Multidimensional Systems and Signal Processing (2021) 32:897–913
https://doi.org/10.1007/s11045-021-00764-1

Two-stream small-scale pedestrian detection network with feature aggregation for drone-view videos

Han Xie[1] · **Hyunchul Shin**[1]

Received: 11 December 2019 / Revised: 26 September 2020 / Accepted: 27 January 2021 /
Published online: 8 February 2021

Abstract
Detecting small-scale pedestrians in aerial images is a challenging task that can be difficult even for humans. Observing that the single image based method cannot achieve robust performance because of the poor visual cues of small instances. Considering that multiple frames may provide more information to detect such difficult case instead of only single frame, we design a novel video based pedestrian detection method with a two-stream network pipeline to fully utilize the temporal and contextual information of a video. An aggregated feature map is proposed to absorb the spatial and temporal information with the help of spatial and temporal sub-networks. To better capture motion information, a more refined flow net (SPyNet) is adopted instead of a simple flownet. In the spatial stream subnetwork, we modified the backbone network structure by increasing the feature map resolution with relatively larger receptive field to make it suitable for small-scale detection. Experimental results based on drone video datasets demonstrate that our approach improves detection accuracy in the case of small-scale instances and reduces false positive detections. By exploiting the temporal information and aggregating the feature maps, our two-stream method improves the detection performance by 8.48% in mean Average Precision (mAP) from that of the basic single stream R-FCN method, and it outperforms the state-of-the-art method by 3.09% on the Okutama Human-action dataset.

Keywords Pedestrian detection · Feature aggregation · Drone vision · Neural network · Deep learning

1 Introduction

The detection of small pedestrians in aerial images is widely utilized in many applications such as human rescue, smart drone monitoring, and video surveillance systems. Detecting pedestrians in single images (Redmon and Farhadi 2018; Ren et al. 2015; Xie et al.

✉ Hyunchul Shin
shin@hanyang.ac.kr

Han Xie
xiehan@hanyang.ac.kr

[1] Division of Electrical Engineering, Hanyang University, 55 Hanyangdeahak-ro, Ansan, Gyeonggi-do, Korea

2019; Barekatain et al. 2017) has achieved momentous progress because of the emergence of deep convolutional neural networks (CNNs). In this field, groundbreaking and rapid adoption of deep learning architectures have produced highly accurate detection methods for traditional pedestrian datasets. The state-of-the-art performance (Liu et al. 2019) on the Caltech pedestrian dataset (Dollar et al. 2011) has achieved about a 4% miss rate for the reasonable case. In another popular dataset, the INRIA pedestrian dataset (Wojek et al. 2009), a 5% miss rate was reported with the method proposed in (Lin et al. 2018). For the KITTI benchmark (Geiger et al. 2012), the accuracy of pedestrian detection is close to 90% according to the KITTI website leaderboards.

Although existing methods can make reasonably good detections for large-scale groups of pedestrians who are close to the camera, their performance suffers serious deterioration with small-scale pedestrians as in drone images because of low resolution, distortional appearances from the top view, small instance sizes, and poor visual cues. Single image detection methods usually lack robustness, especially in small object detection. The drone-view small-scale instances often present obscure appearances and blurred boundaries, thus they result in less effective feature representations for objects in aerial images. Current detectors frequently fail to effectively leverage appearance information to distinguish these objects from the surrounding background or similar objects. In addition, small instances somehow can be suddenly missed in certain frames.

There are two main limitations of single-image detectors. First, the detectors based on a single image are not robust enough because of the fluctuation of detection confidence values, since they can not incorporate temporal consistency and constraints. Second, complicated backgrounds influence detection performance to some extent. Single-image detectors are more likely to generate false positives because information in only one frame is used. However, if the context information of the whole video is exploited, these false positives can be effectively removed as demonstrated in (Kang et al. 2017).

Most of the video-based object detection methods are implemented based on the ImageNet VID dataset (Russakovsky et al. 2015) in which the object lies in the center of an image, and the scale is large enough. However, drone vision is more challenging due to the various view points and scales. Therefore, we make use of the advantages of both the single-image based object detection methods and video-based object detection methods. Among the advanced deep CNN architectures for general object detection, we follow the pipeline of R-FCN (Dai et al. 2016) because it shows superior and faster performance than the R-CNN counterpart (Girshick 2015; Ren et al. 2015) for object detection. We exploit the advantages of video-based object detection as well. Videos or sequences can provide multi-frames of images, and thus per-frame feature learning can be improved by temporal aggregation. Furthermore, motion information, such as an optical flow network (Dosovitskiy et al. 2015), can appraise the motions between frames to further enhance the features.

Inspired by these motivations, we developed a new video based, small-scale pedestrian detection method. To the extent of our knowledge, this is the first work that exploits the video based two-stream architecture for solving the small object detection problem. The main contributions are as follows:

1. A novel deep neural network architecture with two-stream subnetworks incorporates spatial and temporal information to improve detection performance for small scale instances as well as partially occluded objects.
2. Feature aggregation with nearby frames is proposed for our two-stream network. An average operator is applied to aggregate the feature maps after mapping the spatial

feature maps of the nearby frames by flow-guided warping. A more refined flow net (SPyNet), instead of a simple flownet, is adopted as the temporal subnetwork to obtain the motion information and to generate flow feature maps.

