JanusNet: Detection of Moving Objects from UAV Platforms

Yuxiang Zhao, Khurram Shafique, Zeeshan Rasheed, Maoxu Li
Novateur Research Solution
{yzhao, kshafique, zrasheed, mli}@novateur.ai

Abstract

In this paper, we present JanusNet, an efficient CNN model that can perform online background subtraction and robustly detect moving targets using resource-constrained computational hardware on-board unmanned aerial vehicles (UAVs). Most of the existing work on background subtraction either assume that the camera is stationary or make limiting assumptions about the motion of the camera, the structure of the scene under observation, or the apparent motion of the background in video. JanusNet does not have these limitations and therefore, is applicable to a variety of UAV applications. JanusNet learns to extract and combine motion and appearance features to separate background and foreground to generate accurate pixel-wise masks of the moving objects. The network is trained using a simulated video dataset (generated using Unreal Engine 4) with ground-truth labels. Results on UCF Aerial and Kaggle Drone videos datasets show that the learned model transfers well to real UAV videos and can robustly detect moving targets in a wide variety of scenarios. Moreover, experiments on CDNet dataset demonstrate that even without explicitly assuming that the camera is stationary, the performance of JanusNet is comparable to traditional background subtraction methods.

Keywords: Background subtraction, foreground segmentation, moving objects detection, optical flow, UAV, neural network, CNN, video surveillance, tracking.

1. Introduction

The first step in many video processing pipelines is separation of foreground objects from the background. This is typically done through background subtraction algorithms that attempt to identify the most relevant parts of the video stream by online learning and modeling the characteristics of the background and finding pixels that do not conform to the learned model. As one of the most critical components of automated visual surveillance systems, background separation problem has been extensively studied in computer vision literature [14]. Many approaches have been presented to solve its technical challenges, such as illumination changes, dynamic backgrounds (e.g., fluttering leaves, waving flags, water fountains, etc.), camera jitters, and shadows, etc. [6, 40, 42, 50, 51, 4].

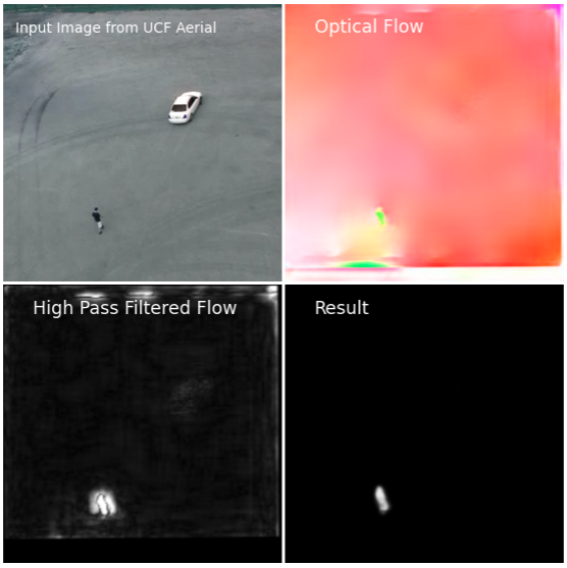

Figure 1. Example on UCF Aerial Action Dataset [2]. The person in the video is walking while the car is stationary.

A large portion of video data generated these days is captured by mobile sensors, e.g., handheld smart phones, wearable devices, and sensors mounted on small UAVs. However, most of the existing work in background subtraction has focused on the videos from static and PTZ cameras used in video surveillance. While some approaches have also been proposed to specifically handle this use case (See Section 2 for a brief overview and [7] for detailed review of the literature in this area), detection of moving objects from moving platforms, especially UAVs, remains a challenging problem. Here, we propose JanusNet, a novel deep network that is capable of efficiently detecting small moving targets from UAVs. JanusNet learns to extract and combine dense optical flow and generate a coarse foreground attention map using high-pass filters. Then JanusNet combines optical flow, foreground attention maps and appearance features to

separate foreground motion from background motion and to generate accurate pixel-wise masks of the moving objects in the scene. We demonstrate the performance of our model using quantitative and qualitative results on both real-world and synthetic datasets. The organization of the paper is as follows. In Section 2, we present a brief survey of the related work and compare the key aspects of proposed model with the state-of-the-art. In Section 3, we detail the JanusNet architecture, its components, and training processes. In Section 4, we present quantitative and qualitative results to validate our claims. A brief discussion of different aspects of the model and results is provided in Section 5 and the paper is concluded in Section 6.

2. Related Work

A large number of background subtraction methods have been proposed in the literature [7, 14]. Earlier methods attempted to distinguish background and foreground pixels by using simple statistical measures, such as the parametric Gaussian Mixture Models [40, 8], nonparametric kernel density models [39], and local binary patterns models [18]. In the last decade, low rank subspace learning models [16, 49, 34, 17] have also gain popularity. By assuming that the backgrounds have a low-rank structure and that the foregrounds are sparse, these methods exploit the mathematical framework of matrix completion for highly efficient and effective background subtraction in a variety of situations. More recently, deep learning based methods have been shown to outperform the traditional approaches on a variety of benchmark datasets, such as CDNet14 [44]. These methods [5, 3, 23] use supervised learning approaches and exploit labeled training datasets to learn to produce foreground segmentation masks for each video frame.

