MBSNet: To Distinguish Motion From Stillness for Airport Traffic Safety

Xiang Zhang, Yingqi Tang, Maozhang Zhou, Celimuge Wu, *Senior Member, IEEE*,
and Zhi Liu, *Senior Member, IEEE*

***Abstract*— Background subtraction forms the basis of many safety applications in airport traffic management, such as the visual conflict warning system. However, deep learning methods often mistakenly identify stationary aircraft as foreground, mainly because they prioritize learning appearance over motion features. This means that stationary aircraft with a similar appearance to moving ones are often incorrectly classified as foreground. To address this issue, a Motion-enhanced Background Subtraction Network (MBSNet) is proposed in this paper. MBSNet is designed to focus more on motion information within an encoder-decoder framework. Firstly, a Motion Augmentation Encoder Module (MAEM) is introduced, which generates a clean background frame without foreground from previous frames. This module compares the background frame with the current frame containing moving objects, indirectly enhancing the motion component in the encoded features. Because targets on the airport ground are relatively sparse, MAEM ensures a clean background image. Secondly, a Motion Accumulation Decoder Module (MADM) is designed, which accumulates motion-augmented features from the current frame and past frames based on feature dissimilarity measurement. Since aircraft exhibit consistent motion patterns, such as continuous straight travel with occasional turns, MADM further enhances the motion component in the accumulated feature vector. Finally, MBSNet is evaluated on the AGVS dataset, and our experiments demonstrate the effectiveness of the proposed method for airport background subtraction.**

***Index Terms*— Airport traffic safety, background subtraction, motion and stillness distinction.**

I. Introduction

ENSURING safety on the roads is a crucial aspect of transportation [1], and this holds especially true for airports, where safety is paramount. However, as the global civil aviation industry continues to expand rapidly, airports are becoming more crowded, leading to an increase in safety incidents. For instance, on January 25, 2023, there was a collision between an aircraft and a vehicle at Narita Airport in Japan, and on February 3, 2023, two aircraft collided at Newark Liberty International Airport in New Jersey, USA. Background subtraction [2] plays a vital role in many intelligent video applications aimed at enhancing airport traffic safety. Take the Visual Conflict Warning System (VCWS) depicted in Fig. 1, for example. Initially, background subtraction is used to isolate moving targets, followed by measuring the minimum distance between these targets to determine if a warning should be issued. It's evident that the accuracy of background subtraction significantly influences the overall performance of the system.

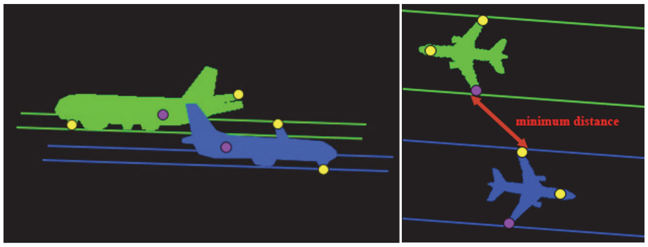


Fig. 1. The principle of VCWS. The moving targets are first segmented and then some feature points are detected or predicted. The subsequent task involves computing the shortest distance between the two targets from a top view and determining whether an alarm should be triggered.

Background subtraction assumes that the background is known a priori, and the target that has relative motion with the background is defined as the foreground. In an airport setting, the established reference is the airport ground, with all ground-moving targets considered as foreground objects. Deep learning-based background subtraction [3] has garnered significant interest lately due to its precise segmentation capabilities. However, it struggles with discerning between motion and stillness within the airport environment. As shown in Fig. 2, some stationary aircraft are misclassified as moving targets. In this paper, this issue is referred to as the Motion and Stillness Distinction (MSD) problem, which presents itself in two distinct cases,

- Case1: MSD between different objects. A stationary object with a resemblance to a moving one is classified as foreground.
- Case2: MSD of the same object. That is, to judge a target as a moving object after it stops or before the movement.

The MSD question violates the most basic goal of background subtraction, only detecting moving objects at the current moment. There are two reasons for the MSD problem in

Manuscript received 4 September 2023; revised 17 March 2024 and 8 June 2024; accepted 8 July 2024. This work was supported in part by the National Science Foundation of China under Grant U1733111 and in part by the Project of Quzhou Municipal Government under Grant 2023D011. The Associate Editor for this article was Q. Wang. *(Corresponding author: Xiang Zhang.)*

Xiang Zhang, Yingqi Tang, and Maozhang Zhou are with the School of Information and Communication Engineering, Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: uestchero@uestc.edu.cn; yingqitang@std.uestc.edu.cn; maozhangzhou@std.uestc.edu.cn).

Celimuge Wu and Zhi Liu are with the Department of Computer and Network Engineering, The University of Electro-Communications, Tokyo 182-8585, Japan (e-mail: celimuge@uec.ac.jp; liuzhi@uec.ac.jp).

Digital Object Identifier 10.1109/TITS.2024.3439107

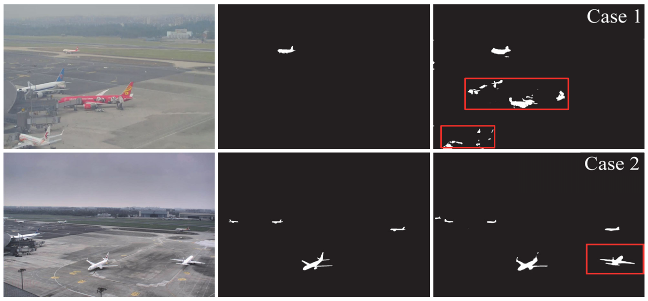


Fig. 2. The left column includes two frames extracted from the Airport Ground Video Surveillance (AGVS) dataset. The middle column includes the ground truth representation of moving objects. The right column includes the background subtraction outcomes generated by the Foreground Segmentation Network (FgSegNet) [4]. The AGVS dataset is specifically designed for airport background subtraction tasks, while FgSegNet operates on an encoder-decoder architecture. It's worth noting that FgSegNet erroneously identifies stationary aircraft (highlighted in red boxes) as foreground objects.

Fig. 2. First of all, the particularity of airport is a cause of MSD. Aircraft in civil airports, whether stationary or in motion, or from which aircraft manufacturing company, have a similar appearance, which results in a lot of Case1. Aircraft in airports often exhibit a stop-and-go motion pattern, which in turn leads to a lot of Case2. Secondly, and more importantly, the deep learning framework tends to learn the appearance features rather than the motion features. In this case, as long as the stationary object has a similar appearance to the moving object, it may be classified as foreground.

In this paper, a Motion-enhanced Background Subtraction Network (MBSNet) is proposed to address the MSD problem. The principle of MBSNet is to strive to enhance the motion component in the extracted features by the neural network, so as to force the network to pay more attention to the motion information for airport background subtraction. To achieve this purpose, MBSNet presents two new modules for motion feature augmentation and accumulation within the encoder-decoder framework. Firstly, a Motion Augmentation Encoder Module (MAEM) is developed where a pure background image without foreground is generated from previous frames. Then the background frame and the current frame is compared in the encoder module, thus indirectly augmenting the motion component in the encoded features. The background frame will be updated in real-time to adapt to the background change. Because the targets on the airport ground are relatively sparse, a clean background image can be guaranteed in MAEM through long-term observation. Secondly, a Motion Accumulation Decoder Module (MADM) is designed, where the motion-augmented features of the current frame and past frames by MAEM are accumulated together based on feature dissimilarity measurement before feature decoding. Considering the motion consistency of the aircraft, that is, continuous straight travel with occasional turns, the motion component will be further enhanced in the accumulated feature vector by MADM. The final experiments are conducted on the AGVS dataset [5] to verify the effectiveness of MBSNet. In summary, the main contributions of this paper are,

- We clearly define two cases of the MSD problem in background subtraction for the first time.

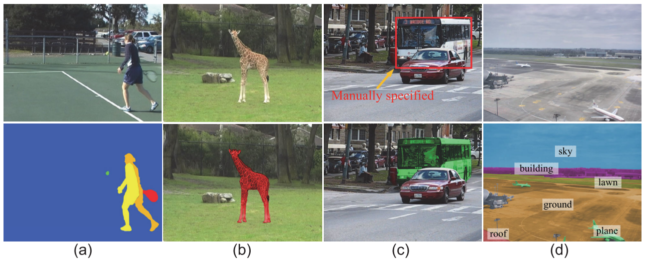


Fig. 3. (a) to (d): illustration of motion segmentation, automatic video object segmentation, semi-automatic video object segmentation, video semantic segmentation, respectively.

- We propose a method to solve the MSD problem in airport background subtraction based on some airport-specific prior information.

The rest of the paper is organized as follows. Section II introduces the related work, the details of MBSNet are described in Section III, the experimental results are shown in Section IV, and conclusion is in Section V.

II. Related Work

We initially delved into spatial-temporal segmentation, which encompasses background subtraction, motion segmentation, video object segmentation, video semantic segmentation, among others. Unlike background subtraction, where there's a known or implied background prior, other research directions lack this context, leading to distinct segmentation outcomes. For instance, motion segmentation may yield results termed moving objects, but these are actually a collection of layers or objects with varied motion characteristics (Fig. 3(a)), rather than a single moving entity relative to a static background. Automatic video object segmentation might produce dominant or general object segments (Fig. 3(b)), while semi-automatic video object segmentation involves manually specifying targets (Fig. 3(c)), which differ from the concept of moving objects in background subtraction. Video semantic segmentation aims to identify objects within predefined semantic categories (Fig. 3(d)), which also differ from moving objects. We briefly surveyed a few papers employing traditional or deep learning methods. For an in-depth exploration of background subtraction, consulting other referenced papers [6], [7] is recommended.

A. Traditional Method

Most traditional methods rely on statistical modeling [8], where statistical distributions are employed to match the changes in each pixel. Statistical modeling generally falls into two categories: parametric modeling and nonparametric modeling. Two critical considerations in statistical modeling are selecting relevant features and addressing variations in the background.

