern Recognit., pages 2468-2473, 2018.

[13] S. Javed, A. Mahmood, S. Al-Maadeed, T. Bouwmans, and S. K. Jung. Moving object detection in complex scene using spatiotemporal structured-sparse RPCA. IEEE Trans. Image Process., 28(2): 1007-1022, 2019.

[14] L. Li, Q. Hu, and X. Li. Moving object detection in video via hierarchical modeling and alternating optimization. IEEE Trans. Image Process., 28(4): 2021-2036, 2019.

[15] Z. Zhong, J. Wen, B. Zhang, and Y. Xu. A general moving detection method using dual-target nonparametric background model. Knowledge-Based Systems, 164: 85-95, 2019.

[16] C. Lin, B. Yan, and W. Tan. Foreground Detection in Surveillance Video with Fully Convolutional Semantic Network. In Proc. IEEE Int. Conf. Image Processing, pages 4118-4122, 2018.

[17] L. A. Lim, and H. Y. Keles. Foreground Segmentation Using a Triplet Convolutional Neural Network for Multiscale Feature Encoding. Pattern Recognit. Lett. 112: 256:262, 2018.

[18] D. Zeng, and M. Zhu. Multiscale Fully Convolutional Network for Foreground Object Detection in Infrared Videos. IEEE Geosci. Remote Sens. Lett., 15(4): 617-621, 2018.

[19] K. Lim, W. D. Jang, and C. S. Kim. Background subtraction using encoder-decoder structured convolutional neural network. In Proc. IEEE Int. Conf. Adv. Video Signal Based-Surveill, pages 1-6, 2017.

[20] T. P. Nguyen, C. C. Pham, S. V. U. Ha, and J. W. Jeon. Change Detection by Training a Triplet Network for Motion Feature Extraction. IEEE Trans. Circuits Syst. Video Technol., to be published, doi: 10.1109/TCSVT.2018.2795657.

[21] M. Babaee, D. T. Dinh, and G. Rigoll. A deep convolutional neural network for video sequence background subtraction. Pattern Recognit., 76:635-649, 2018.

[22] M. Braham, and M. Van Droogenbroeck. Deep background subtraction with scene-specific convolutional neural networks. In Proc. IEEE Int. Conf. Syst., Signal. Image Process., pages 1-4, 2016.

[23] Y. Chen, J. Wang, B. Zhu, M. Tang, and H. Lu. Pixel-wise deep sequence learning for moving object detection. IEEE Trans. Circuits Syst. Video Technol., to be published, doi: 10.1109/TCSVT.2017.277031.

[24] M. C. Bakkay, H. A. Rashwan, H. Salmane, L. Khoudour, D. Puigtt, and Y. Ruichek. BSCGAN: Deep Background Subtraction with Conditional Generative Adversarial Networks. In Proc. IEEE Int. Conf. Image Process., pages 4018-4022, 2018.

[25] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. Int. J. Comput. Vis., 104(2): 154-171, 2013.

[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 580-587, 2014.

[27] R. Girshick. Fast R-CNN. In Proc. IEEE Int. Conf. Comput. Vis., pages 1440-1448, 2015.

[28] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell., 37(9): 1904-1916, 2015.

[29] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 2147-2154, 2014.

[30] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In Adv. Neur. Inf. Process. Syst., pages 1990-1998, 2015.

[31] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell., 39(6): 1137-1149, 2017.

[32] A. Shrivastava, A. Gupta, and R. Girshick. Training region based object detectors with online hard example mining. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 761-769, 2016.

[33] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, and H. Lu. Couplenet: Coupling global structure with local parts for object detection. In Proc. IEEE Int. Conf. Comput. Vis., pages 4126-4134, 2017.

[34] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In Proc. IEEE Int. Conf. Comput. Vis., pages 764–773, 2017.

[35] B. Singh, and L. S. Davis. An analysis of scale invariance in object detection–snip. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 3578–3587, 2018.

[36] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta. Beyond skip connections: Top-down modulation for object detection. *arXiv:1612.06851*, 2016.

[37] L. Tychsen-Smith, and L. Petersson. Improving object localization with fitness NMS and bounded iou loss. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 6877-6885, 2018.

[38] T. -Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and´ S. Belongie. Feature pyramid networks for object detection. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 2117-2125, 2017.

[39] Z. Cai, and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 6154-6162, 2018.

[40] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In Proc. IEEE Int. Conf. Comput. Vis., pages 2961-2969, 2017.

[41] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In Proc. Int. Conf. Learn. Representat., 2014.

[42] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 779-788, 2016.

[43] J. Redmon and A. Farhadi. YOLO9000: Better, faster, stronger. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 7263-7271, 2017.

[44] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. SSD: Single shot multibox detector. In Proc. Eur. Conf. Comput. Vis., pages 21-37, 2016.

[45] C. -Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. DSSD: Deconvolutional single shot detector. arXiv:1701.06659, 2016.

[46] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In Proc. IEEE Int. Conf. Comput. Vis., pages 2980-2988, 2017.

[47] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 7310-7311, 2017.

[48] J. Redmon, and A. Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[49] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen. RON: reverse connection with objectness prior networks for object detection. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 5936-5944, 2017.

[50] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Single-shot refinement neural network for object detection. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 4203-4212, 2018.

[51] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling. M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network. arXiv preprint arXiv:1811.04533, 2018.

[52] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen. Deep learning for generic object detection: a survey. arXiv preprint arXiv:1809.02165, 2018.

[53] Z. -Q. Zhao, P. Zheng, S. T. Xu, and X. Wu. Object detection with deep learning: a review, IEEE Trans. Neur. Network. Learn. Syst., to be published, doi: 10.1109/TNNLS.2018.2876865.

[54] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 770–778, 2016.

[55] Y. Wang, P. M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar. CDnet 2014: an expanded change detection benchmark dataset. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, pages 387-394, 2014.

[56] T. He, H. Mao, and Zhang Yi. Moving object recognition using multi-view three-dimensional convolutional neural networks. Neural Comput. Applications., 28(1): 3827-3835, 2017.

[57] M. Heikkila, and M. Pietikäinen. A texture-based method for modeling the background and detecting moving objects. IEEE Trans. Pattern Anal. Mach. Intell., 28(4): 657-662, 2006.

[58] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikäinen, and S. Z. Li. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 1301-1306, 2010.

[59] H. Han, J. Zhu, S. Liao, Z. Lei, and S. Z. Li. Moving object detection revisited: Speed and robustness. IEEE Trans. Cir. Syst. Video Technol., 25(6): 910-921, 2014.