3. Some effective techniques in single-image based methods are also adopted in our spatial-stream network, including less downsampling and dilation convolution. Less downsampling, which results in relatively larger resolution can keep more detailed information of small-scale instances in the spatial stream network. In order to offset for the receptive field, dilation convolution is applied in the deep layers of the network to generate a final spatial feature map that includes richer information.
4. Our method shows state-of-the-art performance in drone view datasets, such as the Okutama human action dataset (Barekatain et al. 2017) and the VisDrone dataset (P. Zhu et al. 2018). By additionally exploiting temporal information, it improves a mAP by 3.09% more than the state-of-the-art method (Xie et al. 2019) on the Okutama Human-action dataset. To further verify the performance for general drone view object detection, we have also performed experiments with the VisDrone dataset and achieved 14.06% improvement in mean Average Precision at a 0.5 IoU threshold (mAP@0.5) when compared to the well-known SSD-PeleeNet method (Ozge Unel et al. 2019) on the VisDrone VID validation set.

The remainder of this paper is arranged in the following manners. Section 2 introduces some recent works related to both single-image based detection and video-based detection methods. Section 3 explains the proposed two-stream detection network with feature aggregation (TDFA) in detail. Experiments and results are discussed in Sect. 4. At last, Sect. 5 summarizes conclusions and future work.

2 Related works

2.1 Single-image based pedestrian detection

With the rapid growth of deep CNN technologies recently, many general pedestrian detectors have achieved good performance. For small pedestrian detection, a common and popular strategy is the multi-layer approach, which generates multi-branches or subnetworks for different scale training. The MS-CNN (Cai et al. 2016) is performed with output of multi-layers to detect pedestrians of various scales. Similarly, SAF R-CNN (Li et al. 2017) proposes a divide-and-conquer approach with Fast R-CNN pipeline. This strategy detects pedestrians by two built-in subnetworks at diverse scales from disjoint ranges. Another way to enhance the feature presentation is by incorporating both the rich semantic information from deeper layer features and the fine-grained information from shallow layer feature maps. In order to extract strong semantics representation at all scales, including small scales, lateral connections with a top-down pathway have been proposed in Feature Pyramid Network (FPN) (Dollár et al. 2014). YOLO-v3 (Redmon and Farhadi 2018) uses a similar method, but replaces the nearest neighbor upsampling in deconvolution to achieve better performance of small-size object detection.

2.2 Video-based object detection

Since ImageNet proposed the challenge for video-based object detection (VID) and provided the dataset, there have been many various works that have focused on the video object detection. One of the typical architectures is ConvNets + LSTM, which extracts features individually on each frame, then pools the predictions through the entire video. For example, ROLO (Ning et al. 2017) develops a recurrent convolutional neural network (RCNN) with spatially supervised for the task of visual object tracking. It concatenates high-level spatial features captured by convolutional networks with regional information and executes Long Short Term Memory (LSTM) in the temporal domain. This type of ConvNets + LSTM approach can deliver high-level semantic information but is not able to obtain fine low-level detailed information, which is important for small-scale detection. It is also time-consuming for training due to the network unrolling with multiple frames for backpropagation across time. The second typical architecture is 3D ConvNets (Varol et al. 2017), which directly creates hierarchical representations of spatio-temporal features. However, the 3D ConvNets models take many more parameters than those of 2D ConvNets, owing to the extra kernel dimension, which makes the training more difficult. The third typical architecture is the two-stream network structure put forward by Simonyan and Zisserman (Simonyan and Zisserman 2014). To capture spatio-temporal information about the appearance as well as the movement of objects, both the RGB and optical flow frames are given into deep ConvNets architectures separately, and finally their softmax scores are joined with late fusion. An extended work (Feichtenhofer et al. 2016) combines the spatial and flow branches at the last convolutional layer of the network. In more recent works such as FGFA (Zhu et al. 2017) and MANet (Wang et al. 2018a, b), the features of a single frame are enhanced by utilizing an optical flow network to measure the motions between the reference frame along with the nearby frames and by using a more advanced deep learning framework. This framework investigates temporal information based on the feature level, rather than the final box level, as ConvNets + LSTM does. Compared with two other typical architectures, this type of two-stream network shows better performance and requires relatively less time for training and testing.

2.3 Video-based pedestrian detection

In TLL (Song et al. 2018), somatic topological line localization (TLL) is integrated with a temporal feature aggregation that utilizes a joint Conv-LSTM model for detecting multi-scale pedestrians. ADM (Zhang et al. 2018) introduces a RCNN based localization policy that uses the sequences of coordinate transformation actions to get the final detection of the pedestrian instances. In (Wang et al. 2018a, b), a part and context network (PCN) is proposed by incorporating a primary branch, a context branch, and a part branch into an integrated architecture with an LSTM module for communicating the body part semantic information.

In this research, we also adopt a similar two-stream network architecture to incorporate temporal as well as spatial information for better accuracy and effectiveness. To effectively capture the features of small-scale instances, we design a network architecture based on the R-FCN pipeline, and a variant ResNet is used as the backbone network in the spatial stream. Furthermore, we aggregate the features of the nearby frames to acquire more temporal information. Our method achieves more robust detection performance in difficult

Table 1 Notations

i	Reference frame index
k	Nearby frame index
I	Video frame
f, wf, af	Spatial feature map, warped feature map and aggregated feature map
p, q	2D location
$\mathrm{M}_{i\to i+k}$	2D flow field
$\mathcal{F}$	Functions of flow estimation
W, G	Bi-linear interpolation function and the bilinear interpolation kernel

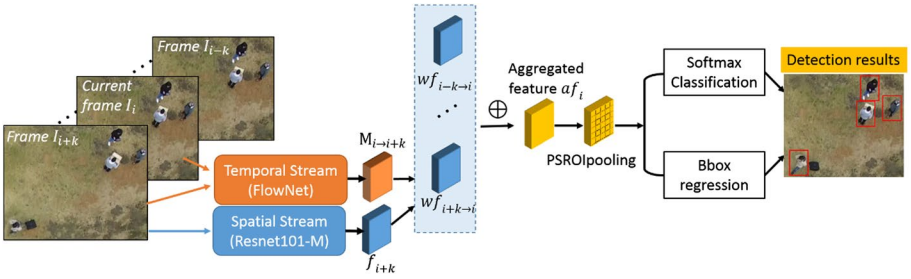

Fig. 1 Architecture of our proposed two-stream detection network with feature aggregation (TDFA)

examples such as the detection of small-scale instances from drone-view images and can effectively reduce false positive detections.