Parametric modeling tends to be more efficient but may not accurately capture the background distribution as effectively as nonparametric modeling since the probability distribution is predetermined. The most classic work of parametric modeling is the Gaussian Mixture Model (GMM) [9]. Stauffer and Grimson employed a mix of Gaussian distributions to model the probability that each pixel corresponds to either the foreground or the background. Several enhanced GMMs were later proposed to bridge the gap between the specified distribution and the underlying distribution, such as using adaptive update strategies [10].

When it comes to nonparametric methods, let's start with Kernel Density Estimates (KDE) [11], [12], where the probability distribution is directly estimated from samples without specifying a particular distribution model. Kim et al. proposed a method which clusters pixels into codewords to represent the background [13]. Barnich and Droogenbroeck developed Visual Background Extractor (ViBe) [14], which selects samples through the diffusion of adjacent pixels to capture fast-moving objects. Binary features are utilized in SuBSENSE [15] to detect subtle local changes, especially in cases where foreground and background have similar colors. St-Charles et al. proposed a nonparametric approach named PAWCS [16], which combines color intensities and texture features for modeling. Hybrid approaches like IUTIS [17] were also introduced to combine multiple models for more effective background subtraction, especially in challenging environments.

Commonly used features in statistical modeling are color, texture, motion or saliency descriptors [18], etc. There have also been some recent studies on background variations. Isik et al. presented a new background model in SWCD [19] by sliding window with dynamic parameters to adapt to background changes. Lee et al. proposed WisenetMD [20] to search for dynamic background region and then select samples with high confidence for background modeling. Besides statistical modeling, there are some other traditional methods like the Robust Principal Component Analysis (RPCA) [21] and subspace learning [22] based solutions.

B. Deep Learning Based Method

The deep learning approach outperforms the traditional method by a significant margin in terms of segmentation accuracy. Braham and Droogenbroeck [23] suggested a technique where Convolutional Neural Networks (CNNs) are employed to learn spatial features from image patches, which are subsequently utilized for background modeling. Wang et al. [24] presented a cascade structure of deep networks for background subtraction. The output of the first level was concatenated with the original frame and fed to the second level to refine the segmentation result. As stated in [24], such a cascade structure can be used to enforce the spatial coherence constraint so that better results can be obtained with more cascaded levels. The FgSegNet [4] had a triplet CNN for encoding and a transposed network for decoding. The triplet CNN operated in three different scales in parallel to get richer features. In the upgraded version FgSegNet2 [25], feature fusion is incorporated to boost the effectiveness of multi-scale features. Patil et al. [26] introduced an edge extraction method within the encoder-decoder framework to capture multi-scale foreground edge details for background subtraction.

It has been proven that the background subtraction performance under some challenges like camouflage can be improved by combining semantic information [32]. However, the production of high-quality semantic mask is time consuming. RT-SBS [33] attempted to request semantic information at a lower frequency, request incomplete semantic information, or reuse the previous semantic information to reduce the computation load due to semantic segmentation. Some other methods used Generative Adversarial Network [34] or Multiscale Network [35] to capture appearance cues for background subtraction. In addition, scholars also discussed the generalization problem of background subtraction based on deep learning, and tried to propose universal methods that can be used in any test scenario [36].

To address the challenge of unseen videos in background subtraction, Tezcan et al. [27] presented a fully-convolutional neural network based method, where the input consisted of two reference backgrounds at different time scales along with the semantic information. A key feature of this method is that the training and test sets were composed of frames from different videos. Later, spatial-temporal data augmentation is presented in [28] to replace the data augmentation step in [27]. This data augmentation method could mimic more challenges in background subtraction, e.g. Pan-Tilt-Zoom (PTZ) camera and camera jitter. Mandal et al. [29] presented a completely end-to-end spatio-temporal network 3DCD for simultaneous background estimation and other tasks, e.g. saliency detection. Multiple cues are employed in MU-Net [30] for background subtraction, including tensor-based motion estimation and GMM based background subtraction. Reference [31] is the early version of this paper, where both appearance and motion features are combined to identify general moving targets without considering airport prior information. The above methods all recognize to some extent that deep learning-based methods have difficulties in detecting moving objects, and they are more or less effective in solving this problem. However, they do not clearly define MSD, nor do they consider using prior information in specific scenarios to solve the MSD problem. This paper clearly defines MSD for the first time and proposes a method based on airport-specific prior information to solve the MSD problem in airport background subtraction.

C. Motion Segmentation and Video Object Segmentation

Motion segmentation aims to segment each frame into regions with different motion parameters [51]. Motion segmentation generally relies on dense optical flow [52]. However, Yue et al. [53] proposed a method that combines deep learning and geometric reasoning, eliminating the need for optical flow. Siam et al. [54] introduced a method that integrates motion and appearance cues for joint motion segmentation and object detection. This technique was further improved in [55] by explicitly modeling vehicle motion. Mariotti and Eising [56] considered four types of geometric constraints for motion segmentation using fisheye cameras.

There is a large number of video object segmentation papers. A big family of automatic video object segmentation is built upon two-stream networks [37]. Some automatic video object segmentation algorithms not only separated the foreground from background, but also discriminated different

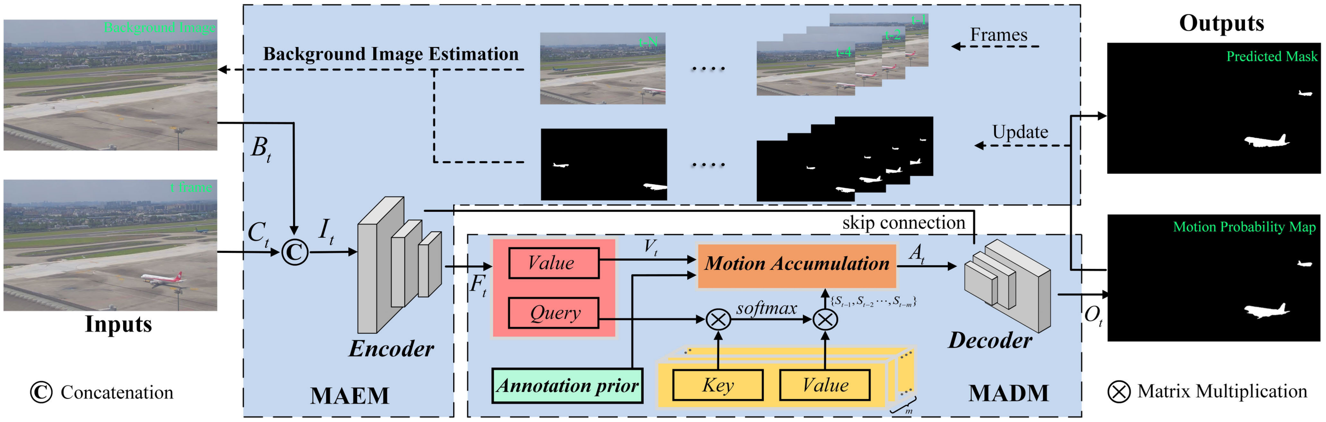


Fig. 4. An overview of MBSNet. Firstly the current frame C_t and the background frame B_t from the Background Image Estimation step are compared in the Motion Augmentation Encoder Module (MAEM) to augment the motion component. Secondly, in the Motion Accumulation Decoder Module (MADM), the encoded feature F_t are accumulated with stored features from previous frames by the Motion Accumulation step to further enhance the motion component. Finally, the decoded motion probability map O_t is thresholded to generate the output mask.

object instances [38]. There were also methods learn to perform automatic video object segmentation from unlabeled or weakly labeled data [39]. Semi-automatic video object segmentation algorithms involve limited human inspection. Oh et al. [40] utilized the given mask of the first frame as a template and match the pixel-level feature embeddings in new frames. The ranking attention mechanism was leveraged in RANet [41] to construct pixel-wise similarity maps.

III. Proposed Method

Since the original deep learning network tends to learn appearance features, the stationary aircraft with a similar appearance to the moving aircraft may be misclassified as foreground. An obvious line of thought is that if more motion features can be utilized, the classification accuracy will be improved. This problem is easy to solve for traditional methods by using or developing motion features for modeling. However, because of the interpretability of deep learning, we cannot determine which features learned by the network belong to motion features and which belong to appearance features, and hence it is infeasible to conduct motion modeling directly as the traditional background subtraction.

Our idea is to indirectly enhance the motion component in the learned features to force the network to pay more attention to motion for MSD. MBSNet is illustrated in Fig. 4, which has two modules, Motion Augmentation Encoder Module (MAEM) and Motion Accumulation Decoder Module (MADM). A clean background image is estimated in MAEM by the weighted summation of previous frames with the predicted motion probability as the weights. Next, the background image and the current frame are used as the input of encoder operation for feature extraction. Given a specific task and the inputs, the deep learning network will automatically learn the required features from inputs. For background subtraction, if the inputs are images with and without moving objects as in Fig. 4, the network tends to learn various features of the moving object, which must include motion information. In other words, the motion component in the encoded features is indirectly augmented by MAEM.

The motion component will be further enhanced by MADM. For the appearance component in the encoded features, it changes little between adjacent frames for both foreground and background, unless the object deformation is large. For the motion component reflecting the position attributes, it is different for foreground and background. The position change of background object in video surveillance generally is small and hence the background motion component between adjacent frames should be similar. On the contrary, the foreground position along with the corresponding motion features between adjacent frames are always changing because the foreground object is always moving. Therefore, if the similarity of encoded features between adjacent frames is measured, the feature values with small similarity degrees are likely to belong to the foreground motion component. Based on this fact, the MADM accumulates the encoded features of previous frames to the current frame based on feature dissimilarity measurement. Because the accumulation principle is negative accumulation when similar and positive accumulation when dissimilar, the result is that the foreground motion component is enhanced. After decoder operation, the accumulated features will be retrieved to the original resolution, generating a pixel-wise map of motion probability, and moving objects can be segmented by thresholding the probability map.

