Graph Regularized Flow Attention Network for Video Animal Counting From Drones

Pengfei Zhu, Tao Peng, Dawei Du, Hongtao Yu, Libo Zhang, and Qinghua Hu, *Senior Member, IEEE*

***Abstract*— In this paper, we propose a large-scale video based animal counting dataset collected by drones (AnimalDrone) for agriculture and wildlife protection. The dataset consists of two subsets, *i.e.*, PartA captured on site by drones and PartB collected from the Internet, with rich annotations of more than 4 million objects in 53, 644 frames and corresponding attributes in terms of density, altitude and view. Moreover, we develop a new graph regularized flow attention network (GFAN) to perform density map estimation in dense crowds of video clips with arbitrary crowd density, perspective, and flight altitude. Specifically, our GFAN method leverages optical flow to warp the multi-scale feature maps in sequential frames to exploit the temporal relations, and then combines the enhanced features to predict the density maps. Moreover, we introduce the multi-granularity loss function including pixel-wise density loss and region-wise count loss to enforce the network to concentrate on discriminative features for different scales of objects. Meanwhile, the graph regularizer is imposed on the density maps of multiple consecutive frames to maintain temporal coherency. Extensive experiments are conducted to demonstrate the effectiveness of the proposed method, compared with several state-of-the-art counting algorithms. The AnimalDrone dataset is available at https://github.com/VisDrone/AnimalDrone.**

***Index Terms*— Animal counting, drone, graph regularized flow attention network, multi-granularity loss.**

I. Introduction

THE world of artificial intelligence (AI) is quickly on the rise as it has already been used in many different industries, from manufacturing to automotive industry. It is also interesting to see that AI meets agriculture and wildlife protection. For example, we can use drones equipped with cameras to detect diseases, identify crop readiness, and monitor animals. The specific view by drones can avoid the problem of mutual occlusion between individuals of high density population in flat view observation. Therefore, drones are very suitable for counting animals.

Despite much progress achieved in recent years, counting animals captured by drones remains challenging due to several factors such as motion blur, scale variation, sparse positive samples and tiny objects. Development of drone-based animal counting algorithms still lacks publicly available large-scale benchmarks and datasets. Although there exist several animal counting datasets for bats [1], penguins [2] and elephants [3], they are still limited in data volume, animal species and covered scenarios.

To advance state-of-the-art animal counting algorithms, we collect a large-scale high-resolution drone-based animal counting dataset named AnimalDrone (see Fig. 1), which consists of two subsets, *i.e.*, AnimalDrone-PartA and AnimalDrone-PartB. Specifically, videos in AnimalDrone-PartA (PartA for short) are shot on site by drones, while AnimalDrone-PartB (PartB for short) is collected from Internet. PartA contains 59 video sequences with 18, 940 fully annotated frames and PartB is formed by 103 videos sequences with 34, 704 annotated frames. In summary, AnimalDrone contains 53, 644 frames with over 4 million object annotations in diverse scenes. There are 10 kinds of animals in the AnimalDrone dataset, *e.g.*, *sheep*, *horse*, *wolf* and *yak*.

Moreover, we propose a Graph regularized Flow Attention Network (GFAN) to solve the animal counting task on drones. Compared with existing works [4], [5] using pre-computed optical flows, we use the warping loss to enforce the optical flow network [6] trainable together with counting network, resulting in discriminative features for counting. Then, we develop the multi-granularity scheme to generate discriminative features for different scales. Meanwhile, we employ the graph regularizer to maintain the temporal continuity across multiple frames in the video clip. In addition, we apply the attention module [7] on the aggregated feature maps gradually to enforce the network to exploit enhanced features for better performance. The whole network is trained in an end-to-end manner with the multi-task loss, formed by four terms, *i.e.*, the MSE loss and multi-granularity loss for for multi-scale density estimation, the warping loss and graph regularizer loss for temporal consistency. Extensive experiments are carried out on our AnimalDrone dataset and two publicly avaliable video based datasets (*i.e.*, FDST [8] and UCSD [9]). For example, our GFAN method achieves 2.2 and 4.9 lower MAE score on PartA and PartB compared with the second best

Manuscript received July 9, 2020; revised January 21, 2021 and May 10, 2021; accepted May 11, 2021. Date of publication May 28, 2021; date of current version June 4, 2021. This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2019B010153002; in part by the National Key Research and Development Program of China under Grant 2018AAA0102402 and Grant 2017YFB0801900; in part by the National Natural Science Foundation of China under Grant 61732011, Grant 61876127, Grant 61925602, and Grant 61876088; in part by the Natural Science Foundation of Tianjin under Grant 17JCZDJC30800; and in part by the Applied Basic Research Program of Qinghai under Grant 2019-ZJ-7017. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Senem Velipasalar. *(Corresponding author: Dawei Du.)*

Pengfei Zhu, Tao Peng, Hongtao Yu, and Qinghua Hu are with the College of Intelligence and Computing, Tianjin University, Tianjin 300403, China (e-mail: zhupengfei@tju.edu.cn; wspt@tju.edu.cn; hongtaoyu@tju.edu.cn; huqinghua@tju.edu.cn).

Dawei Du is with the Computer Science Department, University at Albany, State University of New York, Albany, NY 12222 USA (e-mail: cvdaviddo@gmail.com).

Libo Zhang is with the Institute of Software Chinese Academy of Sciences, Beijing 100190, China (e-mail: libo@iscas.ac.cn).

Digital Object Identifier 10.1109/TIP.2021.3082297

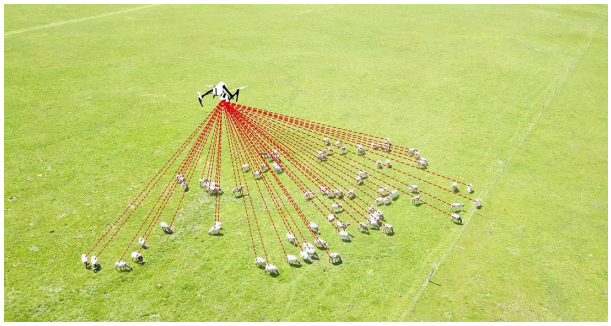


Fig. 1. Illustration of data collection by drones.

counting method, respectively. The main contributions of this paper are summarized as follows.

- We collect a large-scale drone-based video animal counting dataset, *i.e.*, AnimalDrone, which consists of 53, 644 frames with more than 4 million object annotations. **To our knowledge, AnimalDrone is the largest drone-based video animal counting dataset to date.**
- We propose the graph regularized flow attention network to deal with animal counting, which applies warping loss and multi-granularity loss to train the network and uses the attention module on the aggregated feature maps to enforce the network to focus on discriminative features for different scales of objects.
- Comprehensive experiments are conducted on the AnimalDrone dataset comparing with several state-of-the-art counting methods to demonstrate the effectiveness of our GFAN in animal counting.