3 Two-stream detection network with feature aggregation

In this section, we demonstrate the details of our proposed method. The main notations adopted in this paper are declared in Table 1.

3.1 Two-stream network design

Our architecture consists of two subnetworks. For one subnetwork, a variant ResNet is utilized to extract the spatial feature map f_i on frame I_i, and the other is the temporal stream network. Given a reference frame I_i and a nearby frame I_{i+k}, a two-dimensional flow filed $\mathrm{M}_{i\to i+k} = \mathcal{F}(I_i, I_{i+k})$ is obtained by the optical flow estimation algorithm (Ranjan and Black 2017). $\mathcal{F}(I_i, I_{i+k})$ denotes the flow field estimated from frame I_i to I_{i+k}. Figure 1 shows a flowchart of the two-stream detection network with feature aggregation. By using a sequence of images, a series of spatial feature maps $f_{i-k}, \ldots, f_{i+k}$ are generated by passing the frame $I_{i-k}, \ldots, I_{i+k}$ through the spatial stream, and then the temporal feature map $\mathrm{M}_{i\to i+k}$ is calculated by applying the FlowNet with frame I_i and frame I_{i+k}. The feature warping is used to generate the warped feature $wf_{i+k\to i}$. Similarly, the warped feature map $wf_{i-k\to i}$ to $wf_{i+k\to i}$ is generated. These features are then aggregated as af_i and delivered to the PS ROI

pooling layer. At last, we can get the final detection results by detection module with softmax classification and bounding box regression.

3.2 Feature aggregation

As motivated by (Zhu et al. 2017) and (Wang et al. 2018a, b), we adopt flow-guided feature warping to capitalize on the temporal information. In the spatial stream, the spatial network is applied to the nearby $I_{i-k}, \ldots, I_{i+k}$ frames to get the corresponding feature maps $f_{i-k}, \ldots, f_{i+k}$. Then, the feature map on the nearby frame I_{i+k} is warped to the current frame I_i as follows:

$$wf_{i+k\to i} = \mathrm{W}\left(f_{i+k}, \mathrm{M}_{i\to i+k}\right) = \mathrm{W}\left(f_{i+k}, \mathcal{F}\left(I_i, I_{i+k}\right)\right) \tag{1}$$

where $f_{i+k\to i}$ are the warped features that denote the feature map from frame I_{i+k} to frame I_i. $W(\cdot)$ denotes the bi-linear interpolation function, it applied to each location for all the feature maps. In the reference frame i, a location p maps to the location $p + \Delta p$ in frame I_{i+k}, as presented in the Eqs. (2) and (3):

$$\Delta p = \mathcal{F}\left(I_i, I_{i+k}\right)(p) \tag{2}$$

$$wf_{i+k\to i}(p) = \sum_q G(q, p + \Delta p) f_{i+k}(q) \tag{3}$$

where q stands for all spatial locations in the feature maps $f_{\mathrm{i+k}}$, Δp is the output of the flow estimation at location p, and G(·) is the bilinear interpolation kernel as

$$G(q, p + \Delta p) = \max\left(0, 1 - \|q - (p + \Delta p)\|\right). \tag{4}$$

When the warped features of nearby frames $wf_{i-k\to i}, \ldots, wf_{i+k\to i}$ are obtained, the feature map of the reference frame can be enhanced by accumulating the multiple feature maps of nearby frames, which provides the temporal information of the object instances. We aggregate the feature maps by averaging them. The aggregated feature af_i at the reference frame i is generated as

$$af_i = \frac{\sum_{t=i-k}^{i+k} wf_{t\to i}}{2k+1} \tag{5}$$

The procedure of generating the aggregated feature is presented in Algorithm 1.

Algorithm 1 The pseudocode of generating the aggregated features from the video.

Input: video frames $\{I_i\}$, aggregation number of frames K

Output: aggregated features

1: **for** $k = 1$ **to** $K + 1$ **do** ▷ feature buffer initialization

2: $f_{\mathrm{k}} = N_{feat}(I_k)$

3: **end for**

4: **for** $i = 1$ **to** ∞ **do** ▷ the reference frame

5: **for** $j = max(1, i - K)$ **to** $i + K$ **do** ▷ nearby frames

6: $wf_{i+k\to i} = \mathrm{W}(f_{\mathrm{i+k}}, \mathcal{F}(I_i, I_{i+k}))$ ▷ flow-guided warp

7: **end for**

8: $af_i = \frac{\sum_{t=i-k}^{i+k} wf_{t\to i}}{2k+1}$ ▷ aggregate features

9: $f_{\mathrm{i+k+1}} = N_{feat}(I_{i+k+1})$ ▷ feature buffer update

10: **end for**

3.3 Flow network of the temporal stream

Instead of using a simple version of FlowNet (Dosovitskiy et al. 2015), we adopted the Spatial Pyramid Network (SPyNet) (Ranjan and Black 2017) which captures residual flow based on a coarse-to-fine spatial pyramid structure. In our case, the motions of the object instances between the frames are small. SPyNet is better able to deal with a more detailed and precise motion optical flow. Furthermore, as proposed in (Ranjan and Black 2017), the SPyNet model is faster and smaller than FlowNet. To further reduce the computation time, we apply the flow network on non-adjacent frame pairs as in (Zhu et al. 2017). By compositing the intermediate flow fields, the flow field between the non-adjacent frames can be measured. As a result, the computation time can be reduced in half with almost the same accuracy.