Most of the above approaches assume that the camera is static. Even the approaches that attempt to tackle sensor motion either impose restrictions on the sensor motion types, e.g., jitter, panning cameras, or the extent of the camera motion, e.g., a handheld camera mostly viewing a small area of a 3D scene while moving freely. This holds true even for some of the most recent algorithms that employ subspace learning [47, 28, 13] or modern deep learning techniques for background subtraction [24, 26]. Therefore, these algorithms cannot be directly applied to videos from sensors on-board mobile platforms, such as UAVs, which cover large scene regions with rapidly changing background. In addition to handling sensor motion, background subtraction for UAVs face two additional challenges. First, foreground objects in UAV videos, such as pedestrians, are typically very small (less than 10×10 pixels) compared to the size of video frame and must be disambiguated from clutter. Second, many UAVs and mobile platforms must operate within size, weight, and power budgets, and therefore have limited onboard computational resources.

The methods that do attempt to solve the background subtraction problem for mobile sensors with rapidly changing background often borrow the basic methodology from algorithms for static cameras [36, 19]. These methods first create new representations that cancel the effect of platform motion, e.g., background mosaics (generated by stitching imagery from subsequent frames) [46, 43] or explicit 3D models [31]. The background subtraction techniques borrowed from static camera domain are then applied to these new representations. These approaches are not only time complex but are also error prone as minor errors in alignment are propagated and compounded over time. Some recent approaches in this class of algorithms attempt to tackle this complexity by limiting the background model size close to the image size, e.g., by using only the recent few images to create the model [27, 48]. All of these techniques still make additional assumptions about the sensor motion or the scene structure, such as, high altitude sensors, largely planar scenes (for mosaicking, plane-parallax models, etc.) to create these representations that may not necessarily hold in general scenarios.

Almost all the approaches discussed above mainly use image features to discriminate foreground pixels from background. Another class of algorithms exploit motion features, such as key point trajectories and optical flow to make this determination [21, 38, 45, 29, 30]. These approaches use a variety of constraints (geometric, low-rank background motion, etc.) to separate the motion vectors of independently moving objects from those generated by the scene background. The advantage of these approaches is that they do not need to create explicit background models and in many cases do not make limiting assumptions about the scene structure or the motion of the camera. However, without appearance and context features, these methods struggle to determine object boundaries and reject background motions such as waving trees and shadows. In addition, these approaches suffer from the difficulties of reliably detecting features and generating long-term feature trajectories or robust extraction of dense optical flow from videos in real time.

The proposed JanusNet model addresses the above-mentioned limitations of the state-of-the-art while also maintaining some of their most advantageous attributes. As opposed to many existing methods, the network does not make limiting assumptions about the sensor motion or the structure of the scene and is capable of operating in a variety of scenarios. It also does not require creation and maintenance of explicit background models, a memory and time-complex step. It leverages recent advances in deep learning techniques for robust estimation of dense optical flow from videos. This enables JanusNet to exploit both motion (dense optical flow) and image attributes (deep fea-

tures) to identify independently moving objects from moving camera videos. The joint modeling of motion and appearance features for foreground segmentation has also recently been suggested in [9, 20, 33], etc. However, most of these methods simply concatenate raw optical flow features and image features and pass them to convolutional layers that generate foreground segmentation. Our experiments have shown that while such models perform well on large and known objects, they do not perform well on previously unseen scenes/objects or disambiguating small objects as in the case of UAV videos. JanusNet tackles these challenges by using high-pass filters to generate multi-scale foreground attention maps and using a context layer that learns to combine these attention maps with raw optical flow, deep appearance features to improve results.

3. Approach

Figure 2 summarizes the JanusNet architecture. Given two adjacent frames of a video, a sub-network first roughly estimates a global parametric motion between the two frames and produces roughly aligned frames. The two roughly aligned frames are used by the optical flow subnetwork (that includes feature pyramid, warping, correlation, and flow estimation) similar to PWC-Net [41] to generate dense optical flow from low to high resolutions. The resulting estimated optical flow is then passed through multiple high-pass filters to highlight pixels with motions different from their surroundings. Finally, the context layer combines these highlighted pixels with image features, optical flow, as well as foreground priors (up-sampled foreground estimation from lower resolution, at the lowest resolution, no foreground priors are not used) to fine tune the results based on semantic and contextual information. The network is trained end-to-end using two training goals, one for optical flow generation, and another for foreground estimation. The remaining part of this section explains the designs and ideas behind each of these sub-networks, and details the training procedures.

3.1. Rough Global Motion Estimation

When the sensor is moving, the two adjacent video frames are not aligned with each other. Though the unaligned frames can be used directly for optical flow estimation and subsequent foreground separation, the large global motion between the frames may sometimes lead to poor optical flow performance, which in turn affects the quality of foreground. Therefore, we use a global motion estimator to roughly align the two frames to reduce global motion. Our global motion estimator follows a similar design as the deep homography model [10] except that our model ingests higher resolution inputs and uses fewer layers and channels for the computational efficiency. In particular, the network uses 10 layers of convolutional layers with stride of 2 in every other layer. The first 4 layers have $32, 64, 96$, and 128 channels respectively whereas the remaining 6 layers each have 128 channels. Finally the network uses a linear layer to output the movement of 4 predefined corners of the image pairs to compute homography.