II. Related Work

In this section, we briefly review several related counting datasets and state-of-the-art counting algorithms.

A. Existing Counting Datasets

1) Image Based Datasets: To date, there are some crowd counting datasets based on still images [10]–[13]. Zhang *et al.* [10] develop the Shanghaitech dataset, which includes 1, 198 images and 330, 165 annotated people in total. Hsieh *et al.* [12] present a drone-based vehicle counting dataset, which approximately contains 90, 000 cars captured in different parking lots. The UCF-QNRF dataset [11] is released with 1, 535 images and 1.25 million head annotations. Yan *et al.* [14] collect a crowd surveillance dataset including 13, 945 images and 511, 386 marked people in total with high resolutions. Recently, the GCC dataset [15] consists of 15, 212 images with a resolution of 1080×1920, containing 7, 625, 843 persons from synthetic crowd scenes. NWPU-Crowd [13] is a large-scale high-resolution congested crowd counting dataset, including 5, 109 images with a total of 2, 133, 238 instances. JHU-CROWD++ [16] is another large-scale challenging dataset with 4, 372 images and 1.5 million annotations, which is collected from various scenes with different weather and light condition.

2) Video Based Datasets: Different from image based datasets, researchers propose several video based counting datasets [9], [17], [18]. The UCSD dataset [9] is the first video counting dataset with a resolution of 238×158. Similar to UCSD, the Mall dataset [17] is a whole sequence with 2, 000 frames and a resolution of 320×240. Zhang *et al.* [18] present the WorldExpo2010 dataset with 3, 980 annotated video frames, which is captured in 108 different scenarios. Fang *et al.* [8] establish a large-scale video crowd counting dataset which contains $15K$ frames with approximate $394K$ annotated heads from 13 different scenes. Recently, another large-scale DroneCrowd dataset [19], [20] is proposed for crowd counting on drones with over $33K$ frames.

3) Animal Based Datasets: Except for human objects, there exist other animal based datasets [1]–[3], [21]–[24]. Wu *et al.* [1] collect a thermal infrared video dataset to count all bats within the given bounding boxes, where two sequences have the mean numbers of objects per frame 356 and 250 respectively. van Gemert *et al.* [21] focus on drone-centered nature conservation tasks and propose a new dataset with 30 distinct animals in 18, 356 frames. Arteta *et al.* [2] propose a large and challenging dataset of penguins in Antarctica, including 500 thousand images with resolutions between 1MP and 6MP in over 40 different sites. Rey *et al.* [22] ask 232 volunteers to draw 7, 474 polygons around animals they detected in all the 654 images. Kellenberger *et al.* [23] collect a total of 654 images and annotate 976 large animals by convex hull polygons, over the Kuzikus wildlife reserve in eastern Namibia. In [3], the aerial elephant dataset includes 2, 101 images with 15, 511 African bush elephants in their natural habitats. Shao *et al.* [24] construct two subsets of pasture aerial images, *i.e.*, 656 images in subset 1 and 14 images in subset 2.

The majority of aforementioned datasets focus on counting human heads, and other animal based datasets are still limited in scales, species and covered scenarios. CIW [2] is the largest animal dataset to date by containing more than 80k frames. However, it only focuses on penguin counting. Beyond that, most previous animal datasets only include hundreds of frames. To the best of our knowledge, our AnimalDrone dataset is the first large-scale drone-based animal video counting dataset for agriculture and wildlife protection to date. It consists of 162 video sequences in unconstrained scenes with more than 4 million animals including sheep, horses, wolves and yaks. The comparison with existing crowd counting datasets is presented in Table I.

B. Crowd Counting Methods

Generally speaking, existing crowd counting methods fall into three categories: detection-based [26]–[29], regression-based [25], [30]–[33] and density-based [10], [18], [34]–[39]. In addition, we review several related video based counting methods [8], [19], [40], [41].

1) Detection Based Methods: Crowd counting can be formulated as pedestrian or vehicle detection in crowded scenes. The general framework is to detect the location of objects based on the hand-crafted or deep features [26]–[29]. However, it is difficult for detection-based methods to detect small objects in very crowded scenarios with high occlusion.

2) Regression Based Methods: Regression based methods directly estimate the number of the crowd to avoid difficulties in detection [25], [30]–[33]. Tan *et al.* [30] develop a semi-supervised elastic net regression method by considering sequential information between unlabelled samples and their temporally neighboring samples. Chan and Vasconcelos [31]

TABLE I
COMPARISON WITH EXISTING CROWD COUNTING DATASETS. "–" INDICATES DIFFERENT RESOLUTIONS IN THE DATASET

Dataset	Video	Object	Resolution	Frames	Max	Min	Ave	Total	Year
UCSD [9]	✓	people	158×238	2,000	46	11	24.9	49,885	2008
Mall [17]	✓	people	640×480	2,000	53	13	31.2	62,315	2013
UCF_CC_50 [25]		people	-	50	4,543	94	1,279.5	63,974	2013
WorldExpo2010 [18]		people	576×720	3,980	568	1	106.2	199,923	2015
Shanghaitech A [10]		people	-	482	3,139	33	501.4	241,677	2016
Shanghaitech B [10]		people	768×1024	716	578	9	123.6	88,488	2016
UCF-QNRF [11]		people	-	1,535	12,865	49	815.4	1,251,642	2018
FDST [8]	✓	people	1920×1080	15,000	57	9	26.7	394,081	2018
DroneCrowd [19]	✓	people	1920×1080	33,600	455	25	144.8	4,864,280	2019
NWPU-Crowd [13]		people	-	5,109	20,033	0	418	2,133,375	2020
JHU-CROWD++ [16]		people	-	4,372	25,791	0	346	1,515,005	2020
CARPK [12]		vehicle	1280×720	1,448	188	1	62.0	89,777	2017
BUTIV [1]		animal	-	450	392	167	320.9	144,424	2014
NCDALCA [21]		animal	1920×1080	18,356	16	1	5.4	99,122	2014
CIW [2]		animal	-	80,926	1,382	0	73.2	5,919,855	2016
DAASUC [22]		animal	4000×3000	654	30	0	1.5	955	2017
DMU [23]		animal	4000×3000	654	30	0	1.8	1,183	2018
Cattle [24]		animal	4000×3000	670	16	0	2.9	1,948	2020
AnimalDrone-PartA	✓	animal	4096×2160	18,940	568	1	106.0	2,008,570	2020
AnimalDrone-PartB	✓	animal	-	34,704	647	0	58.8	2,040,598	2020
AnimalDrone	✓	animal	-	53,644	647	0	76.4	4,049,168	2020

propose a new Bayesian regression method to estimate the size of inhomogeneous people crowds. Chen *et al.* [32] develop the cumulative attribute based regression model to perform counting when only sparse and imbalanced data are available. In [25], a Markov Random Field method is proposed to estimate counts based on multiple information including low confidence head detections, repetition of texture elements and frequency-domain analysis. Wang *et al.* [33] propose an end-to-end deep CNN regression model for counting people of images in extremely dense crowds, where the training data is enriched with expanded negative samples with zero counting.