In SPyNet, the residual flow is computed by convolution at the high level of the pyramid with the low-resolution feature map. At each pyramid level, the residual flow is computed and successively propagates to the next lower levels with higher resolution for every pyramid level. Eventually, the flow is captured at the lowest levels of the pyramid. These types of procedures can be treated as a flow-block, which is illuminated in Fig. 2. We adopt a 5-level SPyNet, the flow chart of the SpyNet architecture is also illuminated in Fig. 2.

3.4 Feature network of the spatial stream

Following R-FCN (Dai et al. 2016), we adopt a variant ResNet-101 (ResNet101-M) as the backbone network for spatial feature extraction. Compared with the original ResNet-101 network (He et al. 2016), the ending average pooling and the fully convolutional (fc) layer have been cut out for the object detection task. The proposed variant ResNet-101 is specifically designed for feature extraction of small objects. At the last block in conv5 stage, the stride of the convolution layer with 2 is modified to 1 in order to keep the relatively large

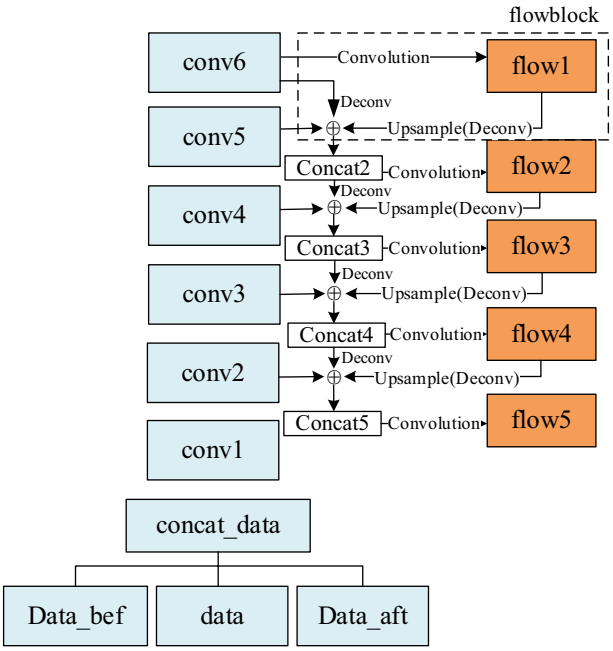

Fig. 2 Flowchart of SPyNet architecture

spatial resolution of the feature map. To further increase the feature resolution, the effective stride of the last block in the conv5 stage is changed from 32 to 16 pixels. Then, the dilation convolution is used to offset the size of receptive field, the kernel size of the dilation convolutional layers larger than 1 is set as 2 in the last block of the conv5 stage.

4 Experimental results

4.1 Implementation details

Our experiments use the ResNet-101-M model pre-trained on the ILSVRC-CLS image classification dataset (Russakovsky et al. 2015), and the base SPyNet model pre-trained on the Flying Chairs dataset (Dosovitskiy et al. 2015). To augment the training data, image flipping is adopted. We use single scale images with 720×1280 pixels in training to avoid GPU memory overflow. To fine-tune the detection bounding boxes and to choose hard examples automatically in training, non-maximum suppression (NMS) and online hard example mining (OHEM) (Shrivastava et al. 2016) were adopted. We chose Mxnet as the platform and trained the network on four parallel Nvidia GeForce GTX TITAN X GPUs with 12 GB of memory, while testing was performed on a single GPU.

4.2 Okutama human-action dataset

The Okutama Human-action Dataset is a real-word aerial view video dataset with high image resolution. A total of 43 video sequences are captured at 30 FPS, including 33

training sequences and 10 testing sequences. The videos were recorded with UAVs flying varying 10–45 m altitudes. The camera angle is between 45 and 90 degrees. The dataset can be used for both human detection and action understanding. In this paper, we aim at the human detection task. The dataset is spatio-temporal fully-annotated. Each instance has a tracking id. In experiments, we used 54,503 images for training and 14,114 images for testing.

Following the object detection protocols in (Everingham et al. 2010; Barekatain et al. 2017), the mean Average Precision at a 0.5 IoU threshold (mAP@0.5) is used as the evaluation metric. The Intersection Over Union (IOU) considers the overlap of areas between the prediction bounding box and the ground truth bounding box, which is calculated by the Eq. (6):

$$IOU = (area\ of\ intersection)/(area\ of\ union). \tag{6}$$

If $IOU \geq 0.5$, the detection is classified as a true positive. Otherwise, the detection is false positive. Precision is the fraction of positive instances among the detected instances. Recall is the proportion of instances that are correctly detected among the ground truth. The Average Precision (AP) is computed by averaging the precision over a set of evenly spaced recall levels [0, 0.1, … 1.0]. The definitions of precision, recall, and AP, in terms of true positive (TP), false positive (FP), and false negative (FN), are as follows:

$$Precision = TP/(TP + FP) \tag{7}$$

$$Recall = TP/(TP + FN) \tag{8}$$

$$AP = \frac{1}{11} \sum_{r \in [0,0.1,\ldots,1]} p_{interp}(r) \tag{9}$$

where $p_{interp}(r) = \max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r})$, and $p(\tilde{r})$ is the measured precision at recall $\tilde{r}$. The precision-recall curves were drawn using the precision $p(r)$ as a function of recall r. The area under the curve was summarized to obtain Average Precision. The mean Average Precision (mAP) for a set of classes is the mean of the Average Precision (AP) scores for each class, which is computed as:

$$\text{mAP} = \frac{\sum_{n=1}^{N} AP(n)}{N} \tag{10}$$

where N is the number of classes. In our task, we only consider the class “pedestrian”. Therefore N = 1, the mAP equals to AP.