A. Motion Augmentation Encoder Module

1) Background Image Estimation: An intuitive idea to augment the motion component in learned features is to compare images with and without moving objects. A background image estimation step is necessary to achieve this goal. The commonly used method to estimate a background image is the temporal mean background model [42],

$$\hat{p}_{i,j,r} = \frac{1}{N} \sum_{t=1}^{N} p_{i,j,t}, \tag{1}$$

where r is the current time instant, t is the frame index, N is the number of images evolved in computation, p represents the pixel value at spatial location (i, j), and $\hat{p}$ is the estimated

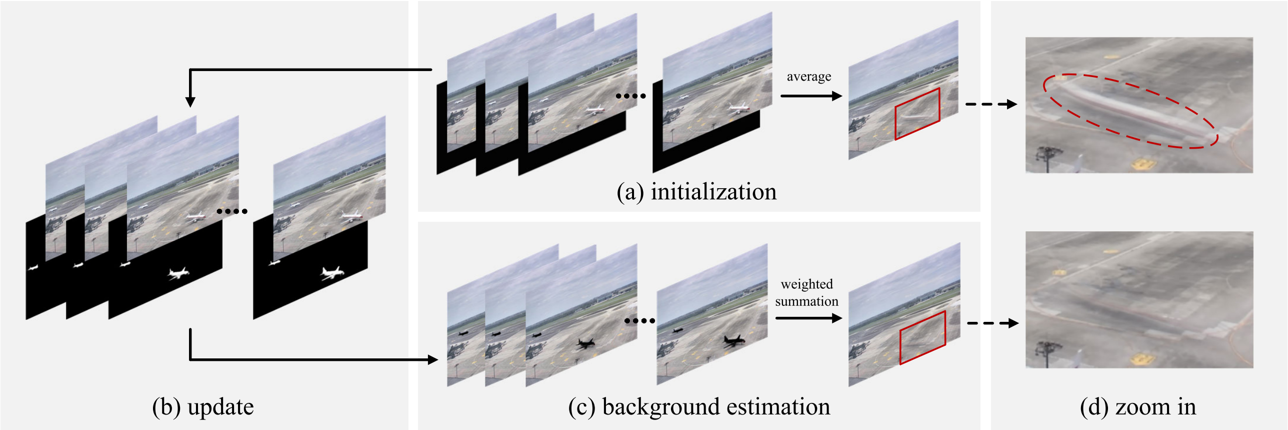


Fig. 5. The scheme of Background Image Estimation. The background image is set up by averaging the initial frames with empty motion probability maps. Then, the sample set is updated in (b) with incoming frames and their associated motion probability maps. Lastly, a new background image is calculated through weighted summation in (c).

pixel value. This method is very simple and brings some problems. Specifically, sometimes there will be ghosts or shadows in the estimated background image, resulting in hollows in the final segmentation results. The reason for this phenomenon is the presence of foreground pixels in training samples. If the samples used for modeling include both foreground and background pixels, the simple mean operation can not remove the foreground pixels, and hence it is difficult to generate a clean background image.

Hence, we present a new background image estimation solution to address the above problem, as shown in Fig. 5. We consider using the predicted pixel-wise motion probability map by the network as the weights of the weighting summation of previous frames. If a pixel has a small motion probability value, it is more likely to be a background pixel and it should contribute more to the background image estimation, and vice versa. The scheme of the background estimation includes three steps, corresponding to (a-c) in Fig. 5. Firstly, the background image is initialized by the average summation of several previous frames with empty motion probability maps. At this time, the model is similar to the temporal mean background model. Secondly, the sample set is updated with the First-In-First-Out (FIFO) rule, that is, to delete the frame furthest from the current time, and insert the newly-arriving frame and its motion probability map. Finally, new background image is estimated by weighted summation as follows,

$$\hat{p}_{i,j,r} = \frac{1}{\varepsilon + \sum_{t=1}^{N} w_{i,j,t}} \sum_{t=1}^{N} w_{i,j,t} \, p_{i,j,t}, \tag{2}$$

where r is the current time instant, t is the frame index, N is the number of previous frames involved in computation, $p_{i,j,t}$ is the pixel value of frame t at the spatial location (i, j), and $\hat{p}_{i,j,r}$ is the estimated background pixel value. The weight $w_{i,j,t}$ is computed as $w_{i,j,t} = 1 - o_{i,j,t}$, where $o_{i,j,t}$ just is the predicted motion probability of pixel $p_{i,j,t}$. ε is a small constant scalar to avoid zero denominator. It can be seen that the larger the motion probability $o_{i,j,t}$, the smaller the contribution of $p_{i,j,t}$ to the background image estimation. In this way, the negative impact of moving object pixels in the training frames can be reduced. As shown in Fig. 5(d), a better background image can be obtained by our method compared with the temporal mean. The generation of the motion probability map will be described in Motion Accumulation Decoder Module.

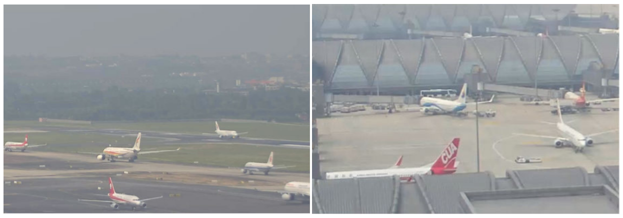


Fig. 6. Illustration of the airport flight area and apron during crowded hours. It can be seen that even at crowded moments, the targets in the airport are relatively sparse, which ensures that a clean background image can always be obtained based on long-term observation.

The premise of our background image estimation method is that there are not too many moving objects in the scene. The airport scenario fits this requirement. Two examples of crowded airport are shown in Fig. 6. It can be seen that the number of aircraft in the airport flight zone and apron area is limited even at crowded moments. This is because aircraft are massive objects and must be separated by sufficient distance to ensure safety. Therefore, based on long-term observation, we can always get a clean background image.

2) Encoder Operation: In the framework of deep learning, the network will automatically learn suitable features from the input to accomplish the established goal of the network. As shown in Fig. 4, the estimated background image and the current frame are used as the input of the proposed network. In this case, what the network learns must be various features of the moving object, such as appearance features, motion features, and so on. In this way, the motion information will be indirectly enhanced in the encoded features by MBSNet. However, due to the interpretability of deep learning, currently, we can not judge which features represent motion information, appearance, or other information. The interpretability problem is still an open question in deep learning.

We use Resnet50 [43] as a backbone in our encoder module. The encoder takes a pair of RGB images (the current frame and estimated background) as input ($I \in \mathbb{R}^{6\times H\times W}$) in a manner of concatenation, where H and W are the height and width of an input image and the 6 represents the number of channels. Layers 1-4 of Resnet50 are utilized to extract multi-scale features and sent to different stages of the decoder, and there are three stages in the decoder for upsampling correspondingly. We take the output of layer-4 of Resnet50 as encoding feature map ($F \in \mathbb{R}^{1024\times H/16\times W/16}$). The feature map will be matched with past frame features in the Motion Accumulation Module and then sent to the decoder.

B. Motion Accumulation Decoder Module

1) Motion Accumulation: A straightforward idea is whether the multi-frame features by the MAEM module can be added up to further highlight the motion component? The premise of such accumulation is motion consistency, otherwise the accumulated features are chaotic. The motion consistency means that the motion pattern remains unchanged as much as possible. The aircraft has good motion consistency. Because the aircraft is a huge object, it cannot move arbitrarily, but must travel at a constant speed along a straight runway or taxiway. At this time, the motion pattern of the aircraft, including direction and speed, remains unchanged. However, when the aircraft turns, the motion consistency is destroyed. At the turning stage, the direction of the aircraft, and even the appearance pattern, changes drastically.

We find that the motion pattern of the aircraft on the ground is simple and predictable, that is, straight ahead with occasional turns. This allows us to manually label the turning areas in the scene and then perform motion accumulation only in non-turning areas. Before the algorithm runs, it is easy to distinguish where the turning area is by observing the structure of the airport ground. Of course, other methods can also be used to mark the turning area. For example, when calculating the motion direction vector based on the object tracking trajectory, the area where the motion direction vector significantly changes is the turning area. Because the algorithm is designed for practical application in airport surveillance systems, such as the conflict warning system, it should be simple and efficient, so we finally use the manual annotation solution. Manual annotation is shown in Fig. 7, where the turning area is represented by simple lines. Before the current frame is segmented, it is not known whether the target to be segmented falls in the turning area or not, so it is obtained by judging the past frames to be accumulated. If any pixel of a target in the past frame to be accumulated falls in the turning area, i.e. overlapping with the annotation line, this target will not be accumulated.

In the non-turning area, the position of the moving aircraft is always changing between adjacent frames. On the contrary, in video surveillance with a fixed camera, the background object usually does not move or has only weak motion. This is reflected in the encoded features, that is, the dissimilarity between the encoded features of adjacent frames generally is where the foreground motion features are. We can make use of this phenomenon to further enhance the motion component of the encoded features after MAEM. To achieve this goal, we develop the Motion Accumulation in MADM by referring to the attention mechanism [44], as shown in Fig. 8. The accumulation principle is *the negative accumulation when similar and positive accumulation when dissimilar*. The Motion Accumulation includes two steps, dissimilarity measurement and feature accumulation. The dissimilarity of encoded features between adjacent frames is measured in the first step, which is then used to accumulate past frame features to the current frame in the second step.