3) Density Based Methods: Recent methods have changed the focus onto regarding the crowding counting problem as density map estimation by neural networks due to the impressive performance of deep learning [10], [18], [34]–[37]. Zhang *et al.* [18] solve the cross-scene crowd counting problem using a switchable training scheme with two related learning objectives of CNN model, estimating density map and global count. In [10], the multi-column CNN network is used to learn multi-scale features by each column CNN. Boominathan *et al.* [34] employ a combination of deep and shallow fully convolutional networks to estimate the density map, which captures both the high-level semantic information and the low-level features. Zhao *et al.* [35] determine the crowd counts based on pairs of video frames using a two-phase training scheme. Sam *et al.* [36] propose a novel switching CNN model to handle the variations of crowd density, where the switch classifier can select the optimal regressor for a particular region. Li *et al.* [37] take advantage of dilated kernels to deliver larger reception fields and extract deeper features without losing resolutions. Shi *et al.* [38] introduce attention from segmentation and global density to re-purpose the point annotations used as supervision for density-based counting. Liu *et al.* [39] develop a deep structured scale integration network for crowd counting by using structured feature representation learning and hierarchically structured loss function optimization. Recently, Yang *et al.* [42] excavate the perspective information and quantify the perspective space into several separate scenes by the multi-column framework.

In terms of crowd counting in videos, spatio-temporal information is essential to improve the counting accuracy. Xiong *et al.* [40] design a convolutional LSTM model to fully capture both spatial and temporal dependencies for crowd counting. Zhang *et al.* [41] combine fully convolutional neural networks and LSTM by residual learning to perform vehicle counting. Fang *et al.* [8] present a locality-constrained spatial transformer network by employing a manifold regularization on the neighbourhood frames. Wen *et al.* [19] propose a space-time multi-scale network for video based counting, localization and tracking. Different from the previous methods, our GFAN uses optical flow to warp the multi-scale feature maps in sequential frames to exploit the temporal coherency, and then combines the enhanced features to predict density maps.

III. ANIMALDRONE DATASET

As discussed above, there are few existing datasets suitable for the animal counting task in agriculture and wildlife protection. Therefore, we collect a new large-scale animal counting dataset named AnimalDrone. In this section, we introduce our dataset in detail.

A. Data Collection

The AnimalDrone dataset consists of two subsets, *i.e.*, AnimalDrone-ParA and AnimalDrone-PartB. The videos of

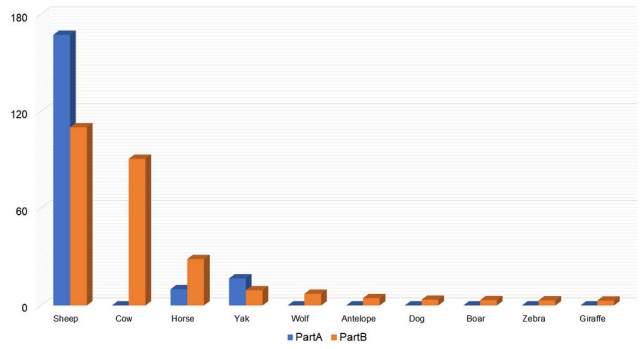


Fig. 2. The distribution of animal species of the AnimalDrone dataset. The y-axis indicates the percentage of videos in the subset.

AnimalDrone-PartA are captured on site by drones in Qinghai and Heilongjiang Provinces of China, with a vast grassland and various types of animals. Specifically, the drone is controlled by an operator to capture the animals in either bird-view or side-view. Notably, we are given permission by the animal owner before data collection and guarantee no harassment to animals. We spend about one month on data acquisition and collected 112 minutes videos. All the videos are captured by DJI MAVIC PRO 2 drone with 4K resolution and the number of frames per second is 24. In order to consider the impact of object scales into account, we choose different flight altitudes to collect the data, *i.e.*, $30m$, $50m$, $80m$. In this way, our dataset can cover different scenes including mountains, grasslands, streams, and wetlands, as well as different animal species such as sheep, horses and yaks.

On the other hand, the videos of AnimalDrone-PartB are collected from two popular video websites, *i.e.*, YouTube[1] and YouKu.[2] We set the key words including *drone videos*, *animal groups herding*, *wildlife*, and *aerial photography animal group*. Finally, we select more than 600 minutes videos captured by drones, which contain 10 animal species including horse, sheep, zebra, giraffe, wolf, cow, yaks, dog, antelope and boar. The distribution of animal species is shown in Fig. 2. Since the data is collected from the Internet, the duration and resolution of videos in AnimalDrone-PartB are not uniform, and the distribution of scenes and heights is also relatively diverse.

B. Data Pruning and Annotations

1) Data Pruning: To remove redundant information in the videos, we perform data pruning on our dataset. Due to the difference between the characteristics of PartA and PartB, we apply different pruning strategies for these two subsets. In PartA, the animals move slowly during most of the time because the drone is usually stationary. To maintain the diversity, the videos are filtered based on the following principles:

- We collect at least one video per height per scene to ensure the variation of perspectives.
- We remove low-quality videos with camera motion and jitter blur.
- We keep the videos where the number of animals changes obviously.

[1]http://www.youtube.com
[2]http://www.youku.com

In this way, we select 59 videos out of the original 112 minutes raw videos in PartA. In PartB, the videos from the Internet contain a lot of irrelevant information with quick perspective changes of drones. We simply select the videos that contain animal communities as many as possible. Finally, we select 103 videos from the original 600 minutes videos in PartB.

2) Data Annotations: Similar to the head annotation in the previous counting datasets [10], [18], the animals in our dataset are annotated by dots instead of rectangular bounding boxes. To this end, we modify the VATIC tool [43] by changing rectangular boxes into dots. The process of annotation includes two steps. We first distribute images to different expert annotators. Then, all the annotated frames are collected from annotators and given to the other experts to find the wrong annotated videos and notify the original annotators to correct the errors until no errors are found. Some annotated examples are shown in Fig. 3.

C. Data Statistics and Attributes

There are totally 162 videos, $53,644$ frames with $4,049,168$ annotated objects in the AnimalDrone dataset. AnimalDrone-PartA is split into two subsets, including the training set (46 sequences) and the testing set (13 sequences). It contains $18,940$ images with $2,008,570$ annotated objects. The maximum and average counts are 568 and 106.0, respectively. AnimalDrone-PartB consists of 103 video clips including 66 training videos and 37 testing videos. It contains $34,704$ frames with $2,040,598$ annotated objects. The maximum and average counts are 647 and 58.8, respectively. The statistics of the existing counting datasets and the proposed AnimalDrone dataset are summarized in Table I. As shown in Fig. 2, our dataset contains 10 animal species including horse, sheep, zebra, giraffe, wolf, cow, yak, dog, antelope and boar. To analyze the counting algorithms comprehensively, we also define several attributes of the dataset as follows.