4.2.1 Ablation experiments of aggregation number of frames in training and inference

In Table 2, a comparison of the performance and runtime for the utilization of a diverse number of frames is given. The case of $k = 0$ is the single image detection based on our network architecture without multi-frame input. The parameter k is the number of additional nearby frames that we used for feature aggregation. One can observe that the performance improves with increasing additional input frames up to a certain level. When a greater number of frames are aggregated, the runtime gradually increases. Notice that

Table 2 Comparison of the results by using a different number of frames as input

Frame num (k)	10	9	8	4	2	1	0
mAP@0.5	87.03%	87.18%	86.84%	83.65%	82.05%	81.25%	78.70%
Runtime (s/frame)	0.2955	0.2831	0.2746	0.2576	0.2554	0.2480	0.0862

the performance has a bit of a decrease with aggregating 10 frames in this example. This is probably because the information of an image 10 frames away is not very "useful" for the current frame image. Our model can reach the best performance of 87.18% mAP when $k=9$.

4.2.2 Ablation experiments

We evaluated the effect of each component of our proposed approach. In the spatial stream, we compared the performance of original ResNet-101 and ResNet-101-M, with R-FCN as the basic architecture. In the temporal stream, we compared the performance of FlowNet and SpyNet for capturing temporal information, with the number of input frames $k=9$. As shown in Table 3, the ResNet-101-M performs better than the original ResNet by 0.75% in mAP, to extract the features of the spatial domain. A significant improvement of 5.94% in mAP has been achieved after using the temporal steam with FlowNet. Then the FlowNet is replaced with the SpyNet to capture a more detailed and precise motion optical flow, resulting an additional improvement of 2.54% in mAP.

4.2.3 Comparison with the state of the arts on Okutama dataset

Table 4 gives a comparison of the detection results and runtime between our method and the state-of-the-art methods on the Okutama dataset. The R-FCN with ResNet-101-M is our base network of the spatial stream, which is based on single-image based detection with $k = 0$. By exploiting the temporal information and aggregating the feature maps of nearby frames, our two-stream method finally improves the performance from 78.7 to 87.18% in mAP. When compared with the existing best method DIF R-CNN (Xie et al. 2019), our method outperforms by 3.09%. In DIF R-CNN (Xie et al. 2019), which was our previous work, 90.3% mAP was reported on the Okutama validation set. However, the test set in (Xie et al. 2019) is different from the Okutama official test set, since (Xie et al. 2019) was published before the official data was released. DIF R-CNN (Xie et al. 2019) achieved

Table 3 Influence of each component of our proposed method on the Okutama test dataset

Spatial stream ($k=0$)		Temporal stream ($k=9$)		Performance (mAP@0.5) (%)
R-FCN + ResNet-101	R-FCN + ResNet-101-M	FlowNet	SpyNet	
√				77.95
	√			78.70
	√	√		84.64
	√		√	87.18

Table 4 Performance comparison with other existing methods on the Okutama test dataset

Methods	mAP@0.5 (%)	Runtime (s/frame)
SSD-Okutama (Barekatain et al. 2017) (baseline)	72.30	0.028
R-FCN with ResNet101-M (ours without multi-frame)	78.70	0.08
DIF R-CNN (Xie et al. 2019)	84.09	0.22
TDFA with ResNet101-M ($k=9$) (Ours)	87.18	0.28

84.09% of mAP for the official test set. All the results presented in Table 4 are by using the official test set. The proposed approach takes about 50–52 h to train the best model with $k=9$ on the Okutama dataset, and the runtime is 0.28 s/f for testing with the original image size of 3480×2160 pixels. Figure 3 indicates the precision-recall curve of our proposed TDFA and other existing methods on the Okutama human-action dataset.

Figure 4 is a visualization comparison of the detection results. Our proposed method generates less false positive detections and is better to handle the partially occluded cases than SSD-Okutama (Barekatain et al. 2017) and DIF R-CNN (Xie et al. 2019). Figure 5 compares the detection results for a sequence by using the test set *Drone 2-Noon*-1.2.1. For this example, we choose the frame $id = \{1100, 1105, 1110, 1115\}$. The images are displayed with partial magnification. Compared with these single-image based methods (Xie et al. 2019; Barekatain et al. 2017), our approach is more robust to detect the persons in every frame without sudden missing.

4.3 VisDrone dataset

To further verify the performance of our proposed method in general small-scale object detection with drone videos, we also use the Visdrone dataset (P. Zhu et al. 2018) for our experiment. The VisDrone dataset focus on advancing visual understanding tasks such as object detection and tracking for the drone applications. It was collected by drone-mounted camera with various aspects including diverse location, environment (urban and country),

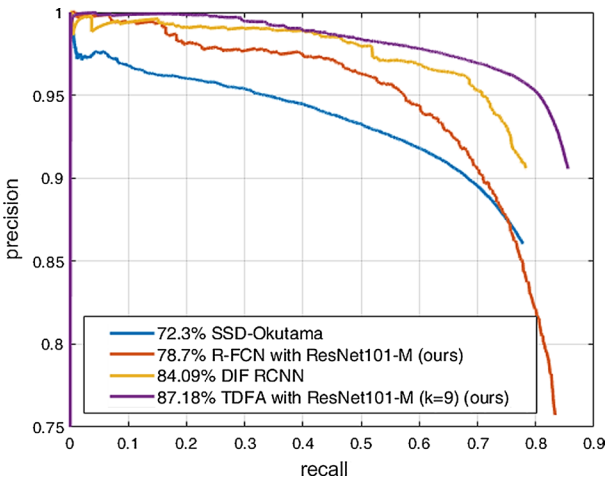

Fig. 3 The pedestrian detection comparison of our proposed TDFA and other methods on the Okutama human-action dataset