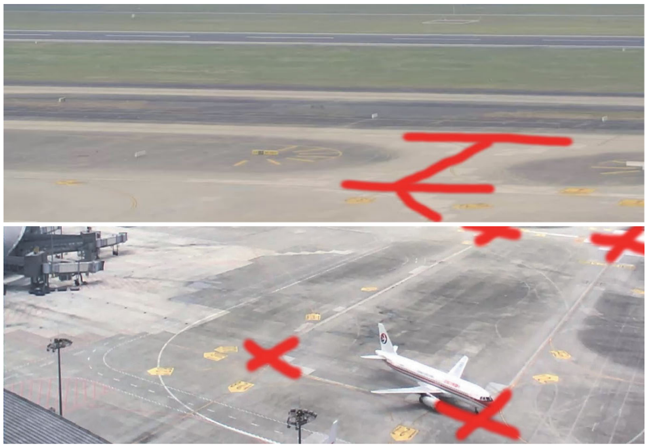
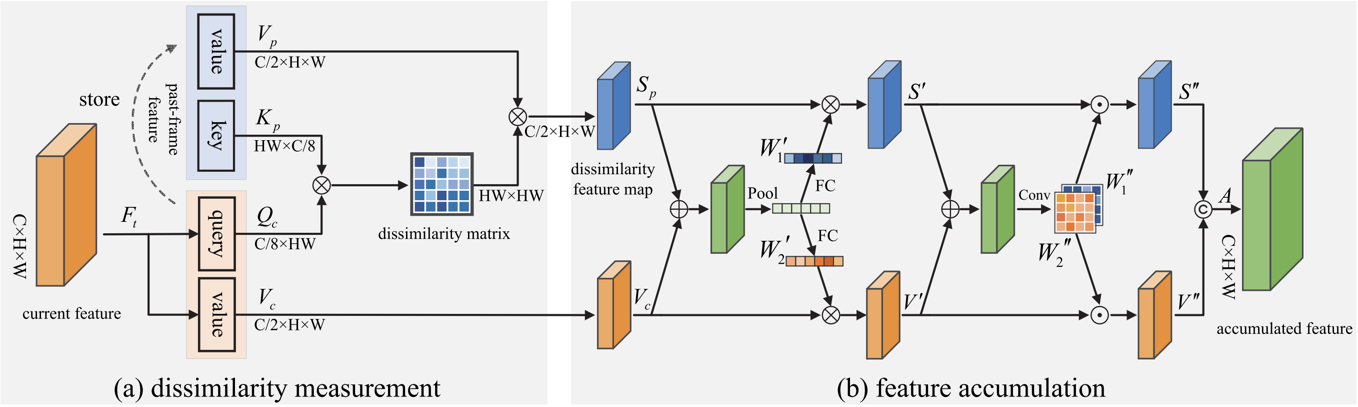


Fig. 7. Illustration of manually annotated turning areas in the flight zone and apron. When the aircraft passes through a turning area, the motion pattern, especially the direction, will change dramatically, while there is good motion consistency in the non-turning area. Therefore, only performing motion accumulation in the non-turning area can ensure the effectiveness of the accumulated features.

Motion Accumulation of two consecutive frames is shown in Fig. 8. In the first step of dissimilarity measurement, there are three different 3 × 3 convolution layers that are exploited for query, key, and value [44] feature extraction after the layer-4 of Resnet50 [43]. The current query represents the feature map relationship between the current frame and the previous frame. The key is calculated and applied for future feature matching. Value stores detailed information that we consider as motion and other features to segment moving objects. Each frame produces three features so that two consecutive frames would have six features. However, only four of these features are used when matching, namely Q_c, V_c, K_p, and V_p, which represent the query and value feature of the current frame and the key and value feature of the previous frame, respectively. Then the query of the current frame will be multiplied by the previous key to compute similarity in an embedding space. It will be normalized by the softmax function two times to obtain the dissimilarity matrix. Next, the dissimilarity matrix is weighted to the value of the previous frame to obtain a dissimilarity feature map that covers a potential representation with the previous frame. The dissimilarity feature map can be calculated as follows,

$$S_p = softmax(1/softmax(\frac{K_p^T \cdot Q_c}{\sqrt{D}})) \cdot V_p, \tag{3}$$

where Q, K and V represent query, key and value of features ($Q \in \mathbb{R}^{C/8\times H\times W}$, $K \in \mathbb{R}^{C/8\times H\times W}$, $V \in \mathbb{R}^{C/2\times H\times W}$),



(a) dissimilarity measurement

(b) feature accumulation

Fig. 8. The scheme of Motion Accumulation. The dissimilarity between frames is first calculated, which is then used as a weight for accumulation. The feature maps are represented in tensor form, with H, W and C denoting height, width, and channels, respectively. The blue, orange and green entities indicate cues of the previous frame, the current frame, and the accumulated result, respectively.

$D = H \times W$ is the dimension of the dissimilarity matrix, and the subscript c and p identify the current and previous frame. The dissimilarity measurement shown in Eq. 3 is the core idea of MADM, because the motion component can be further enhanced by the dissimilarity feature map.

The dissimilarity feature map and the current frame value are the input of the second step feature accumulation. The goal of this step is to enhance the motion component in the features and weaken other components. To achieve this, we use the dissimilarity map as the weights of accumulation by referencing the channel attention [45] and spatial attention [46]. The basic idea of channel-wise attention is to control the flow of information by the squeezed feature at the channel dimension. As shown in Fig. 8, we first fuse two kinds of features by an element-wise addition. Then, global average pooling is used to perform spatial squeeze,

$$S'_{squeezed} = \mathcal{F}_{avgpool}(S_p + V_c). \tag{4}$$

Next the squeezed feature is further feed into two fully connected layers to generate the adaptive weight vectors Z'_1, Z'_2, which are split from Z',

$$Z' = \mathcal{F}_{FC}(S'_{squeezed}). \tag{5}$$

The channel-wise softmax function is used to generate adaptive weights W'_1, W'_2 corresponding to different level features Z'_1, Z'_2 as,

$$W'_i = \frac{exp(Z'_i)}{\sum_{j=1}^{2} exp(Z'_j)}, \quad i \in \{1, 2\}, \tag{6}$$

where W'_i represents relative importance of features Z'_i at channel C. The weighted feature maps for two information flows can be formulated as,

$$S' = S_p \odot W'_1, \quad V' = V_c \odot W'_2. \tag{7}$$

After we obtain the channel-selected features S', V', they will be fed into the next attention operation for spatial enhancement. Similar to channel-wise attention, we firstly fuse two features by element-wise addition. And then, we employ 1×1 convolutional filters to perform the channel squeeze to reinforce the features on the spatial dimension,

$$Z'' = \mathcal{F}_{conv}(S' + V'). \tag{8}$$

Next, the channels are compressed into two layers, where the feature map of each channel corresponds to the spatial weights for each level feature (i.e. S', V'). After that, a softmax function is used to rescale activations so that we can obtain pixel-wise adaptive weights W'' in spatial dimension,

$$W'' = \mathcal{F}_{softmax}(Z''). \tag{9}$$

Then the channel-selected features (S', V') are weighted by $W''_1, W''_2 \in W''$ to get spatial enhanced features,

$$S'' = S' \odot W''_1, \quad V'' = V' \odot W''_2. \tag{10}$$

At last, the accumulated features can be obtained by the concatenation of the two enhanced features as $A = \mathcal{F}_{concat}(S'', V'')$. The key and value of the current frame feature will be stored in memory for the next round of processing.

The principle of the above operations is that positive accumulation when dissimilar and negative accumulation when similar. Considering that the dissimilarity between adjacent encoded features is where the foreground motion features lie, the goal of further enhancement of the motion component can be achieved by the Motion Accumulation.

Please note that Fig. 8 only illustrates two-frame accumulation. Better results may be obtained by multi-frame accumulation. For m past frames, we compute the dissimilarity map of each past frame with the current frame, and the average of multiple dissimilarity maps is then used as the input of the feature accumulation step. Generally speaking, it would be better to use multiple frames, but this is based on the premise of motion consistency. If the motion state, e.g. the velocity of the aircraft changes, multi-frame accumulation does not necessarily achieve the result we want or may even cause side effects. Therefore, m needs to be carefully set.

2) Decoder Operation: The task of decoder operation is to generate a pixel-wise motion probability map based on extracted features. Since the motion component in the encoded

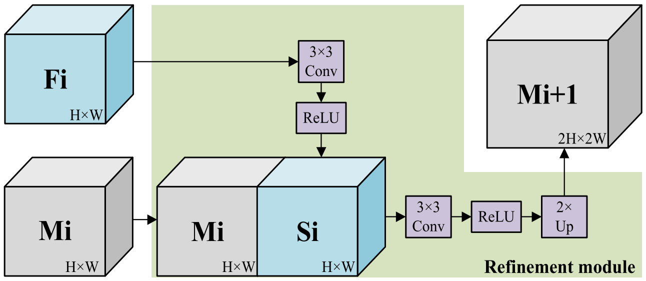


Fig. 9. Illustration of the decoder network. The refinement module aims to enhance the mask encoding Mi using features Fi. It begins by reducing the channels in Fi through a 3×3 convolutional layer followed by a ReLU activation, generating skip features Si. Next, Fi and Si are concatenated along the depth dimension and convolved along the spatial dimensions. Finally, the output is upsampled using bilinear interpolation by a factor of 2, resulting in Mi+1.

feature vector is firstly augmented by MAEM, and further enhanced in MADM, the generated probability map is able to more clearly show that where the moving object is located. As a result, the MSD problem in background subtraction can be better addressed by MBSNet.

To efficiently explore features in different scales, we use the refinement module [47] as the building block of our decoder, which is composed of several convolution layers in the form of residuals and interpolation to expand the resolution of the feature map. As shown in Fig. 9, each stage of the decoder takes the output of the previous step and the feature map from different encoder layers as inputs in the way of skip connections. The last feature map is fed into a 3 × 3 convolution without ReLu function, and the output is a probability map ($O \in \mathbb{R}^{H/4 \times W/4}$) gained by a softmax layer. This map represents an estimation of the background and foreground motion probability. Linear interpolation is used to retrieve the probability map to the original resolution, and then the moving object can be detected by thresholding the probability map.