- *Density:* We define the average number of objects in each frame as the density. Thus we divide the dataset into two density levels including `high-density` (with the number of objects in each frame larger than 100), and `low-density` (with the number of objects in each frame less than 100).
- *Altitude:* The object scales vary in different altitudes. Therefore, we divide the videos based on the flying heights that the data is captured by the drone at `low-height`, `med-height` and `high-height`. For PartA, the height information is recorded during collection. Hence, we set `low-height` ($30m$), `med-height` ($50m$) and `high-height` ($80m$). For PartB, we empirically set the altitude attribute for all videos because the accurate height information is unavailable.
- *View:* The data is captured in different views including `bird-view` and `side-view`, which influences the appearance of the objects. Specifically, bird-view and side-view indicate the camera shooting on the top and on the side of animals, respectively.

The distribution of our dataset is shown in Fig. 4. In summary, the scale variations, tiny objects, view and altitude changes make our proposed AnimalDrone dataset challenging.

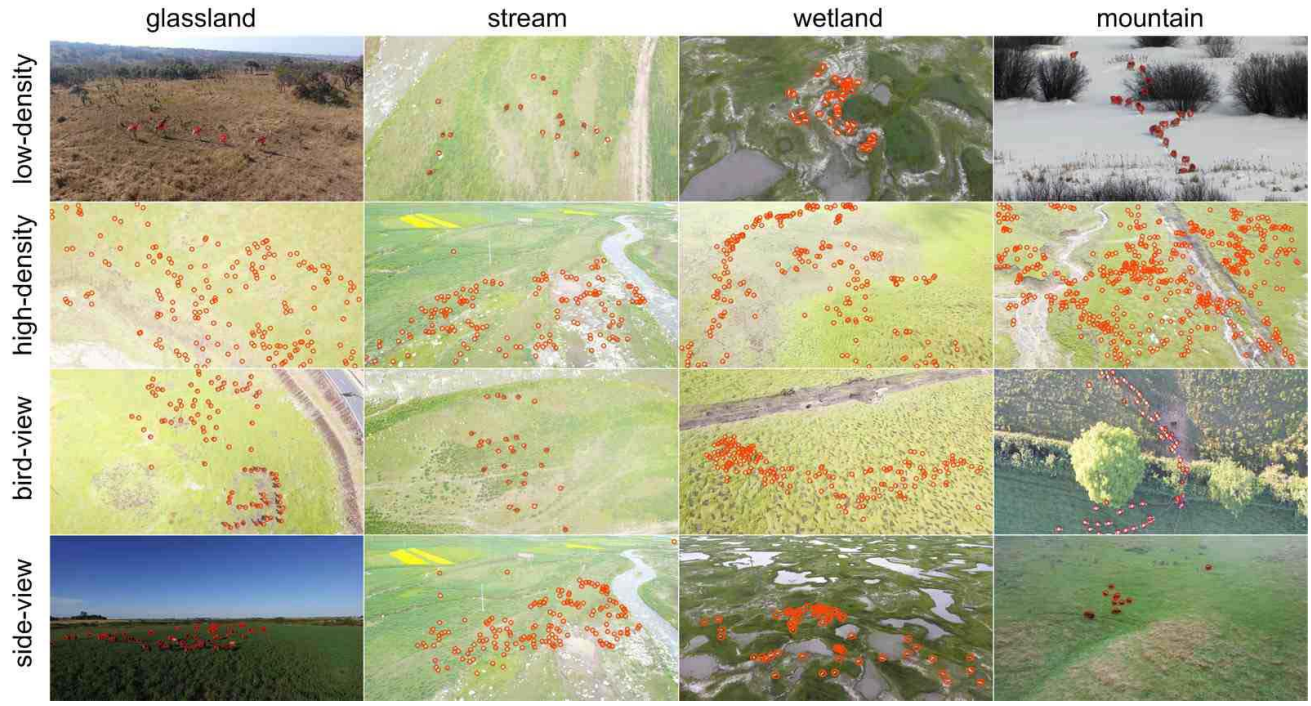


Fig. 3. Some annotated example frames of the AnimalDrone Dataset in various scenes such as glassland, stream, wetland, and mountain. Red circles correspond to animal instances.

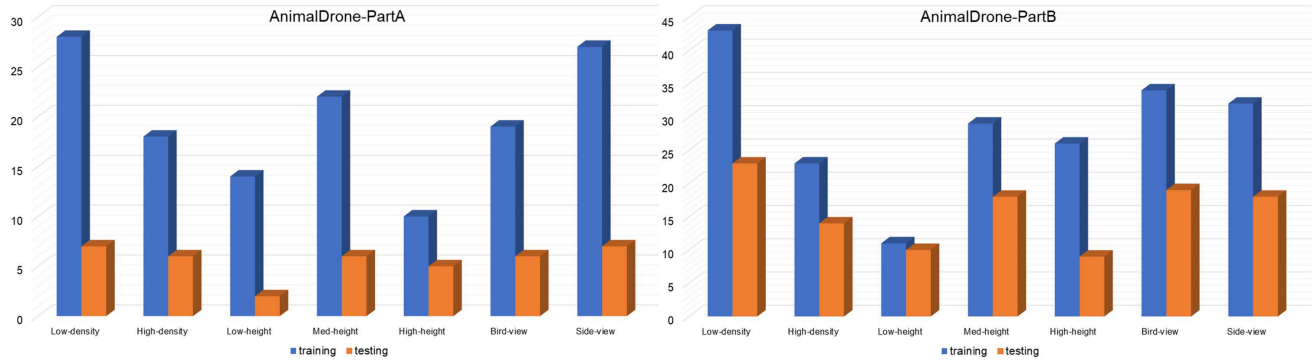


Fig. 4. The attribute statistics of the training and testing sets in the AnimalDrone dataset. The y-axis indicates the number of videos.

IV. Graph Regularized Flow Attention Network

Compared with previous image based crowd counting tasks, there are several issues for video based animal counting. First, different from persons and vehicles, animals could be sparsely or densely distributed because of great scale variations from different heights and views in complex scenes. Second, the temporal coherency across adjacency frames should be well exploited to deal with motion blur and video defocus. To this end, we present the graph regularized flow attention network (GFAN). In the following, we describe each module in our GFAN model and the loss functions in detail.