Method	Resolution	Time/frame
SIFT + RANSAC	480x480	37.4ms
ORB + RANSAC	480x480	23.2ms
Original DeepH	resize to 128x128	7.5ms
Ours, small DeepH	resize to 320x320	5.2ms

Table 1. Homography Transform Speed Comparison. SIFT and ORB are tested on CPU. Original DeepH and our customized DeepH are tested on GTX1070, image resize time included.

Compared with SIFT [25] + RANSAC, ORB [35] + RANSAC, a deep homography estimator is much faster but not as accurate (Table 1). However, in contrast to many existing approaches that assume a specific motion model and then use it as the basis of motion compensation and model generation, the type of motion model used (affine, homography, etc.) or the accuracy of estimation is not important here. Since the goal of this global motion estimation and alignment step is simply to reduce the effect of global motion on optical flow quality rather than completely eliminate the ego-motion from imagery, a rough alignment is sufficient for the model to produce accurate foreground segmentation (See Section 4).

3.2. Optical Flow Sub-Network

The Optical Flow Sub-Network in JanusNet consists of feature pyramid extraction, warping, correlation, and flow estimation layers. JanusNet follows a similar design as PWC-Net [41] but trades accuracy for speed as discussed below:

3.2.1 Feature Pyramid Extraction Layer

For feature pyramid extraction, the network uses L-level pyramids consisting of convolutional layers (with input images at 0^{th} level and the the deepest layer at the L^{th} level) to extract image features at different resolutions. To extract features from input image I_t at level l: C_t^l, the network uses ResNet-style convolutional layers to down-sample C_t^{l-1} by 2. In our experiments, we used $L = 4$ levels with channels 16, 32, 64, and 96 respectively.

3.2.2 Feature Warping Layer

For each level, the network warps features of Image I_{t-1} towards Image I_t using the up-sampled optical flow from $(l+1)^{\text{th}}$ level O^{l+1}:

$$C_w^l(x) = C_{t-1}^l(x + up(O^{l+1})(x)) \quad (1)$$

where x is the pixel position, $up(O^{l+1})(x)$ is the upsampled optical flow from the $(l+1)^{\text{th}}$ level at position x.

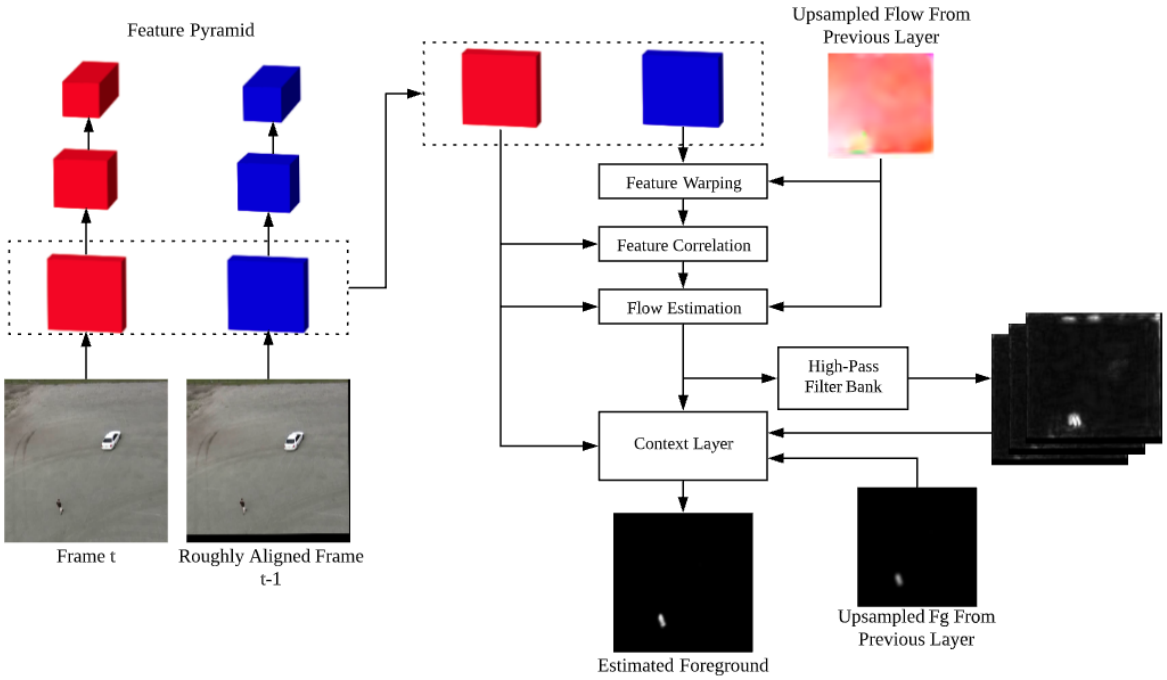

Figure 2. Network Structure

3.2.3 Feature Correlation Layer

Given the features of the warped image at the l^{th} level: C_w^l, the network calculates correlation scores for each of its pixel with its corresponding neighboring pixels in C_t^l. We define the correlation score as:

$$corr^l(x_1, x_2) = \frac{1}{N^l}(C_t^l(x_1))^T(C_w^l(x_2)) \qquad (2)$$

where x_1, x_2 are pixel positions, T is the transpose operator, N^l is the number of channels for l^{th}-level feature pyramid. Calculating correlation between all possible combinations of pixel pairs x_1, x_2 is computationally very expensive. As a trade-off, for each pixel x_1 in image features C_w^l, the network only computes correlations of pixel pairs within a $d \times d$ square region centered at x_1. The time complexity of this module is $O(d^2 \times H \times W \times N)$, where H, W are the dimensions of C^l. At low levels, where H and W are large, the correlation can still be a bottleneck depending on the available computational power on-board the platform. In our implementation, the network only implements correlation layers for levels $l \geq 2$ and uses the search region of $d = 5$. Because of these optimizations, JanusNet relies on the rough global motion estimation and initial warping (Section 3.1) to tackle large global motions and to improve optical flow performance in resource constrained environments.