C. Inference

Considering the imbalance in the number of foreground and background pixels, we use joint losses, Dice loss and binary Cross-Entropy loss, for pixel-wise classification. The joint loss equation is as follows,

$$\mathcal{L}_{joint} = \lambda \mathcal{L}_{dice} + (1-\lambda)\mathcal{L}_{bce}, \tag{11}$$

where the λ is a weight scalar which we set 0.5 in our experiment. The Dice loss $\mathcal{L}_{dice}$ is used for extracting the object regions, while the Binary Cross-Entropy loss $\mathcal{L}_{bce}$ encourages the network to get the contour of the objects. Their formula are as follows,

$$\mathcal{L}_{dice} = 1 - \frac{2\sum_i q_i y_i + \varepsilon}{\sum_i q_i + \sum_i y_i + \varepsilon}, \tag{12}$$

$$\mathcal{L}_{bce} = -\sum_i y_i log q_i - \sum_i (1-y_i) log(1-q_i), \tag{13}$$

where q is the predicted probability of pixels and y is the groundtruth label, ε is a constant number.

The proposed network is initialized by a pre-trained model on the ImageNet, and an Adam optimizer is used for optimization with an initial learning rate of 5e-4 and a learning rate reduction strategy (divided by 5 at epoch 8 and 12). We train a total of 15 epochs with a batch size of 4, and the threshold for the motion probability map is set to 0.5.

IV. Experiments

A. Experimental Settings

We evaluated the performance of MBSNet on the AGVS [5] dataset, which is currently the only dataset for airport ground surveillance. AGVS contains 25 long video clips (S1 to S25) captured in the airport, amounting to about 100000 frames from 1280×720 to 1920×1080 resolution with accurate pixel-wise groundtruth. There are multiple challenges in AGVS, e.g. haze, camouflage, small target, special shapes, different weather conditions, shadow and illumination changes, etc. AGVS is a perfect MSD-type dataset. AGVS has many stationary and moving aircraft with similar appearances, so that there are plentiful instances of Case1. Furthermore, the aircraft in AGVS is often intermittent, resulting in many instances of Case2.

We chose Recall (Re), Specificity ($Spec$), False Positive Rate (FPR), False Negative Rate (FPR), Precision (Pr), F-Measure (FM) and Percentage of Wrong Classification (PWC) as performance evaluation metrics,

$$Re = \frac{TP}{TP+FN}, \quad Spec = \frac{TN}{FP+TN},$$
$$FPR = \frac{FP}{FP+TN},$$
$$FNR = \frac{FN}{TP+FN}, \quad Pr = \frac{TP}{TP+FP},$$
$$FM = \frac{2 \times Re \times Pr}{Re+Pr},$$
$$PWC = \frac{100 \times (FN+FP)}{TP+FN+FP+TN},$$

where TP, FP, TN and FN are the numbers of true positives, false positives, true negatives and false negatives, respectively. Among these metrics, the FM is particularly important since it measures the overall performance of the algorithm. We also used Frames Per Second (FPS) to assess the algorithm's operational efficiency.

Six traditional methods (GMM [10], ViBe [14], PAWCS [16], SuBSENSE [15], SWCD [19] and WisenetMD [20]) and nine deep learning methods (FgSegNet [4], Cascade CNN [24], BSUV-net [27], RGMP [40], SegFlow [49], STA-Net [50], 3DCD [29], MU-Net [30] and RT-SBS [33]) are used for comparison. The complete names and abbreviations of all comparison algorithms are provided in Tab. I for reader convenience. For the first four algorithms, we utilized public codes and default parameter settings from the BGSLibrary [57]. For the remaining algorithms, we employed the public code and recommended experimental approaches suggested by the respective authors. The unsupervised method was executed on a PC equipped with an Intel i5-11600KF CPU and 32-GB

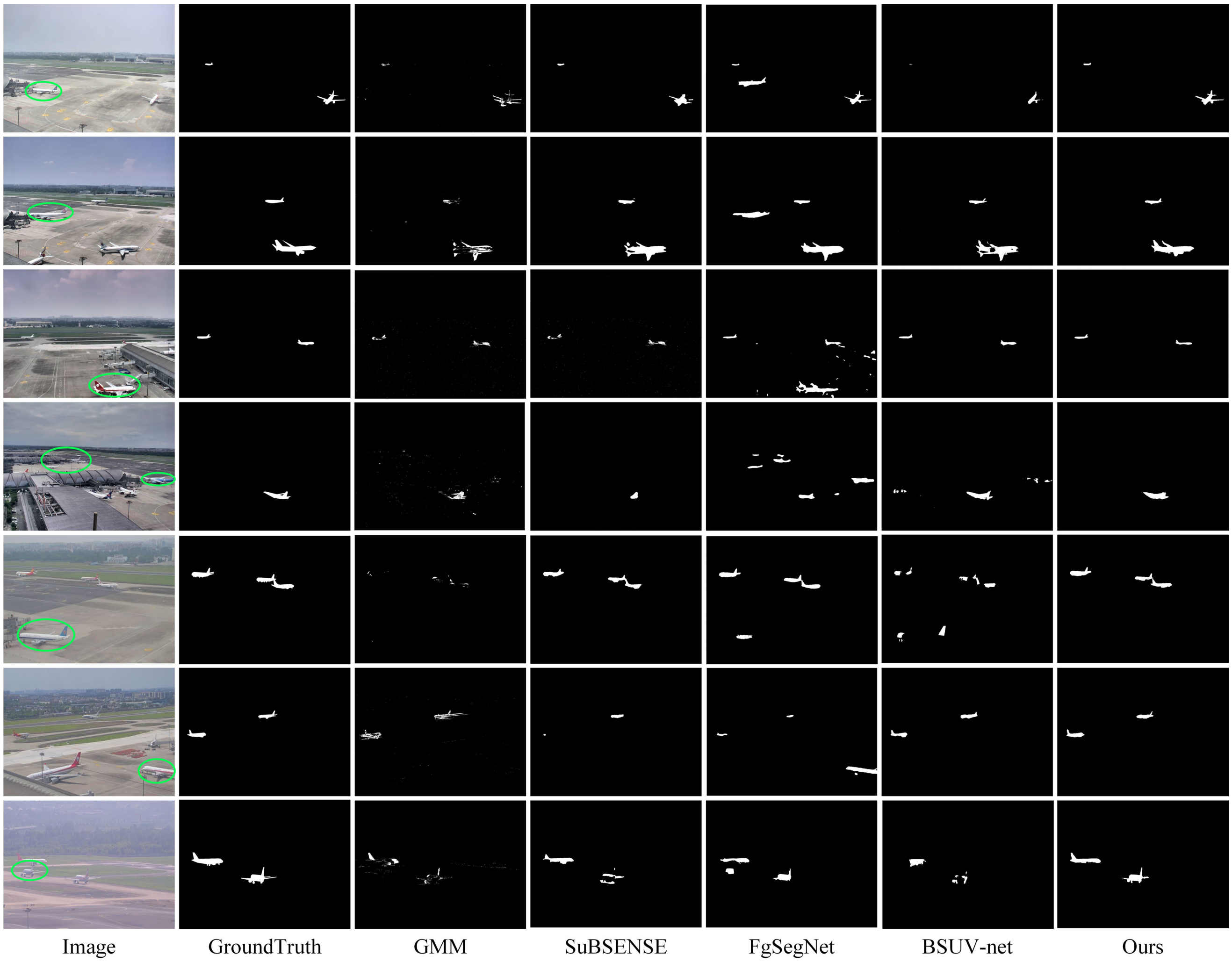


Fig. 10. In the first column, from top to bottom, they are instances of Case1 (highlighted with green ellipses) from S2, S6, S10, S14, S16, S21 and S22 in AGVS, respectively. These instances can be distinguished by comparing the original images to the ground truth of the moving target shown in the second column. While traditional methods like GMM [10] and SuBSENSE [15] are useful for Case1, they often yield low accuracy. With the exception of MBSNet, deep learning approaches like FgSegNet [4] and BSUV-net [27] is prone to misclassifying stationary aircraft.

TABLE I
THE FULL NAMES AND ABBREVIATIONS OF ALL COMPARISON ALGORITHMS

Full Names	Abbreviations
Gaussian Mixture Model [10]	GMM
Visual Background extractor [14]	ViBe
Pixel-based Adaptive Word Consensus Segmenter [16]	PAWCS
Self-Balanced SENsitivity SEgmenter [15]	SuBSENSE
Slide Window for Change Detection [19]	SWCD
Wisenet for Motion Detection [20]	WisenetMD
Foreground Segmentation Network [4]	FgSegNet
Cascade Convolutional Neural Network [24]	Cascade CNN
Background Subtraction for Unseen Videos [27]	BSUV-net
Reference-Guided Mask Propagation [40]	RGMP
Segmentation Flow [49]	SegFlow
Spatio-Temporal Alignment Network [50]	STA-Net
3D-CNN based Change Detection Network [29]	3DCD
Motion U-Net [30]	MU-Net
Real-time Semantic Background Subtraction [33]	RT-SBS

RAM, with the addition of a single NVIDIA GeForce GTX 1080Ti GPU for the deep learning-based method.

Our proposed method was assessed using a two-fold cross-validation strategy on the AGVS dataset. This involved dividing the dataset into two equally sized subsets, G1 and G2, which were alternated between training and testing sets for evaluation. The division principle for AGVS was to ensure each group contained videos from different viewing angles. Following this principle, videos S1-S6, S10, S11, S17, S18, and S21 were categorized into G1, while videos S7-S9, S12-S16, S19, S20, and S22 were placed in G2. Each subset now comprised 11 videos. For a fair comparison with traditional methods, PTZ (S23 to S25) were not included, since the traditional methods are invalid for such videos.