A. Network Architecture

As shown in Fig. 5, our GFAN consists of three parts: the shared feature encoder module, temporal consistence module, and counting decoder module. The shared feature encoder module uses the first four groups of convolution layers in the VGG-16 network [44] as the backbone. We extract feature maps of the t-th and $(t+\tau)$-th frames in a video. Note that the parameter τ determines the temporal distance between two frames. The temporal consistence module employs the optical flow network [6] to capture the motion information between feature maps in two frames. Specifically, the feature map is warped from the $(t+\tau)$-th frame to the t-th frame to enhance the feature representation. The counting decoder module applies the deconvolutional layer on enhanced feature maps at different scale gradually. That is, we add the feature with lateral connections, and then apply one 1×1 convolution layer to obtain the intermediate density map. After that, one 1×1 convolution layer is used to generate the final density map. It is worth mentioning that we employ the channel attention module [7] to combine multi-scale features, making the network focus on the discriminative features for crowd counting.

B. Overall Loss Function

As discussed above, to train our network, we use three loss terms to consider scale variation and temporal consistency.

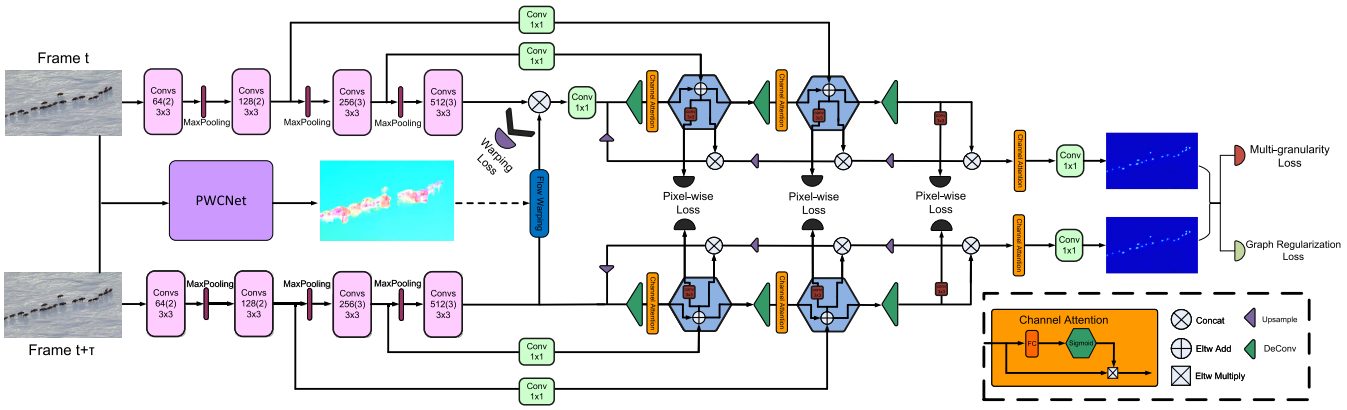


Fig. 5. An overview of the architecture of graph regularized flow attention network (GFAN), which consists of three parts: the shared feature encoder module, temporal consistence module, and counting decoder module.

First, given the aggregated multi-scale feature maps, the multi-granularity loss function is proposed to measure density maps at different levels. Second, to exploit temporal coherence in two adjacent frames, the warping loss is used to calculate the error between feature map obtained from optical flow warping and original feature encoder module. Third, the graph regularization loss is developed to further capture the underlying temporal relations within multiple frames of the clip. In summary, the overall loss function is computed as

$$\mathcal{L} = \mathcal{L}_m + \mathcal{L}_w + \mathcal{L}_g, \tag{1}$$

where $\mathcal{L}_m$, $\mathcal{L}_w$, and $\mathcal{L}_g$ are the multi-granularity loss, warping loss, and graph regularization loss, respectively. Thus our GFAN method can be trained in an end-to-end manner based on large-scale video clips. The three loss terms are defined as follows.

1) Multi-Granularity Loss: As shown in Fig. 3, animals usually gather together, resulting in non-uniform density all over the image. Different from the previous works [10], [36], [37] focusing on counting in high density regions, we pay more attention on the regions with low density to decrease the count error of animals. To this end, we construct a new multi-granularity loss function including pixel-wise density loss and region-wise count loss. Specifically, the pixel-wise loss measures the difference of density between the estimated and ground-truth density maps; while the region-wise loss measures the relative difference of the number of animals in different regions. It is calculated as

$$\mathcal{L}_m = \frac{1}{N}\sum_{n=1}^{N}\left(\begin{array}{l}\sum\limits_{i=1}^{W}\sum\limits_{j=1}^{H}\left\|\mathcal{M}^{(n)}(i,j)-\hat{\mathcal{M}}^{(n)}(i,j)\right\|_2^2 \\ +\lambda\sum\limits_{r\in\mathcal{R}}\left\|\frac{C^{(n)}(r)-\hat{C}^{(n)}(r)}{C^{(n)}(r)+1}\right\|_2^2\end{array}\right), \tag{2}$$

where N is the batch size. W and H are the width and height of the density map, respectively. $\mathcal{M}^{(n)}(i,j)$ and $\hat{\mathcal{M}}^{(n)}(i,j)$ are the ground-truth and estimated density map of location (i,j) of the n-th training sample, respectively. $C^{(n)}(r)$ and $\hat{C}^{(n)}(r)$ are the ground-truth and estimated counts in region r of the n-th training sample, respectively. Specifically, we divide the image into $k \times k$ sub-regions, where the count of each region is calculated as $C(r) = \sum_{(i,j)\in r}\mathcal{M}(i,j)$. $C^{(n)}(r)+1$ is used to avoid the 0 denominator in the region without any object. λ is the factor to balance the two loss terms. As shown in Fig. 6, both the predicted and ground-truth density map are divided into 2×2 sub-regions. Region-wise count loss is the sum of losses of all regions, which is suitable for sparse distribution of objects. Notably, as shown in Fig. 5, we only consider pixel-wise loss of the density maps at intermediate scales, but both pixel-wise and region-wise loses of the final fused density map.

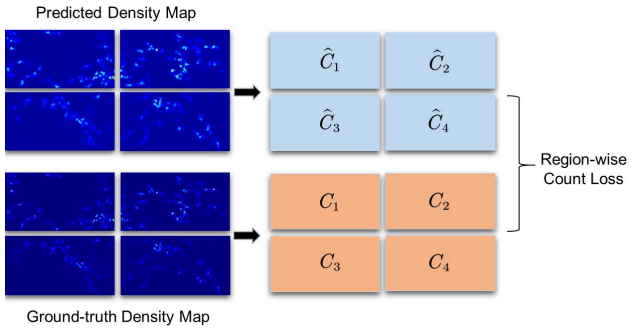


Fig. 6. Illustration of region-wise count loss. The density map is divided into 2×2 sub-regions.