3.2.4 Flow Estimation Layer

The Flow Estimation Layer at level l consists of ResNet-style convolutional layers that take concatenated correlation $corr^l$, features of image C_t^l, and up-sampled optical flow $up(O^{l+1})$ as inputs to output estimated optical flow O^l. Compared to PWC-Net, JanusNet uses a reduced number of layers (4 layers with channels 64, 32, 32, 2), where the 2 channels of the final layer correspond to horizontal and vertical pixel movements.

3.3. Foreground Attention Maps using High-pass Filter Bank

As mentioned earlier, JanusNet exploits the optical flow of the scene from the flow estimation layer to separate foreground and background motions. As opposed to existing methods discussed in Section 2 that use explicit motion factorization or subspace modeling for this task, JanusNet uses a convolutional model (learned in a supervised fashion) to identify independently moving objects. As we will show in Section 4, learning such a model directly from raw optical flow is extremely challenging given the large number of variables (scene structure, viewing geometry, etc.) that the output depends upon. Therefore, to guide this learning, JanusNet employs a bank of high-pass filters that act as a focus-of-attention mechanism and highlight the regions that are more likely to contain independently moving objects, i.e., objects that exhibit different motion from their surroundings:

$$m_k^l = O^l - Avg(O^l, s_k^l) \qquad (3)$$

where m_k^l is the filtered optical flow at level l from high-pass filter k, and $Avg(O^l, s_k^l)$ is an average filter, with ker-

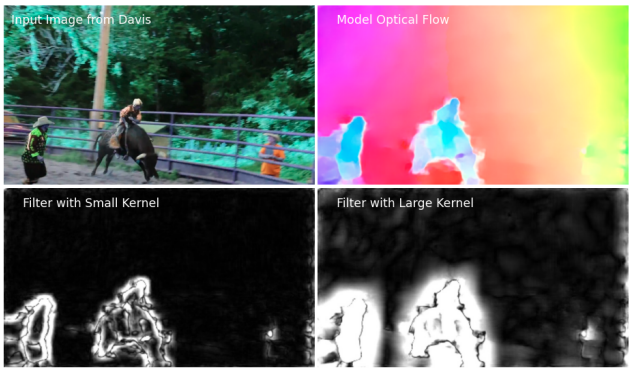

Figure 3. Output of two high-pass filters with different values s_k

nel size s_k^l at level l. Different values of s_k^l provide attention to objects at different scale (Figure 3). In our experiments, we have found that $k = 2$ with with $s_1^l = 2^{l+1} + 1$ and $s_2^l = 2^{l+3} + 1$ provide sufficient coverage and accuracy for both large and small objects, though additional filters can easily be added without any significant loss of performance.

The network uses the output of each filters, m_k^l to generate attention maps (at different scales) as follows:

$$f_k^l(x) = min(1.0, \frac{||m_k^l(x)||_2^2}{v}) \tag{4}$$

$||m_k^l(x)||_2^2$ is the magnitude of the filtered optical flow m_k^l, f_k^l is the foreground attention map of filter k at level l, v is the minimum motion of a pixel x to consider it as a moving pixel. If $||m_k^l(x)||_2^2 \geq v$, then $f_k^l(x) = 1.0$, else $f_k^l(x) < 1.0$. In our experiments, we used $v = 0.5$, i.e., the motion vectors smaller than 0.5 pixels at pyramid level 1 (corresponds to 1 pixel at image resolution) between 2 frames. For all practical purposes, motions smaller than 1 pixels between two frames can be discarded without affecting the performance of the model.

3.4. Context Layer

The Context Layer produces the pixel-wise foreground mask by combining information from different sources that include: i) image features from video frames, ii) raw optical flow, iii) foreground attention maps of different scales from high-pass filter bank, and iv) foreground priors in the form of output from lower resolution:

$$fg^l = convs(C_t^l, corr^l, O^l, F_1, F_2, \ldots, F_k) \tag{5}$$

$$F_k = tanh(up(fg^{l+1}) \times f_k^l) \tag{6}$$

where fg^l is the foreground segmentation at l^{th} level, $convs(.)$ represents convolutional layers. In Eq 6, we multiply $up(fg^{l+1})$ with foreground attention map f^l to exclude motionless pixels. $tanh$ is used to normalize the input of the CNN to range between -1 to 1.

Using supervised training, the context layer learns to combine and exploit semantics from images, motion cues from optical flow, attention maps, and priors from lower resolutions to determine object boundaries, remove shadows and spurious background objects, and identify pixels in the image belonging to independently moving objects.