B. Visual Analysis

Fig. 10 displays the comparative experimental outcomes for Case1 of the MSD problem. In the second column, the ground truth of the aircraft in motion at the present time is illustrated. Upon comparing the original images in the first column with the ground truth in the second column, it's evident that numerous aircraft remain stationary (highlighted with green ellipses), illustrating Case1 of the MSD problem.

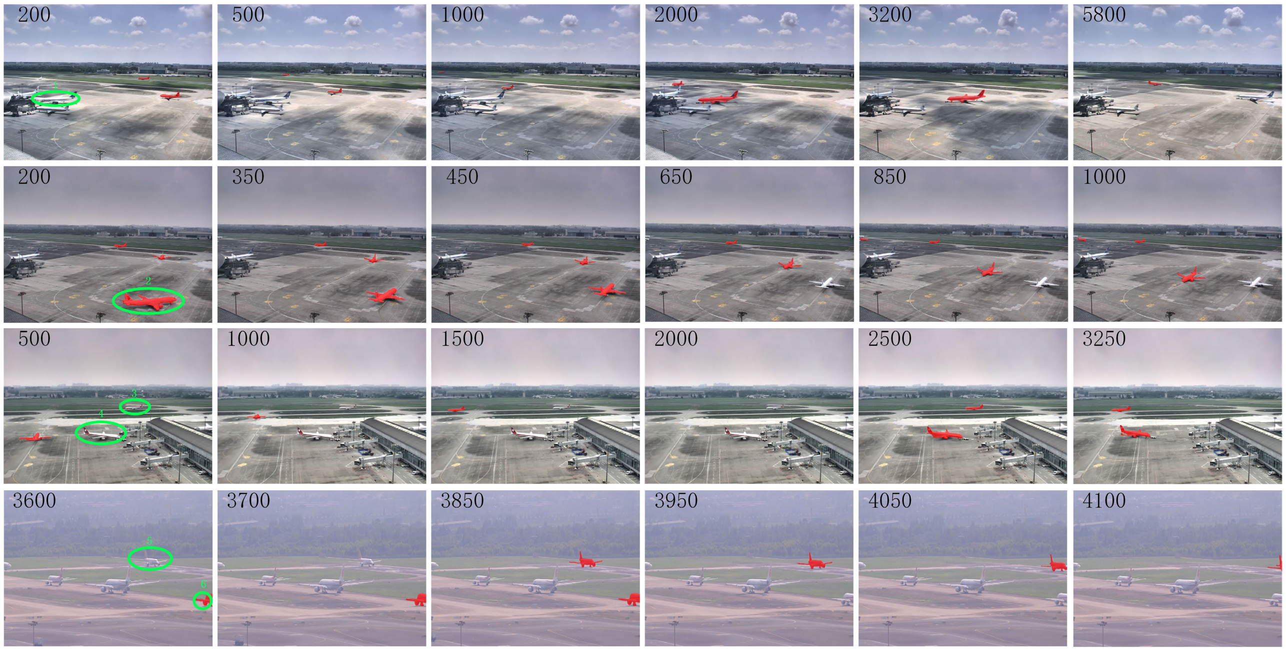


Fig. 11. In the first column, from top to bottom, they are instances of Case2 (highlighted with green ellipses) from S4, S8, S11 and S22 in AGVS, respectively. The detection results of MBSNet are indicated by red masks. Aircraft 1 through 6 were in motion during specific frames: frames 2000 to 3200, 200 to 450, 2500 to 3250, 2500 to 3250, 3850 to 4050, and 3600 to 3850, respectively. At other times, they remained stationary. In each of these instances, MBSNet accurately determined the motion status of the target.

There are four comparison algorithms in Fig. 10, GMM [10], SuBSENSE [15], FgSegNet [4] and BSUV-net [27]. The first two are traditional methods, and the last two are deep learning methods. FgSegNet is almost completely invalid for Case1 of MSD, where many stationary planes were detected as foreground, especially in S6 and S14. BSUV-net [27] which was presented for the unseen video problem performs better than FgSegNet, but there were still failures for Case1, e.g. in S14 and S16. On the contrary, the traditional methods can distinguish between motion and stillness well, completely without misidentifying stationary aircraft as foreground. This is because of the natural advantage of traditional methods, which can directly model motion. However, the traditional methods have great problems in segmentation accuracy, such as incomplete segmentation, unsmooth segmentation, hole and fracture (S2 and S16), false detection under bad weather condition (S22), segmentation noise (S14), ghost (S6) and so on. The inaccuracy phenomenon of traditional methods can be seen in almost all examples. The segmentation results by our method are shown in the last column of Fig. 10, where all moving aircraft were correctly detected without misclassification of any stationary aircraft.

Fig. 11 illustrates the results of our method in Case2 of the MSD problem. In Case2, the moving target exhibits intermittency, meaning it alternates between motion and stillness. Figure 11 comprises four sequences (S4, S8, S11, and S22), with each sequence demonstrating Case2 through consecutive frames at specific intervals. In the first column of Figure 11, intermittent targets are highlighted with green ellipses. Additionally, the foreground masks detected by our method are directly overlaid onto the original frames to provide a visual representation. We can see that our method is sensitive to the motion state transition of the same object. For example, the stationary aircraft in frames 200, 500, and 1000 is detected as foreground when it is moving in frames 2000 and 3200 for S4, and the moving aircraft in frames 200, 350, and 450 is no longer detected in frame 650, 850, and 1000 after it stops for S8. There is also motion state transition from frame 500 to frame 3250 in S11 and from frame 3600 to frame 4100 in S22, which are also correctly detected by MBSNet. Case2 is more challenging than Case1 because the moving and stationary objects have exactly the same appearance. Case2 is a great challenge for deep learning based methods, and almost all existing methods can not solve this problem well.

Our method also has some failure cases. In sequence S4 of Fig. 11, aircraft 1 is stationary at frame 1000, and at frame 2000, the aircraft is in motion and can be correctly detected by our method. However, the aircraft actually started moving from frame 1450, and was first detected at about frame 1500. In other words, our method does not respond promptly to aircraft that change from stationary to moving. Similar phenomena can be seen for other aircraft, such as aircraft 3 and 4 in sequence S11. This may have two reasons. First, the motion accumulation operation takes some time to take effect. Second, the aircraft targets in the AGVS dataset, especially those taxiing on the apron, move slowly, which poses a great challenge to distinguishing between motion and stillness.

C. Quantitative Analysis

Quantitative results of all comparison algorithms on the whole AGVS dataset are shown in Tab. II, where the FM is used to evaluate the comprehensive performance. We can see that except BSUV-net, STA-Net, and our method, the FM

TABLE II

QUANTITATIVE COMPARISON OF 16 ALGORITHMS ON THE AGVS DATASET. THE FIRST SIX ALGORITHMS ARE TRADITIONAL METHODS, AND OTHERS ARE DEEP LEARNING BASED METHODS. THE PROPOSED METHOD HAS THE BEST $F-Measure$ VALUE

Method	supervised	$Re\uparrow$	$Spec\uparrow$	$FPR\downarrow$	$FNR\downarrow$	$PWC\downarrow$	$Pr\uparrow$	$F-Measure\uparrow$	FPS
GMM [10]		0.5321	0.9957	0.0043	0.4679	1.3194	0.5107	0.5212	28.5
ViBe [14]		0.4721	0.9951	0.0049	0.5279	1.4023	0.4629	0.4675	22.5
PAWCS [16]		0.5332	0.9973	0.0027	0.4668	1.2344	0.6086	0.5684	2.3
SuBSENSE [15]		0.6229	0.9969	0.0031	0.3771	1.2084	0.5971	0.6097	6.7
SWCD [19]		0.3420	0.9987	0.0013	0.6580	0.6899	0.6955	0.4585	3.6
WisenetMD [20]		0.4667	0.9871	0.0129	0.5333	1.7384	0.2371	0.3144	6.2
Cascade CNN [24]	✓	0.4582	0.9936	0.0064	0.5418	1.7215	0.4220	0.4394	13.9
RGMP [40]	✓	0.5540	0.9929	0.0071	0.4460	1.0918	0.3700	0.4437	8.5
FgSegNet [4]	✓	0.4671	0.9963	0.0037	0.5329	1.5878	0.4373	0.4517	16.4
SegFlow [49]	✓	0.6455	0.9955	0.0045	0.3545	1.0247	0.3931	0.4886	0.12
3DCD [29]	✓	0.4503	0.9970	0.0030	0.5497	0.7220	0.5410	0.4915	4.0
MU-Net [30]	✓	0.3688	**0.9999**	**0.0001**	0.6312	0.5463	**0.9814**	0.5361	11.2
RT-SBS [33]	✓	0.4419	0.9991	0.0009	0.5581	**0.4471**	0.7629	0.5596	1.9
STA-Net [50]	✓	0.6602	0.9932	0.0068	0.3398	0.8617	0.6014	0.6294	8.1
BSUV-net [27]	✓	0.7516	0.9992	0.0008	0.2484	0.9203	0.7374	0.7444	2.1
MBSNet	✓	**0.8228**	0.9989	0.0011	**0.1772**	0.7425	0.7859	**0.7974**	12.2

TABLE III

COMPARISON OF BSUV-NET AND MBSNET WITH THE SAME OR DIFFERENT BACKBONE NETWORKS IN TERMS OF $F-Measure$. OUR METHOD HAS THE BEST $F-Measure$ VALUE IN BOTH CASES

backbone network	BSUV-net	MBSNet
VGG	0.7296	0.7346
ResNet-50	0.7511	0.7974

TABLE IV

ABLATION EXPERIMENTS BY MASKING SOME STEPS IN THE MBSNET AND USING A DIFFERENT VALUE OF m

Network Variant	Re	Pr	FM
MBSNet w/o BIE	0.5315	0.5097	0.5204
MBSNet w/o MA	0.6966	0.6552	0.6752
MBSNet w/o Acc.	0.7450	0.7216	0.7331
MBSNet (m=1)	0.7738	0.7561	0.7698
MBSNet (m=3)	0.7903	0.7695	0.7797
MBSNet (m=10)	0.8022	0.7701	0.7858
MBSNet (m=5)	**0.8228**	**0.7859**	**0.7974**

of all supervised methods is inferior to that of traditional methods. The reason for this phenomenon is that there is a large number of MSD instances in AGVS, which brings great difficulties to the deep learning methods. On other datasets, such as CDnet2014 [48], the performance of supervised methods is much better. MBSNet has the best performance in terms of FM, which shows that it is effective for the MSD problem. In addition, MBSNet also achieves competitive time efficiency in terms of FPS. The FM of BSUV-net is good, but the FPS indicates that it has a large computational load.