2) Warping Loss: Low accuracy of optical flow directly affects the validity of the aggregated features. The naive strategy is to use the pre-trained PWCNet [6] to extract optical flow between two frames. However, it is difficult to adapt to specific datasets by the fixed optical flow network, resulting in failures of capturing motion coherence.

To consider temporal coherence between two frames, we first generate the bidirectional optical flow $\{f_{t\to t+\tau}, f_{t+\tau\to t}\}$ according to the frame pair $\{I_t, I_{t+\tau}\}$. At the same time, we feed $\{I_t, I_{t+\tau}\}$ into shared feature encoder network to obtain the corresponding feature maps $\{S_t, S_{t+\tau}\}$. Then, we warp S_t and $S_{t+\tau}$ to S'_t and $S'_{t+\tau}$ based on optical flow $\{f_{t\to t+\tau},\ f_{t+\tau\to t}\}$. The offset between S'_t, S_t and $S_{t+\tau}$, $S'_{t+\tau}$ is due to inaccurate optical flow. Finally, we apply the warping loss in Eq. (3) to train the optical flow

network to adapt to our dataset, *i.e.*,

$$\mathcal{L}_w = \sum_{i,j} \left\| \mathcal{S}_{i,j} - \text{warp}(\mathcal{S}_{i,j}, f_{i,j}) \right\|^2, \tag{3}$$

where $\text{warp}(\cdot,\cdot)$ is the warping function for feature maps. $\mathcal{S}_{i,j}$ and $f_{i,j}$ denote the pair of feature maps and optical flow in frame i and j.

3) Graph Regularization Loss: Although optical flow can extract motion information in two frames, underlying relations among multiple frames in a video clip should be considered into account for further improvement. To this end, we introduce the graph regularization loss. According to the manifold assumption [45], [46], the relations of the τ frames of the video clip in the original feature space should be kept in the projected density map space.

First of all, we define the temporal density graph as $\mathcal{G} = \{\mathbf{V}, \mathbf{A}\}$, where the node set $\mathbf{V}$ denotes the density maps of τ frames in a clip and the adjacent matrix $\mathbf{A}$ represents the similarity relations between frame pairs. Specifically, we compute the matrix $\mathbf{A}$ using a RBF-kernel, *i.e.*, $a_{ij} = \exp(-\alpha \left\| \hat{G}^i - \hat{G}^j \right\|_F^2)$, where $\hat{G}^i$ is the ground-truth density map of the i-th frame and α is a pre-set positive constant. Thus we can summarize the graph regularization losses of different frame pairs in a clip, which is defined as

$$\mathcal{L}_g = \sum_{i=1}^{\tau} \sum_{j=1}^{\tau} a_{ij} \left\| \mathcal{M}^i - \mathcal{M}^j \right\|_F^2, \tag{4}$$

where $\mathcal{M}^i$ denotes the density map of the i-th frame and τ is the length of video clip. However, it is time-consuming to calculate the graph regularization loss of each frame pair in a clip. In practice, we choose different temporal difference γ ($1 \leqslant \gamma \leqslant \tau/2$) between frame i and frame j to reduce the computational cost. Therefore, the graph regularization loss in Eq. (4) can be rewritten as

$$\mathcal{L}_g = \sum_{i=\gamma, j=\tau-\gamma+1} a_{ij} \left\| \mathcal{M}^i - \mathcal{M}^j \right\|_F^2. \tag{5}$$

Notably, we calculate the three above loss functions if and only if $i = 1$ and $j = \tau$. In this way, the similarity relations of the density maps in a clip are preserved to keep the long-term temporal coherence.

C. Ground-Truth Map Generation

Similar to the strategy in [10], we generate ground-truth density maps using geometry-adaptive kernels. Specifically, we first blur the center of each object using a normalized Gaussian kernel. Then, we generate the ground-truth map considering the spatial distribution of all objects. The geometry-adaptive kernel is defined as

$$\mathcal{M}(x) = \sum_{i=1}^{\mathcal{N}} \delta(x - x_i) * \mathcal{G}(x, \sigma_i), \quad s.t.\ \sigma_i = \beta \hat{d}_i, \tag{6}$$

where $\mathcal{M}(x)$ is the density map and $\mathcal{N}$ is the number of objects. The delta function $\delta(x - x_i)$ indicates the object at pixel x_i. $\hat{d}_i$ indicates the average distance of the object to its nearest neighbours. To generate the density map, we can convolve $\delta(x - x_i)$ with a Gaussian kernel with the standard deviation $\sigma_i = \beta \hat{d}_i$ ($\beta = 0.3$ in the experiment). Besides, since we use three max pooling layers in the network, the spatial resolution of estimated density map is reduced by $1/8$ for each frame. Therefore, we down-sample the ground-truth density map by $1/8$ in the training stage to calculate the loss.

V. Experiment

We evaluate our GFAN method and 11 state-of-the-art methods on the AnimalDrone dataset, including MCNN [10], MSCNN [47], CSRNet [37], Switch-CNN [36], C-MTL [48], DA-Net [49], CFF [38], DSSIM [39], BL [50], STANet [19], and FCN-rLSTM [41]. All codes of the evaluated methods are publicly available. All counting methods are trained on the `training` set and evaluated on the `testing` set. For comprehensive evaluation, we also evaluate our method on two publicly available video counting dataset with dense annotations, *i.e.*, FDST [8] and UCSD [9]. Finally, we conduct an ablation study on the AnimalDrone dataset to show the effectiveness of three loss functions in the proposed method.

A. Implementation Details

The GFAN method is implemented by Pytorch 1.0.0 [51]. All the experiments are conducted on a workstation with 2.10 GHz Intel E5-2609 CPU, 32GB RAM, and two NVIDIA GeForce GTX 1080Ti GPU cards. The length of the video clips is set as $\tau = 10$ frames. Each frame is divided into 2×2 sub-regions to apply the multi-grantity loss empirically. We use the Adam optimization method [52] to train the network with 200 epochs. The initial learning rate is set as 1×10^{-6} and the weight decay as 5×10^{-4}. We use horizontal flipping strategy to augment the training data. Due to the limit of computational resources, we first resize the frames in the video as 1920×1080 and then randomly crop a patch with a resolution of 960×540 for training. Similarly, during the testing stage, we first resize the frames as 1920×1080 and then divide each frame into 2×2 patches. Finally, the counting results of all the 4 patches are summarized.

B. Evaluation Metrics

According to the work in [10], we use the mean absolute error (MAE) to measure the accuracy of density map estimation, and mean squared error (MSE) to measure the robustness of density map estimation, respectively.

$$\begin{cases} \text{MAE} = \dfrac{1}{K} \displaystyle\sum_{n=1}^{K} |Z_n - \hat{Z}_n|, \\ \text{MSE} = \sqrt{\dfrac{1}{K} \displaystyle\sum_{n=1}^{K} |Z_n - \hat{Z}_n|^2}, \end{cases} \tag{7}$$

where K is the number of images, and n is the n-th image. Z_n and $\hat{Z}_n$ are the ground-truth and estimated counts of the n-th image, respectively. The estimated count is calculated as

$$Z_n = \sum_{i=1}^{W} \sum_{j=1}^{H} \mathcal{M}^n(i, j), \tag{8}$$

where $\mathcal{M}^n(i, j)$ is the density of the pixel (i, j) in the n-th image.