3.5. Loss Function

The model is trained using two training goals: optical flow and foreground segmentation. Optical flow is trained with mean squared loss for each level with weights α^l; foreground segmentation is trained with binary cross entropy loss with the same weights α^l. In our experiments, we set α to be (1.6, 0.4, 0.2, 0.1) from level $l = 1$ to $l = 4$ respectively.

$$L_{flow} = \sum_l (\alpha^l \times MSE(O^l, O_{gt}^l)) \tag{7}$$

$$L_{fg} = \sum_l (\alpha^l \times BCE(sigmoid(fg^l), fg_{gt}^l)) \tag{8}$$

3.6. Janus Synthetic Video Dataset

Due to lack of annotated datasets for background subtraction from moving camera, to train and validate the proposed model, we created Janus Dataset, a dataset of synthetic videos created in Unreal4 game engine. The dataset includes city, forest, beach, and castle scenes. We used various 3D models with animations and applied simple AIs to let them move randomly and intermittently. We set the camera above the ground and let it move and rotate randomly between frames. For each environment, we captured 10 to 20 videos, each with 200 frames, and each having different lighting effects, object positions, etc. We then used the Airsim package [37] to generate the ground truth segmentation. We made JanusDataset publicly available on Kaggle to support researches on UAV background subtraction.

3.7. Training

Since optical flow itself does not rely on the foreground segmentation, to provide a good initialization to our model, we first train the optical flow sub-network on the Flying Chairs [11] dataset until the loss converges. Then we train one batch on optical flow using Flying Chairs dataset and then the second batch on foreground segmentation using Janus Dataset and so on. Janus Dataset is splited into training and validation videos where the training videos include city, forest, and beach scenes, and the validation videos include the castle scene.

We used various data augmentation techniques: adding Gaussian noise, random rotation, random resize, random

Figure 4. Examples of Janus Dataset with associated groundtruth.

Method	Precision	Recall	Speed
JanusNet	**0.767**	**0.687**	40.8ms(GPU)
w/o Attention	0.704	0.643	40.0ms(GPU)
NoHomography	0.663	0.661	35.2ms(GPU)

Table 2. Performance of JanusNet on Unreal4 videos. Precision and Recall are measured pixel-wise against ground truth. The evaluation is performed on 640x640 images on IntelI7@2.7GHZ CPU with a GTX1070 GPU

shift, random flip, artificial motion blurs, so that the trained model does not overfit on the relatively small Unreal4 training videos. To train the model, we used Adam optimizer with starting learning rate of 1e-4 and gradually decreased the learning rate. The training was continued until the learning rate reached 1e-8.

4. Results and Analysis

4.1. Quantitative Results on Janus Dataset

As mentioned earlier, due to the lack of annotated datasets for background subtraction from moving camera or UAVs, quantitative results on state-of-the-art models are not available or properly bench-marked. DAVIS [32] and Seg-track v2 [22] datasets are not appropriate for evaluation because i) include stationary foreground objects, ii) exclude moving object if they are not the main targets) is an object segmentation dataset. Most of the literature in this area, therefore, relies on qualitative results to demonstrate the proposed approach. A secondary contribution of this paper is Janus Dataset (Section 3.6), a synthetic video dataset with moving cameras that we hope would support such bench-marking and comparative evaluation efforts in the future. We quantitatively evaluated the performance of our approach on this dataset. As mentioned in the previous section, we used city, forest, and beach scenes in the dataset for training the model, and we use castle scenes to evaluate the model. Table 2 shows the pixel-level precision and recall of our approach on Janus dataset. Note that the camera is continuously moving in all videos. Moreover, most of the moving targets in the videos are significantly small as compared to other background subtraction datasets in the literature. Given these challenges, the model performs remarkably well and gives precision and recall of 0.767 and 0.687 respectively.

We also used this dataset to evaluate the contribution of different sub-networks and components. Comparing JanusNet with and without attention maps from high-pass filters, our results demonstrate that the use of attention maps significantly improve precision while also mildly improving recall (See Table 2 and Fig 5). For example in the 1st column in Fig 5, with attention maps, JanusNet is the only method that does not produce false positives on the stationary car. We also compared JanusNet with and without rough global motion compensation. Our results show that the rough global motion compensation also has an impact on both the precision and recall of the system. As discussed in Section 3.2, the use of rough motion compensation enables us to optimize the optical flow network. Given ample computational resources, the motion compensation step can be eliminated by i) adding more layers to the optical flow network and ii) increasing the search radius, d in the feature correlation layer.

4.2. Quantitative Results on CDnet14

CDnet14 [44] is a widely used foreground segmentation benchmark dataset. We trained JanusNet on the full CDnet14 training dataset without using Unreal4 videos. Most videos in CDnet14 are stationary-camera videos, so we turned off the explicit global motion estimation on these videos. Otherwise we used the same model architecture and training procedures as described in Section 3. JanusNet reached F-Measure of 0.56 which is similar to GMM [50] and KDE [12]. (The ground truth of **CDnet14 dataset includes stationary foreground objects**, but JanusNet uses high-pass filters to exclude stationary objects so it puts JanusNet at a disadvantage. Whether stationary foreground objects are wanted depends on applications). On the other hand, FgSegNetV2 [23] and DCBS [4] outperform our model on this dataset (see Table 3). However, JanusNet has several advantages over these approaches: i) JanusNet is much smaller and faster; ii) JanusNet is more general; FgSegNetV2 requires training on the exact scene. DCBS requires a corresponding background image, which is usually not available in practical scenarios; JanusNet works on both stationary and moving cameras. While FgSegNetV2 can handle some camera motion, it is limited in this capability as it must be trained on the same scene.