Although Tab. II shows the FPS of all algorithms, it is unfair to directly compare the FPS of unsupervised and supervised algorithms because they use different hardware. In addition, we note that other code versions of some algorithms in Tab. II are faster. For example, the GMM algorithm in OpenCV can achieve a speed of more than 100 frames per second. This is because there are many optimizations in OpenCV library. Unlike OpenCV, the algorithm implementation in BGSLibrary is very faithful to the original paper, which is helpful for fair comparison.

The backbone network in BSUV-net is VGG while it is ResNet-50 in MBSNet. A new experiment on AGVS is conducted to compare BSUV-net and MBSNet with the same backbone network, as shown in Tab. III. Tab. III indicates that no matter which backbone network is used, the overall performance of MBSNet is better than that of BSUV-net.

The proposed method also has limitations. First of all, this method is specifically targeted at the airport. Although it can also be used in other scenarios, it may not necessarily bring performance improvement. Secondly, this algorithm has an initialization step before running, which requires the operator to have a certain understanding of the airport structure. For the first limitation, we can design MSD-type dataset for other scenarios, and then design MSD algorithms for them based on similar ideas as the presented method.

D. Ablation Study

We carried out ablation experiments on the AGVS dataset to assess the effectiveness of each component within the proposed MBSNet. Our method comprises two main modules: the Motion Augmentation Encoder Module (MAEM) and the Motion Accumulation Decoder Module (MADM). However, it's not feasible to test one module by completely disabling the other, as doing so would render the encoder-decoder framework incomplete. To evaluate MADM independently, we disabled the Background Image Estimation step (BIE) in MAEM and solely retained the original encoder operation, labeled as 'MBSNet w/o BIE' in Tab. IV. Similarly, to test MAEM in isolation, we disabled the Motion Accumulation step (MA) in MADM and solely retained the original decoder operation, indicated as 'MBSNet w/o MA' in Tab. IV. Furthermore, since the MA step comprises two sub-steps—dissimilarity measurement and feature accumulation—we also assessed the effectiveness of the dissimilarity measurement sub-step by disabling the feature accumulation sub-step, denoted as 'MBSNet w/o Acc.' in Tab. IV. Additionally, we examined the complete MBSNet using different parameter settings of m and N, as presented in Tab. IV and Fig. 12.

Comparing 'MBSNet w/o BIE' to the complete MBSNet with arbitrary m, we observe that MAEM plays a crucial role, as performance significantly diminishes without it. Similarly, the importance of MADM becomes apparent when

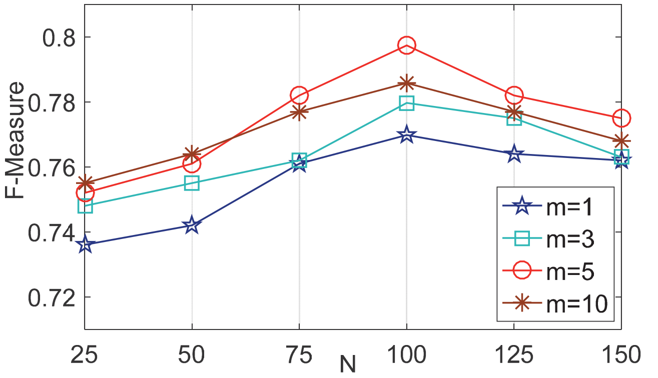


Fig. 12. The results of MBSNet on AGVS with different value of m and N, which are the numbers of frames used for motion accumulation and background image estimation, respectively.

comparing 'MBSNet w/o MA' to the complete MBSNet. However, we can't assert that MAEM is more critical than MADM solely because 'MBSNet w/o BIE' exhibits lower performance than 'MBSNet w/o MA'. This is because MADM relies on MAEM. While MAEM can function without MADM, the reverse scenario severely hampers MADM's effectiveness. Upon comparing 'MBSNet w/o MA' to 'MBSNet w/o Acc', it's evident that the sub-step dissimilarity measurement yields a performance enhancement of 5 to 8 percent across the three metrics Re, Pr, and FM. Similarly, comparing 'MBSNet w/o Acc' to 'MBSNet (m=5)' reveals that sub-step feature accumulation also leads to a 5 to 8 percent performance boost. Ultimately, it's apparent that all modules and sub-steps within the proposed method are indispensable and serve as crucial components.

Moreover, the experimental outcomes of MBSNet with varying values of m and N are presented in Tab. IV and Fig. 12. Here, m denotes the number of previous frames utilized for multi-frame accumulation, while N represents the number of image samples employed for Background Image Estimation (BIE). It's noticeable that as m and N increase, performance initially improves, but beyond a certain critical threshold, performance gradually declines. This phenomenon may occur because errors accumulate over multiple frame accumulations. Once the critical threshold is surpassed, the cumulative error outweighs the performance improvement, leading to a degradation in algorithm performance. Based on these experiments, we ultimately selected $m = 5$ and $N = 100$.

A main contribution of this work is to use manual annotation information to guide the motion accumulation in MADM. Here we conduct an additional experiment to verify whether the annotation information is consistent with the real trace of the airplane, as shown in Fig. 13. An airplane taxis straight in the upper image and middle image of Fig. 13, and then turns in the bottom image of Fig. 13. The actual motion trace of the aircraft is marked with yellow line, while the annotated turning area is marked with red line. It can be seen that the actual motion trace is consistent with the annotated result.

E. Additional Discussion

Please note that the MSD problem is different from another challenge in background subtraction, the ghost phenomenon. Ghost means that the detected object does not actually exist at the corresponding position in the input frame. In contrast, a false alarm target due to MSD is actually present in the input frame, but it is stationary rather than moving.

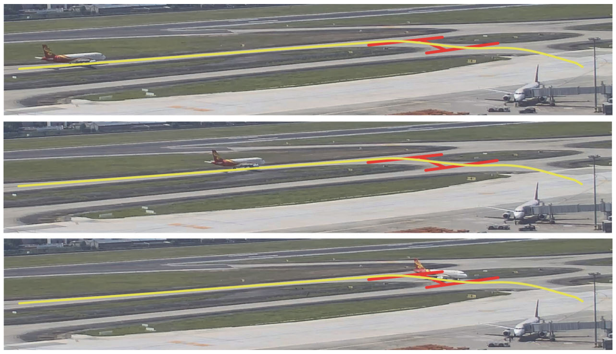


Fig. 13. Comparison between the manual annotation (red line) and the real trace of the moving aircraft (yellow line).

In fact, the MSD phenomenon is widespread. In this paper, we only study the MSD problem in airport surveillance, because there is a lack of MSD-type datasets for other scenarios. The MSD-type dataset should preferably contain a single type of moving object, so that the phenomenon of static and moving objects sharing similar appearance may occur frequently, so as to study MSD between different objects (Case1). Furthermore, the moving object in MSD-type dataset should be intermittent, so as to study MSD of the same object (Case2). Other datasets have more or less MSD instances, e.g. CDnet2014, but the number is small and hence they are not MSD-type datasets.

V. Conclusion

The Motion and Stillness Distinction (MSD) problem in airport background subtraction was studied in this paper. Firstly, we analyzed the MSD phenomenon and discussed the cause of MSD. Then MBSNet was proposed to address the MSD problem by enhancing the motion component of learned features to force the network to pay more attention to the motion information for classification. There were two modules in MBSNet, Motion Augmentation Encoder Module (MAEM) and Motion Accumulation Decoder Module (MADM). The motion component of the encoded features can be augmented in MAEM based on Background Image Estimation and further enhanced in MADM based on Motion Accumulation. The premise of Background Image Estimation and Motion Accumulation is that the aircraft on the ground are sparse and they have good motion consistency. Experiments on AGVS dataset showed that MBSNet was effective to both Case1 and Case2 of the MSD problem. The research of MSD needs MSD-type datasets, and such datasets are now fewer. It is expected that more MSD benchmarks can be made for further research. In addition, the progress on the interpretability of deep learning should also be helpful to solve the MSD problem.

References

[1] M. Mandal and S. K. Vipparthi, "An empirical review of deep learning frameworks for change detection: Model design, experimental frameworks, challenges and research needs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6101–6122, Jul. 2022.

[2] B. Garcia-Garcia, T. Bouwmans, and A. J. R. Silva, "Background subtraction in real applications: Challenges, current models and future directions," *Comput. Sci. Rev.*, vol. 35, Feb. 2020, Art. no. 100204.

[3] T. Zhou, F. Porikli, D. Crandall, L. Van Gool, and W. Wang, "A survey on deep learning technique for video segmentation," 2021, *arXiv:2107.01153*.

[4] L. A. Lim and H. Y. Keles, "Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding," *Pattern Recognit. Lett.*, vol. 112, pp. 256–262, Sep. 2018.

[5] X. Zhang, C. Shu, S. Li, C. Wu, and Z. Liu, "AGVS: A new change detection dataset for airport ground video surveillance," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20588–20600, Nov. 2022.

[6] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Comput. Sci. Rev.*, vols. 11–12, pp. 31–66, May 2014.