TABLE II
RESULTS ON THE ANIMALDRONE-PARTA DATASET IN TERMS OF ATTRIBUTES

Method	overall		low-density		high-density		low-height		med-height		high-height		bird-view		side-view	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
MCNN [10]	23.2	28.4	15.9	19.5	31.7	36.2	13.7	15.3	23.6	29.2	26.5	31.4	26.4	30.5	20.5	26.6
MSCNN [47]	19.6	26.4	17.9	23.5	21.6	29.8	21.3	27.9	17.4	23.2	21.6	29.6	22.6	29.7	17.0	23.5
CSRNet [37]	14.7	17.8	15.3	17.8	14.2	16.7	25.2	25.5	11.8	14.1	14.1	16.8	15.8	17.6	13.9	17.1
Switch-CNN [36]	18.6	22.7	16.7	20.1	20.8	30.1	16.7	21.4	17.5	19.7	20.7	26.8	17.8	21.2	19.3	23.9
C-MTL [48]	48.4	56.4	41.5	49.6	56.5	63.4	50.4	53.9	45.8	52.3	50.7	61.8	51.5	56.4	45.7	56.3
DA-Net [49]	42.2	54.4	36.9	46.4	48.5	62.4	74.6	75.2	46.4	62.2	24.4	27.9	38.3	51.5	45.6	56.8
CFF [38]	10.1	13.2	**7.3**	**9.5**	13.2	16.3	**8.3**	**9.9**	9.2	12.9	12.6	15.0	10.6	13.8	9.7	12.4
DSSIM [39]	9.1	11.8	7.9	10.2	10.4	13.3	9.6	11.4	7.4	10.3	11.3	13.7	8.9	11.7	9.3	11.9
BL [50]	9.8	12.5	9.9	13.1	9.6	11.7	17.7	20.0	8.2	9.8	**7.2**	9.6	11.3	14.1	8.5	11.1
STANet [19]	9.2	12.8	9.5	13.8	9.1	11.4	13.2	15.5	7.1	11.7	9.3	11.6	10.3	15.2	8.2	10.8
FCN-rLSTM [41]	18.5	23.1	15.9	21.8	21.2	29.1	15.8	18.3	21.7	30.9	14.7	18.8	16.7	26.9	11.5	16.2
GFAN	**6.9**	**8.8**	8.1	10.4	5.7	**7.1**	11.7	13.4	4.5	5.7	7.5	**9.3**	7.4	**9.5**	**6.6**	**8.4**
GFAN-w/o-graph	7.1	9.2	8.3	10.8	**5.6**	**7.1**	12.5	14.3	**4.0**	**5.1**	8.1	10.1	**7.3**	9.7	7.0	8.8
GFAN-w/o-warp	7.5	9.7	8.2	11.0	6.6	8.4	13.1	15.2	4.3	5.8	8.6	10.2	**7.3**	10.3	7.7	9.2
GFAN-w/o-cnt	8.3	10.0	9.4	11.3	8.2	9.6	12.4	16.0	5.6	6.7	8.4	10.3	8.1	10.0	8.4	9.9

TABLE III
RESULTS ON THE ANIMALDRONE-PARTB DATASET IN TERMS OF ATTRIBUTES

Method	overall		low-density		high-density		low-height		med-height		high-height		bird-view		side-view	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
MCNN [10]	36.2	82.1	14.9	24.3	70.3	128.6	10.7	16.8	49.9	106.1	35.8	66.6	47.6	99.8	24.4	58.3
MSCNN [47]	32.4	66.1	12.2	14.3	64.6	104.9	14.4	16.7	41.6	81.3	32.9	64.5	45.4	86.4	19.0	34.0
CSRNet [37]	27.4	62.9	7.3	9.9	59.4	100.5	7.5	11.3	36.0	78.8	31.0	58.6	35.7	80.1	18.6	34.8
Switch-CNN [36]	32.6	68.0	8.5	10.7	71.1	108.7	9.9	12.8	42.6	82.8	36.8	69.4	47.8	90.1	17.0	31.9
C-MTL [48]	37.3	79.0	13.5	17.1	75.5	125.6	9.7	13.5	50.6	98.7	40.3	72.9	52.3	105.2	22.0	38.2
DA-Net [49]	35.1	72.0	11.4	13.7	73.2	114.8	10.5	13.1	46.1	87.8	40.6	74.1	46.3	93.4	23.1	37.2
CFF [38]	23.6	45.8	8.6	12.4	48.5	72.3	7.8	11.3	31.4	56.7	24.9	43.8	26.3	53.9	20.3	35.7
DSSIM [39]	21.3	38.1	7.6	11.5	43.2	59.6	7.0	9.5	26.5	44.8	26.3	42.6	24.0	45.7	18.6	28.3
BL [50]	19.9	38.5	6.9	9.8	40.8	60.1	6.8	10.3	26.5	47.9	20.7	35.8	22.6	44.9	17.0	31.2
STANet [19]	21.8	37.2	8.1	11.3	42.5	58.1	8.2	12.4	24.8	42.1	29.4	39.2	27.2	43.5	15.9	30.5
FCN-rLSTM [41]	30.4	68.6	12.3	15.2	58.3	105.9	8.1	11.2	39.7	85.4	33.6	67.3	39.4	82.2	20.7	36.4
GFAN	**15.0**	**28.3**	5.4	**7.3**	**30.4**	**44.7**	5.2	7.0	**20.9**	**35.1**	**13.5**	**26.8**	**15.6**	**31.4**	**14.4**	**24.9**
GFAN-w/o-graph	16.1	30.3	**5.3**	7.8	33.2	47.7	4.5	6.2	22.8	38.6	14.7	26.5	16.2	32.2	15.9	28.3
GFAN-w/o-warp	17.6	31.4	5.7	**7.3**	36.7	49.8	**3.8**	**5.3**	24.3	38.4	18.7	31.3	16.1	32.6	19.2	32.6
GFAN-w/o-cnt	18.7	36.2	6.1	8.3	38.6	57.5	4.7	6.3	25.5	46.1	20.3	40.2	17.9	41.2	19.6	35.4