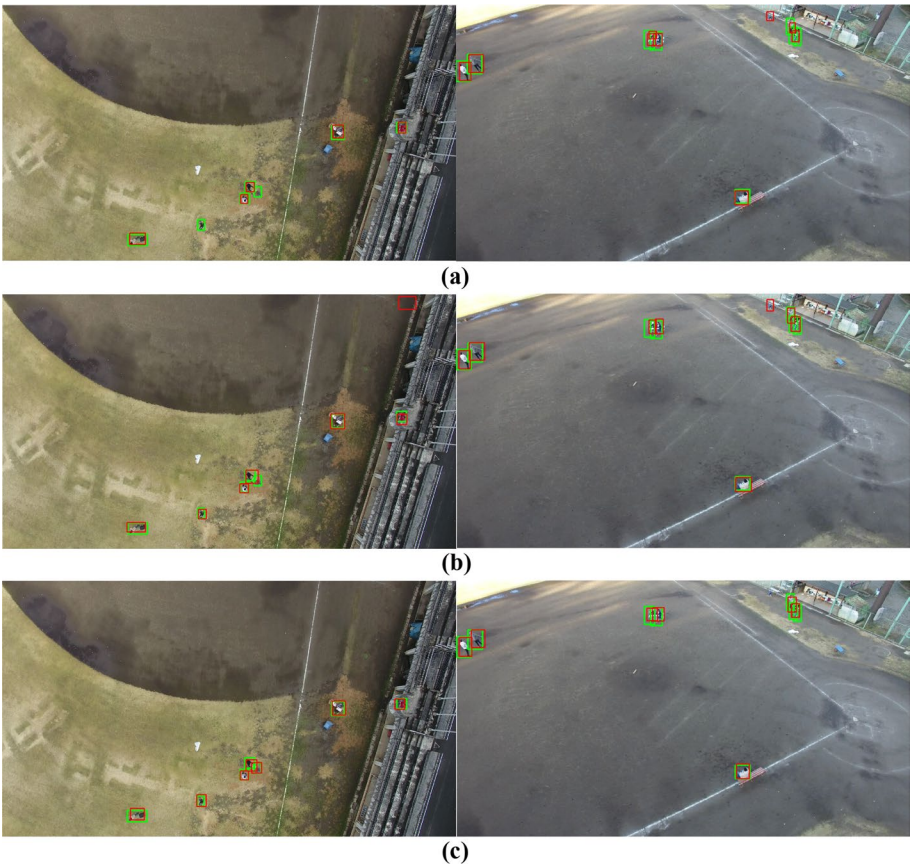

Fig. 4 Visual comparison of the detection results. Top to bottom: **a** SSD-Okutama (Barekatain et al. 2017) (baseline method) showing two false negatives in the left case and two false positives in the right case. **b** DIF R-CNN (Xie et al. 2019) showing one false negative and one false positive in the left case, one false positive in the right case. **c** our results on the Okutama test set detected all persons without false positives. We denote ground truth in green and detection results in red (Color figure online)

density (crowded and sparse scenes), and 10 classes of objects (people, pedestrian, bicycle, bus, car, van, truck, motor, tricycle, awning-tricycle). Following the comparison method, SSD-PeleeNet (Ozge Unel et al. 2019), 10 classes were grouped into two main groups as pedestrian and vehicle. Training and validation were conducted only on the VisDrone-VID training set which contains 56 video clips with 24,201 frames. The performance metric was calculated on the VisDrone-VID validation set involving seven video clips with 2819 images, in which the image sizes of video sequences are not uniform. We trained the network based on variant ResNet (ResNet-101-M) and choose the best training model with 9 frames ($k=9$) to get the final results.

The performance comparison of our detector with SSD-PeleeNet (Ozge Unel et al. 2019) is presented in Table 5, which presents that our method achieves significantly better performance in most cases. It produces a mAP@0.5 of 50.73% for overall classes which is a big improvement (14.06%), from 36.67 to 50.73%. The runtime is 0.26 s/f on a single TITAN X GPU for testing. It takes about 12 h to train the best model with $k=9$