[7] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, "Deep neural network concepts for background subtraction: A systematic review and comparative evaluation," *Neural Netw.*, vol. 117, pp. 8–66, Sep. 2019.

[8] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[9] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1999, pp. 246–252.

[10] D.-S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005.

[11] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1163, Jul. 2002.

[12] S. Sahoo and P. K. Nanda, "Adaptive feature fusion and spatio-temporal background modeling in KDE framework for object detection and shadow removal," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1103–1118, Mar. 2022.

[13] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Background modeling and subtraction by codebook construction," in *Proc. Int. Conf. Image Process. (ICIP)*, Oct. 2004, pp. 3061–3064.

[14] O. Barnich and M. Van Droogenbroeck, "ViBE: A powerful random technique to estimate the background in video sequences," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 945–948.

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

[16] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 990–997.

[17] S. Bianco, G. Ciocca, and R. Schettini, "How far can you get by combining change detection algorithms?" in *Proc. Int. Conf. Image Anal. Process.*, vol. 10484, Sep. 2017, pp. 96–107.

[18] C. Sun, X. Wu, J. Sun, C. Sun, M. Xu, and Q. Ge, "Saliency-induced moving object detection for robust RGB-D vision navigation under complex dynamic environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 1–19, Oct. 2023.

[19] S. Isik, K. Özkan, S. Günal, and Ö. N. Gerek, "SWCD: A sliding window and self-regulated learning-based background updating method for change detection in videos," *J. Electron. Imag.*, vol. 27, no. 2, p. 1, Mar. 2018.

[20] S.-H. Lee, G.-C. Lee, J. Yoo, and S. Kwon, "WisenetMD: Motion detection using dynamic background region analysis," *Symmetry*, vol. 11, no. 5, p. 621, May 2019.

[21] F. De la Torre and M. J. Black, "Robust principal component analysis for computer vision," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, Jul. 2001, pp. 362–369.

[22] D. Meng and F. De la Torre, "Robust matrix factorization with unknown noise," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1337–1344.

[23] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Bratislava, Slovakia, May 2016, pp. 113–116.

[24] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recognit. Lett.*, vol. 96, pp. 66–75, Sep. 2017.

[25] L. A. Lim and H. Y. Keles, "Learning multi-scale features for foreground segmentation," *Pattern Anal. Appl.*, vol. 23, no. 3, pp. 1369–1380, Aug. 2020.

[26] P. W. Patil, K. M. Biradar, A. Dudhane, and S. Murala, "An end-to-end edge aggregation network for moving object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8146–8155.

[27] M. O. Tezcan, P. Ishwar, and J. Konrad, "BSUV-Net: A fully-convolutional neural network for background subtraction of unseen videos," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 2763–2772.

[28] M. O. Tezcan, P. Ishwar, and J. Konrad, "BSUV-Net 2.0: Spatio-temporal data augmentations for video-agnostic supervised background subtraction," *IEEE Access*, vol. 9, pp. 53849–53860, 2021.

[29] M. Mandal, V. Dhar, A. Mishra, S. K. Vipparthi, and M. Abdel-Mottaleb, "3DCD: Scene independent end-to-end spatiotemporal feature learning framework for change detection in unseen videos," *IEEE Trans. Image Process.*, vol. 30, pp. 546–558, 2021.

[30] G. Rahmon, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Motion U-Net: Multi-cue encoder–decoder network for motion segmentation," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 8125–8132.

[31] Y. Tang, X. Zhang, D. Chen, Z. Zhang, and H. Yu, "Motion-augmented change detection for video surveillance," in *Proc. IEEE 23rd Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2021, pp. 1–6.

[32] M. Braham, S. Piérard, and M. Van Droogenbroeck, "Semantic background subtraction," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4552–4556.

[33] A. Cioppa, M. V. Droogenbroeck, and M. Braham, "Real-time semantic background subtraction," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 3214–3218.

[34] M. C. Bakkay, H. A. Rashwan, H. Salmane, L. Khoudour, D. Puig, and Y. Ruichek, "BSCGAN: Deep background subtraction with conditional generative adversarial networks," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 4018–4022.

[35] Y. Yang, J. Ruan, Y. Zhang, X. Cheng, Z. Zhang, and G. Xie, "STPNet: A spatial–temporal propagation network for background subtraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2145–2157, Apr. 2022.

[36] M. Mandal and S. K. Vipparthi, "Scene independency matters: An empirical study of scene dependent and scene independent evaluation for CNN-based change detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2031–2044, Mar. 2022.

[37] S. D. Jain, B. Xiong, and K. Grauman, "FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2126.

[38] C. Ventura, M. Bellver, A. Girbau, A. Salvador, F. Marques, and X. Giro-I-Nieto, "RVOS: End-to-end recurrent network for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5272–5281.

[39] X. Lu, W. Wang, J. Shen, Y.-W. Tai, D. J. Crandall, and S. C. H. Hoi, "Learning video object segmentation from unlabeled videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8957–8967.

[40] S. W. Oh, J.-Y. Lee, K. Sunkavalli, and S. J. Kim, "Fast video object segmentation by reference-guided mask propagation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7376–7385.

[41] Z. Wang, J. Xu, L. Liu, F. Zhu, and L. Shao, "RANet: Ranking attention network for fast video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3977–3986.

[42] B. Lee and M. Hedley, "Background estimation for video surveillance," in *Proc. Int. Conf. Image Vis. Comput.*, Jan. 2002, pp. 315–320.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[44] A. Vaswani, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, vol. 30, Dec. 2017, pp. 5999–6009.

[45] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[46] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[47] P. O. Pinheiro, T. Y. Lin, R. Collobert, and P. Dollar, “Learning to refine object segments,” in *Proc. Eur. Conf. Comput. Vis.*, vol. 9905, Oct. 2016, pp. 75–91.

[48] 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*, Jun. 2014, pp. 393–400.

[49] J. Cheng, Y. Tsai, S. Wang, and M. Yang, “SegFlow: Joint learning for video object segmentation and optical flow,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 686–695.

[50] X. Zhang, S. Wang, H. Wu, Z. Liu, and C. Wu, “ADS-B-based spatiotemporal alignment network for airport video object segmentation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17887–17898, Oct. 2022.

[51] D. Cremers and S. Soatto, “Motion competition: A variational approach to piecewise parametric motion segmentation,” *Int. J. Comput. Vis.*, vol. 62, no. 3, pp. 249–265, May 2005.

[52] T. Brox and J. Malik, “Object segmentation by long term analysis of point trajectories,” in *Proc. Eur. Conf. Comput. Vis.*, vol. 6351, Sep. 2010, pp. 282–295.

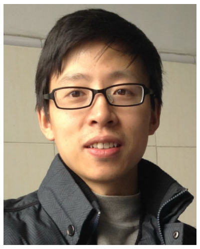
[53] M. Yue, G. Fu, M. Wu, Y. Zhao, and S. Zhang, “Vehicle motion segmentation via combining neural networks and geometric methods,” *Robot. Auto. Syst.*, vol. 155, Sep. 2022, Art. no. 104166.

[54] M. Siam, H. Mahgoub, M. Zahran, S. Yogamani, M. Jagersand, and A. El-Sallab, “MODNet: Motion and appearance based moving object detection network for autonomous driving,” in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2859–2864.

[55] H. Rashed, A. E. Sallab, and S. Yogamani, “VM-MODNet: Vehicle motion aware moving object detection for autonomous driving,” in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 1962–1967.

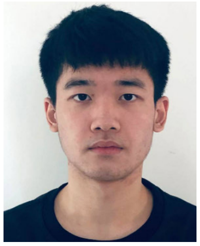
[56] L. Mariotti and C. Eising, “Spherical formulation of geometric motion segmentation constraints in fisheye cameras,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 4201–4211, May 2022.

[57] A. Sobral, “BGSLibrary: An OpenCV C++ background subtraction library,” in *Proc. IX Workshop de Visao Computacional*, Jun. 2013, pp. 1–6.



Xiang Zhang received the B.S. and M.S. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2003 and 2006, respectively, and the Ph.D. degree from Shanghai Jiaotong University, Shanghai, China, in 2010. He is currently an Associate Professor with the School of Information and Communication Engineering, University of Electronic Science and Technology of China. His research interests include intelligent transportation, video analysis, and machine learning.

Yingqi Tang received the B.S. degree from Chongqing University in 2019 and the M.S. degree from the University of Electronic Science and Technology of China in 2022. His research interests include autopilot and computer vision.



Maozhang Zhou received the B.S. degree from Hangzhou Dianzi University. He is currently pursuing the M.S. degree with the University of Electronic Science and Technology of China, Chengdu, China. His research interests include machine learning and semantic segmentation.

Celimuge Wu (Senior Member, IEEE) received the Ph.D. degree from The University of Electro-Communications, Japan. He is currently a Professor and the Director of the Meta-Networking Research Center, The University of Electro-Communications. His research interests include vehicular networks, edge computing, the IoT, and AI for wireless networking and computing. He was a recipient of the 2021 IEEE Communications Society Outstanding Paper Award, the 2021 IEEE Internet of Things Journal Best Paper Award, the IEEE Computer Society 2020 Best Paper Award, and the IEEE Computer Society 2019 Best Paper Award Runner-Up. He is an IEEE Vehicular Technology Society Distinguished Lecturer. He is the Vice Chair (Asia–Pacific) of the IEEE Technical Committee on Big Data (TCBD). He serves as an Associate Editor for IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, and IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING.

Zhi Liu (Senior Member, IEEE) received the Ph.D. degree in informatics from the National Institute of Informatics. He is currently an Associate Professor with The University of Electro-Communications. His research interests include video network transmission and mobile edge computing. He is an Editorial Board Member of *Wireless Networks* (Springer) and IEEE OPEN JOURNAL OF THE COMPUTER SOCIETY.