4.3. Qualitative Results on UAV Videos

We also compared these algorithms on UCF Aerial [2] and Kaggle Drone videos [1]. To our knowledge, there is no benchmark UAV foreground segmentation datasets with ground truth labels, so we only provide qualitative results as shown in Fig 5. The results show that JanusNet successfully separates foregrounds and backgrounds in a vari-

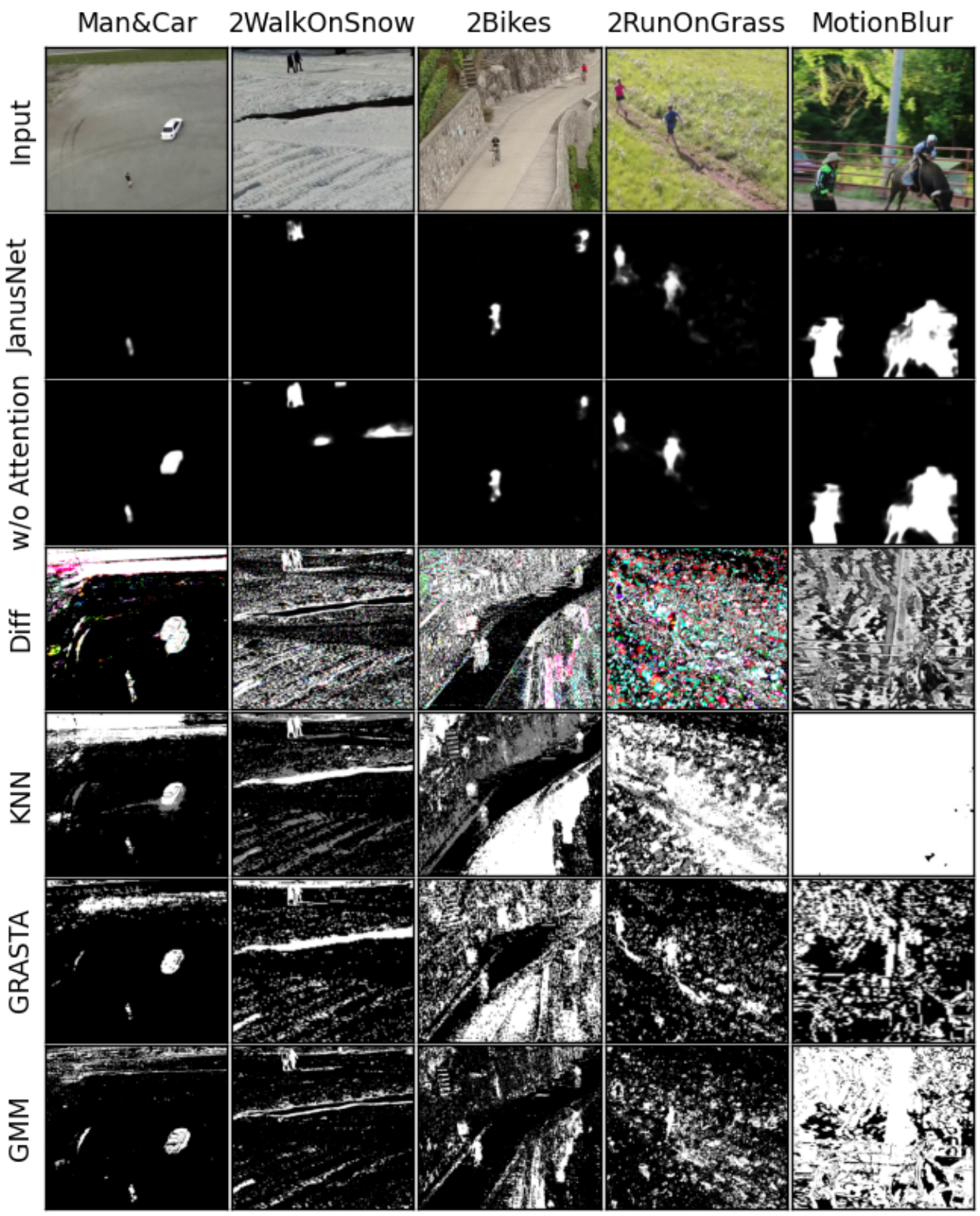

Figure 5. Performance on various videos. Column 1 is from UCF Aerial Action [2], Column 2-4 are from Kaggle Drone Videos [1], Column 5 is from Davis Dataset [44]. All videos have camera movements, and column5 has significant motion blur. The camera movements can be seen from the "Diff" (frame difference) row. The "w/o Attention" row is JanusNet but without the high pass filter bank.