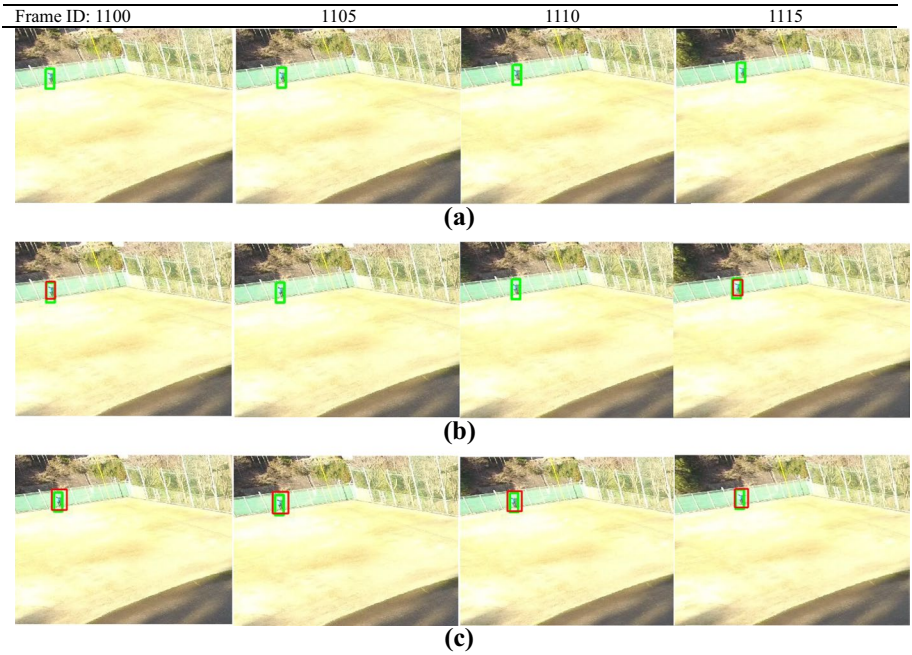

Fig. 5 Comparison of sequence detection results. Top to bottom: **a** SSD-Okutama (Barekatain et al. 2017) (baseline method) with four errors, **b** DIF R-CNN (Xie et al. 2019) with two errors, and **c** our results on the Okutama test set without error. We denote ground truth in green and detection results in red (Color figure online)

Table 5 Comparison of our method TDFA with SSD-PeleeNet on the VisDrone validation set

Methods	Avg. Precision (%), Iou:								
	Vehicle			Pedestrian			Overall (mAP)		
	0.5	0.75	0.5:1.0	0.5	0.75	0.5:1.0	0.5	0.75	0.5:0.95
SSD (Vino)	32.27	17.29	17.21	24.74	3.64	8.75	28.50	10.46	12.98
SSD (Pelee T5×3 I5×3)	41.39	19.55	21.07	**30.26**	2.94	9.61	35.82	11.24	15.34
SSD (Pelee38 T5×3 I5×3)	44.35	22.64	23.53	28.99	3.25	9.69	36.67	12.95	16.61
TDFA (Resnet101-M) (Ours)	**55.74**	**33.45**	**28.44**	28.77	**8.77**	**10.84**	**50.73**	**27.94**	**27.27**

Bold values indicate the best results

on the VisDrone dataset. For vehicle detection, ours performs 11.12% better than the SSD-PeleeNet (Ozge Unel et al. 2019) in mAP@0.5. Our approach also improves the performance with different degrees in mAP (IoU:0.75 and 0.5:1.0) on both vehicle and pedestrian classes. The detection accuracy has a slight of decrease (1.49%) in pedestrian detection with mAP @0.5. In addition to the small-sized instance problem, this dataset contains night time detections without IR-aid, such as infrared images. Furthermore, heavy occlusion, complicated backgrounds, fog, and bad illumination make the detection task even more challenging. The detection examples of our approach are shown in

Fig. 6 Visualization of our detection results on the VisDrone validation set. We denote ground truth in green and detection results in red (Color figure online)

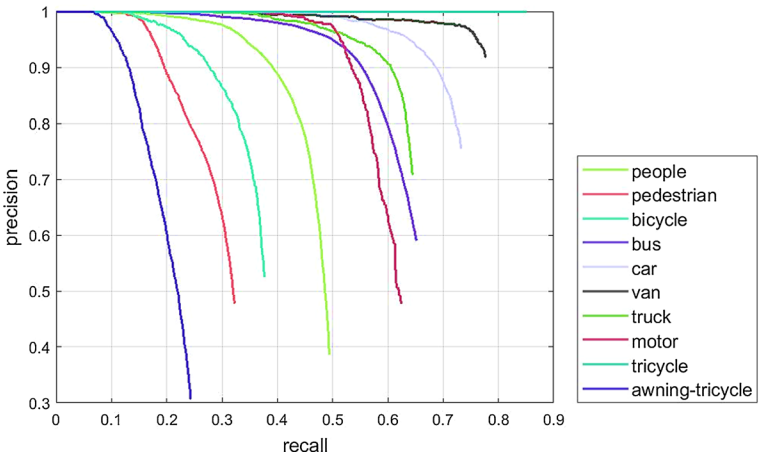

Fig. 7 The precision-recall curve of our proposed TDFA for each class of the Visdrone dataset

Fig. 6. The precision-recall curve of our proposed TDFA for each class of the Visdrone dataset is presented in Fig. 7.

5 Conclusion and future work

In this paper, a novel two-stream detection network with feature aggregation (TDFA) is proposed for small-scale pedestrian detection in drone-view videos. To make a more robust detection performance on drone-view videos, we introduce two-stream video-based detection techniques with the R-FCN pipeline. We follow the traditional single-image based feature map extraction method in the spatial stream. Additionally, we apply SPyNet to extract flow feature maps to catch the tiny motion and incorporate the temporal information. Then, the mapping and warping operations are performed from the flow features to the spatial features. Finally, the feature maps of nearby frames are aggregated. The aggregated feature can give a more effective feature representation with spatio-temporal information.

Experimental evaluations demonstrated that the proposed TDFA is superior when compared to other single-image based detection method, in detecting small-scale pedestrian instances. The performance of our results is 3.09% better in mAP than that of the state-of-the-art results on the Okutama Human-action Dataset. Furthermore, it also achieves good performance on general drone-view object detection tasks, such as the VisDrone VID task. Our method achieved a mAP@0.5 of 50.73% with 14.06% improvement than the SSD-PeleeNet on the VisDrone VID validation set. In the future, we prone to focus on developing an algorithm for handling occlusion and bad illumination cases. By solving these two main challenging cases, we are able to further enhance the overall detection performance.

Acknowledgements This material is based on work supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under the Industrial Technology Innovation Program (10080619).