Method	Moving Camera?	Unseen Videos?	F-Measure
FrameDiff	False	True	0.21
GRASTA[15]	False	True	0.36
JanusNet	**True**	**True**	0.56
GMM[50]	False	True	0.57
KDE[12]	False	True	0.57
KNN[51]	False	True	0.59
DCBS[4]	False	Limited	0.76
FgSegV2[23]	Limited	False	**0.97**

Table 3. Performance on CDnet14. Although DCBS can work on unseen videos it requires background images which are often unavailable

ety of scenarios. In the training Unreal4 videos, the foreground objects only include pedestrians, deer, and various types of automobiles. JanusNet successfully adapts to real world foreground objects that are not in the training set (bikes and bulls in column3 and column5 of Fig 5). This shows the transfer-ability of JanusNet models to different environments. It also demonstrates that the model can successfully utilize and switch to different information sources, for example using motion and attention models when the the scene semantics do not match the learned semantic models.

5. Discussion

In this section, we discuss some of the questions that were raised during our internal review of the paper.

Comment 1: *How do you handle the domain gap between the synthetic data used for training the model and the real-world videos?*

Response: As shown in Fig 5, JanusNet performs well on unseen real videos. To accomplish this, we have included many real world challenging scenarios: waving trees, shadows, different lightings, water reflections, etc in Janus Dataset. We have also used artificial motion blurs to mimic real world motion blurs created from camera movements. Moreover, by incorporating multiple sources of information, especially motion features, the model is not simply reliant appearance features and image semantics where the domain gap is generally more prominent.

Comment 2: *What is the significance of hyper-parameters, d, v, and s_k, how are they selected, and how do they impact the results?*

Response: The hyper-parameter d defines the search space for feature correlation and is included in many state-of-the-art optical flow models. It governs the maximum motion that the optical flow module can detect. At 480×480 resolution, the chosen value of $d = 5$ enables detection of motions up to $52.5 miles/hour$ from UAVs flying at $20m$ altitude, which is sufficient for most applications. On the other hand, the hyperparameter v indicates the minimum motion that the network tries to detect and we believe $v = 0.5$ (1 pixel) motion between adjacent frames is general enough for most applications. The kernel sizes s_k in the high-pass filter bank help highlight attention models and identify objects at different sizes. As shown in Fig 5, JanusNet can detect pedestrians from far-field and also detect a bull-rider in near-field using the same hyperparameters.

Comment 3: *Can you provide results from SOTA model such as FgSegNetV2, DCBS, SegFlow on UAV videos?*

Response: FgSegNetV2 [23] is a scene specific model that requires to be trained on a specific scene to work well on that scene. DCBS [4] requires a corresponding background image. Both are not possible for arbitrary UAV videos. SegFlow and other similar models are trained using a different training goal that does not match our foreground segmentation application. Moreover, at a high level, our model without explicit homography and attention maps (Table 2) is conceptually similar to their model as applied to the problem-at-hand.

Comment 4: *Can you use DAVIS and Seg-Track v2 Datasets for testing?*

Response: DAVIS and Seg-Track v2 datasets are for object segmentation. It differs from our application in two ways, i) it includes stationary objects; and ii) it excludes moving objects if they are not the main targets.

6. Conclusion

In this work, we present JanusNet: a fast but effective foreground segmentation model for videos from UAVs and moving cameras. JanusNet uses the recent advancements in deep learning and employs a convolutional neural network that learns to combine dense optical flow, attention models, image features, and foreground priors to produce accurate foreground segmentation in a variety of scenarios. JanusNet is trained using a simulated video dataset generated with Unreal-4 Engine. As opposed to many deep learning methods for background separation, JanusNet can successfully detect novel foreground objects from unseen videos taken from moving cameras. Our qualitative results (Fig.5) on UCF Aerial[2] and Kaggle Drone videos [1] datasets demonstrate that the network is capable of transferring its learning to real world datasets and can detect small moving targets in a variety of scenarios. JanusNet model also uses an efficient architecture and can process 640×640 videos at 25fps on Nvidia GTX1070 GPU and 3.1fps on Nvidia Jetson Nano.

7. Acknowledgements

This work was supported by AFOSR contract FA9550-18-C-0050. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the US Government.

References

[1] Drone videos. https://www.kaggle.com/kmader/drone-videos. Accessed: 2020-09-16.

[2] Ucf aerial action data set. https://www.crcv.ucf.edu/data/UCF_Aerial_Action.php. Accessed: 2020-06-26.

[3] Long Ang Lim and Hacer Yalim Keles. Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding. *arXiv*, pages arXiv–1801, 2018.

[4] Mohammadreza Babaee, Duc Tung Dinh, and Gerhard Rigoll. A deep convolutional neural network for background subtraction. *arXiv preprint arXiv:1702.01731*, 2017.

[5] Marc Braham and Marc Van Droogenbroeck. Deep background subtraction with scene-specific convolutional neural networks. In *2016 international conference on systems, signals and image processing (IWSSIP)*, pages 1–4. IEEE, 2016.

[6] Volkan Cevher, Aswin Sankaranarayanan, Marco F Duarte, Dikpal Reddy, Richard G Baraniuk, and Rama Chellappa. Compressive sensing for background subtraction. In *European Conference on Computer Vision*, pages 155–168. Springer, 2008.

[7] Marie-Neige Chapel and Thierry Bouwmans. Moving objects detection with a moving camera: A comprehensive review. *arXiv preprint arXiv:2001.05238*, 2020.

[8] M. Chen, X. Wei, Q. Yang, Q. Li, G. Wang, and M. Yang. Spatiotemporal gmm for background subtraction with superpixel hierarchy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1518–1525, 2018.

[9] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the IEEE international conference on computer vision*, pages 686–695, 2017.

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016.

[11] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[12] Ahmed Elgammal, Ramani Duraiswami, David Harwood, and Larry S Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, 2002.

[13] K. Gilman and L. Balzano. Panoramic video separation with online grassmannian robust subspace estimation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 643–651, 2019.

[14] Jhony Giraldo, Huu Ton Le, and Thierry Bouwmans. Deep learning based background subtraction : a systematic survey. In *Handbook of Pattern Recognition and Computer Vision*, number 6, pages 51–73. WORLD SCIENTIFIC, Apr. 2020.

[15] Jun He, Laura Balzano, and John Lui. Online robust subspace tracking from partial information. *arXiv preprint arXiv:1109.3827*, 2011.

[16] Jun He, Laura Balzano, and Arthur Szlam. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1568–1575. IEEE, 2012.

[17] Jun He, Dejiao Zhang, Laura Balzano, and Tao Tao. Iterative grassmannian optimization for robust image alignment. *Image and Vision Computing*, 32(10):800–813, 2014.

[18] M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):657–662, 2006.

[19] M. Irani and P. Anandan. A unified approach to moving object detection in 2d and 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):577–589, 1998.

[20] 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 *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2126, 2017.

[21] L. Kurnianggoro, A. Shahbaz, and K. Jo. Dense optical flow in stabilized scenes for moving object detection from a moving camera. In *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, pages 704–708, 2016.

[22] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2192–2199, 2013.

[23] Long Ang Lim and Hacer Yalim Keles. Learning multi-scale features for foreground segmentation. *Pattern Analysis and Applications*, 23(3):1369–1380, 2020.

[24] Long Ang Lim and Hacer Yalim Keles. Foreground segmentation using convolutional neural networks for multiscale feature encoding. *Pattern Recognition Letters*, 112:256–262, 2018.

[25] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[26] Tsubasa Minematsu, Atsushi Shimada, Hideaki Uchiyama, Vincent Charvillat, and Rin-ichiro Taniguchi. Reconstruction-based change detection with image completion for a free-moving camera. *Sensors*, 18(4), 2018.

[27] Tsubasa Minematsu, Hideaki Uchiyama, Atsushi Shimada, Hajime Nagahara, and Rin ichiro Taniguchi. Adaptive background model registration for moving cameras. *Pattern Recognition Letters*, 96:86–95, 2017.

[28] B. E. Moore, C. Gao, and R. R. Nadakuditi. Panoramic robust pca for foreground–background separation on noisy, free-motion camera video. *IEEE Transactions on Computational Imaging*, 5(2):195–211, 2019.

[29] M. Narayana, A. Hanson, and E. Learned-Miller. Coherent motion segmentation in moving camera videos using optical

flow orientations. In *2013 IEEE International Conference on Computer Vision*, pages 1577–1584, 2013.

[30] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200, 2014.

[31] T. Pollard and J. L. Mundy. Change detection in a 3-d world. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2007.

[32] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[33] Hazem Rashed, Mohamed Ramzy, Victor Vaquero, Ahmad El Sallab, Ganesh Sistu, and Senthil Yogamani. Fusemodnet: Real-time camera and lidar based moving object detection for robust low-light autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[34] P. Rodriguez and B. Wohlberg. Incremental principal component pursuit for video background modeling. *Journal of Mathematical Imaging and Vision*, 55(1):1–18, 2016.

[35] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[36] H. S. Sawhney, Y. Guo, and R. Kumar. Independent motion detection in 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1191–1199, 2000.

[37] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, pages 621–635. Springer, 2018.

[38] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1219–1225, 2009.

[39] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1778–1792, 2005.

[40] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 2, pages 246–252. IEEE, 1999.

[41] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[42] Marc Van Droogenbroeck and Olivier Paquot. Background subtraction: Experiments and improvements for vibe. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 32–37. IEEE, 2012.

[43] M. Vivet, B. Martinez, and X. Binefa. Real-time motion detection for a mobile observer using multiple kernel tracking and belief propagation. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 144–151, 2009.

[44] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar. Cdnet 2014: An expanded change detection benchmark dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 387–394, 2014.

[45] Y. Wu, X. He, and T. Q. Nguyen. Moving object detection with a freely moving camera via background motion subtraction. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(2):236–248, 2017.

[46] K. Xue, Y. Liu, J. Chen, and Q. Li. Panoramic background model for ptz camera. In *2010 3rd International Congress on Image and Signal Processing*, volume 1, pages 409–413, 2010.

[47] H. Yong, D. Meng, W. Zuo, and L. Zhang. Robust online matrix factorization for dynamic background subtraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(7):1726–1740, 2018.

[48] Y. Yu, L. Kurnianggoro, and KH. Jo. Moving object detection for a moving camera based on global motion compensation and adaptive background model. *International Journal of Control, Automation, and Systems*, 17:1866–1874, 2019.

[49] X. Zhou, C. Yang, and W. Yu. Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):597–610, 2013.

[50] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, pages 28–31. IEEE, 2004.

[51] Zoran Zivkovic and Ferdinand Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7):773–780, 2006.