References

Barekatain, M., Marti, M., Shih, H. -F., Murray, S., Nakayama, K., Matsuo, Y., et al. (2017). Okutama-action: an aerial view video dataset for concurrent human action detection. In *30th IEEE conference on computer vision and pattern recognition workshops* (Vol. 2017, pp. 2153–2160). IEEE Computer Society. https://doi.org/10.1109/CVPRW.2017.267.

Cai, Z., Fan, Q., Feris, R. S., & Vasconcelos, N. (2016). A unified multi-scale deep convolutional neural network for fast object detection. In *European conference on computer vision* (pp. 354–370). Springer. https://doi.org/10.1007/978-3-319-46493-0_22.

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems* (pp. 379–387). http://papers.nips.cc/paper/6465-r-fcn-object-detection-via-region-based-fully-convolutional-networks.pdf.

Dollár, P., Appel, R., Belongie, S., & Perona, P. (2014). Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*(8), 1532–1545. https://doi.org/10.1109/TPAMI.2014.2300479.

Dollar, P., Wojek, C., Schiele, B., & Perona, P. (2011). Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(4), 743–761. https://doi.org/10.1109/TPAMI.2011.155.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., et al. (2015) Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2758–2766). https://doi.org/10.1109/ICCV.2015.316.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision, 88*(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4.

Feichtenhofer, C., Pinz, A., & Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1933–1941). https://doi.org/10.1109/CVPR.2016.213.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 3354–3361). IEEE. https://doi.org/10.1109/CVPR.2012.6248074.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448). https://doi.org/10.1109/ICCV.2015.169.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778). https://doi.org/10.1109/CVPR.2016.90.

Kang, K., Li, H., Yan, J., Zeng, X., Yang, B., Xiao, T., et al. (2017). T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology, 28*(10), 2896–2907. https://doi.org/10.1109/TCSVT.2017.2736553.

Li, J., Liang, X., Shen, S., Xu, T., Feng, J., & Yan, S. (2017). Scale-aware fast R-CNN for pedestrian detection. *IEEE Transactions on Multimedia, 20*(4), 985–996. https://doi.org/10.1109/TMM.2017.2759508.

Lin, C., Lu, J., Wang, G., & Zhou, J. (2018). Graininess-aware deep feature learning for pedestrian detection. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 732–747). https://doi.org/10.1007/978-3-030-01240-3_45.

Liu, W., Liao, S., Ren, W., Hu, W., & Yu, Y. (2019) High-level semantic feature detection: A new perspective for pedestrian detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5187–5196). https://arxiv.org/abs/1904.02948.

Ning, G., Zhang, Z., Huang, C., Ren, X., Wang, H., Cai, C., et al. (2017) Spatially supervised recurrent convolutional neural networks for visual object tracking. In *2017 IEEE international symposium on circuits and systems (ISCAS)* (pp. 1–4). IEEE. https://doi.org/10.1109/ISCAS.2017.8050867.

Ozge Unel, F., Ozkalayci, B. O., & Cigla, C. (2019). The power of tiling for small object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. http://openaccess.thecvf.com/content_CVPRW_2019/html/UAVision/Unel_The_Power_of_Tiling_for_Small_Object_Detection_CVPRW_2019_paper.html.

Ranjan, A., & Black, M. J. (2017). Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4161–4170). https://doi.org/10.1109/CVPR.2017.291.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint: http://arxiv.org/abs/1804.02767.

Ren, S., He, K., Girshick, R., & Sun, J. (2015) Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91–99). https://doi.org/10.1109/TPAMI.2016.2577031.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision, 115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y.

Shrivastava, A., Gupta, A., & Girshick, R. (2016) Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 761–769). https://doi.org/10.1109/CVPR.2016.89.

Simonyan, K., & Zisserman, A. (2014) Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems* (pp. 568–576). https://arxiv.org/abs/1406.2199.

Song, T., Sun, L., Xie, D., Sun, H., & Pu, S. (2018). Small-scale pedestrian detection based on somatic topology localization and temporal feature aggregation. arXiv preprint, https://arxiv.org/abs/1807.01438.

Varol, G., Laptev, I., & Schmid, C. (2017). Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(6), 1510–1517. https://doi.org/10.1109/TPAMI.2017.2712608.

Wang, S., Cheng, J., Liu, H., & Tang, M. (2018). Pcn: Part and context information for pedestrian detection with cnns. arXiv preprint, https://arxiv.org/abs/1804.04483.

Wang, S., Zhou, Y., Yan, J., & Deng, Z. (2018) Fully motion-aware network for video object detection. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 542–557). https://doi.org/10.1007/978-3-030-01261-8_33.

Wojek, C., Walk, S., & Schiele, B. (2009) Multi-cue onboard pedestrian detection. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 794–801). IEEE. https://doi.org/10.1109/CVPR.2009.5206638.

Xie, H., Chen, Y., & Shin, H. (2019). Context-aware pedestrian detection especially for small-sized instances with Deconvolution Integrated Faster RCNN (DIF R-CNN). *Applied Intelligence, 49*(3), 1200–1211. https://doi.org/10.1007/s10489-018-1326-8.

Zhang, X., Cheng, L., Li, B., & Hu, H.-M. (2018). Too far to see? Not really!—Pedestrian detection with scale-aware localization policy. *IEEE Transactions on Image Processing, 27*(8), 3703–3715. https://doi.org/10.1109/TIP.2018.2818018.

Zhu, P., Wen, L., Bian, X., Ling, H., & Hu, Q. J. (2018). Vision meets drones: A challenge. arXiv preprint, https://arxiv.org/abs/1804.07437.

Zhu, X., Wang, Y., Dai, J., Yuan, L., & Wei, Y. (2017) Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 408–417). https://doi.org/10.1109/ICCV.2017.52.