C. Evaluation on the AnimalDrone Dataset

1) Overall Evaluation: In Table II and Table III, we present the counting results of compared methods on the AnimalDrone dataset. It can be seen that our method outperforms existing image based and video based methods. On the PartA subset, our GFAN method achieves 6.9 MAE score and 8.8 MSE score. The most competitor DSSIM [39] obtains 9.1 MAE score and 11.8 MSE score. On the PartB subset, we also obtain the best results of 15.0 MAE score and 28.3 MSE score. The second best method is BL [50] that obtains 19.9 MAE score and 38.5 MSE score. The results indicate that our method can generate more accurate density maps in different scenarios. It is worth mentioning that the results in PartB are worse than that in PartA though the density of PartB (58.8) is lower than PartA (106.0). This is maybe because PartB has larger density and scene variety. Besides, CFF [38] and CSRNet [37] also perform well on the AnimalDrone dataset. We speculate that CFF [38] integrates the three tasks of counting, segmentation, and classification to improve counting performance. CSRNet [37] employs dilated convolution layers to enlarge the receptive field, resulting in better discriminative representation for small objects. In addition, as shown in Fig. 7, we randomly select two videos in PartA and PartB to show the comparison between the prediction of our method

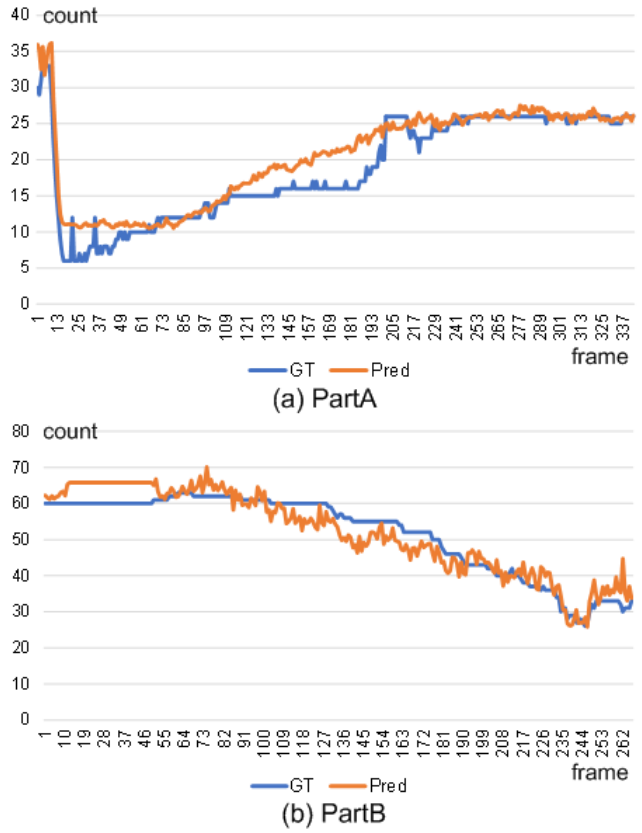


Fig. 7. Smoothness of the crowd counting prediction of our GFAN in videos.

and ground-truth. It indicates that our prediction is relatively smooth across sequential frames.

2) Attributes Based Evaluation: To analyze the results comprehensively, we also report the performance in terms of several attributes including the `low-density` and `high-density` subsets based on the *density* attribute, the `low-height`, `med-height` and `high-height` subsets based on the *altitude* attribute and the `bird-view` and `side-view` subsets based on the *view* attribute.

As presented in Table II and Table III, our method performs much better than the other methods in terms of `high-density` and `high-height` subsets. Note that `high-height` usually corresponds to `high-density` by capturing more objects in a boarder view. This result shows that motion information can be leveraged by joint training to improve the accuracy of density map estimation. Meanwhile, our method performs well in terms of`bird-view` and `side-view` compared with other methods. It indicates that our model is more robust to different perspective views. CFF [38] performs the best in `low-density` on PartA. This is maybe because that the segmentation map can help remove redundant noise in complex scenes. However, if the density becomes higher, it is difficult to obtain an accurate segmentation map, resulting in reduced performance (see the results on PartB). It is worth mentioning that the performance gap between different density of PartB is significantly greater than that in PartA. It shows that PartB is more challenging with great diversity in density due to multiple data source.

3) Per-Category Animal Counting: We also perform per-category animal counting on two subsets of our dataset.

TABLE IV
COUNTING RESULTS OF GFAN IN TERMS OF ANIMAL SPECIES ON THE ANIMALDRONE DATASET

Species	AnimalDrone-PartA		AnimalDrone-PartB	
	MAE	MSE	MAE	MSE
Horse	7.14	7.18	5.56	7.71
Giraffe	–	–	2.08	2.51
Sheep	7.71	8.23	20.12	31.45
Cow	–	–	26.91	47.55
Boar	–	–	5.86	6.27
Wolf	–	–	2.72	3.62
Antelope	–	–	3.26	3.47
Yak	10.99	12.83	3.18	4.61
Dog	–	–	3.32	4.42
Overall	7.30	9.20	15.20	31.20

Note that there are only three animal species in PartA while all nine animal species in PartB. We train our GFAN on the training set, and then calculate both MAE and MSE scores on the corresponding testing set in terms of animal species. As presented in Table IV, our method performs worse in terms of sheep and cow than other animal species on PartB. We speculate that the videos of sheep and cow have large variation on PartB because of multiple sources. Besides, some visual results are shown in the Fig. 8. It show our method can count various animals well on the AnimalDrone dataset.

4) Failure Cases: Although our GFAN performs well on the AnimalDrone dataset, counting performance occasionally degrades or even fails when the drone moves too fast and the animals move too slow. This is due to inaccurate optical flow in complex scenes. Some failure examples are shown in Fig. 9.

D. Evaluation on the FDST and UCSD Datasets

In addition, we compare our method with 7 state-of-art methods including three imaged-based methods (MCNN [10], CSRNet [37] and BL [50]) and four video-based methods (ConvLSTM [40], STANet [19], FCN-rLSTM [41] and LSTN [8]) on two other video based FDST [8] and UCSD [9] datasets.

1) Overall Evaluation: The FDST dataset [8] collects 100 videos captured from 13 different scenes, and contains 150, 000 frames with a total of 394, 081 annotated heads. Besides, the UCSD dataset [9] contains 2, 000 frames with 49, 885. Notably, these two datasets have frame-by-frame annotations rather than only key-frame annotations in the WorldExpo2010 dataset [18]. As presented in Table V, our method achieves the second best 2.45 MAE score and 3.22 MSE score on the FDST datast [8], following the best performer STANet [19] with 2.21 MAE score and 3.00 MSE score. On the UCSD dataset [9], our GFAN method performs the best. By considering scale variation of the objects, the image based MCNN [10] performs better than the video based ConvLSTM [40], *i.e.*, 3.77 vs. 4.48 MAE score. In summary, our GFAN performs better than most video-based methods and all the image-based methods.

2) Complexity Analysis: We analyze the trade-off between error and speed of our method in Table VI. To this end,

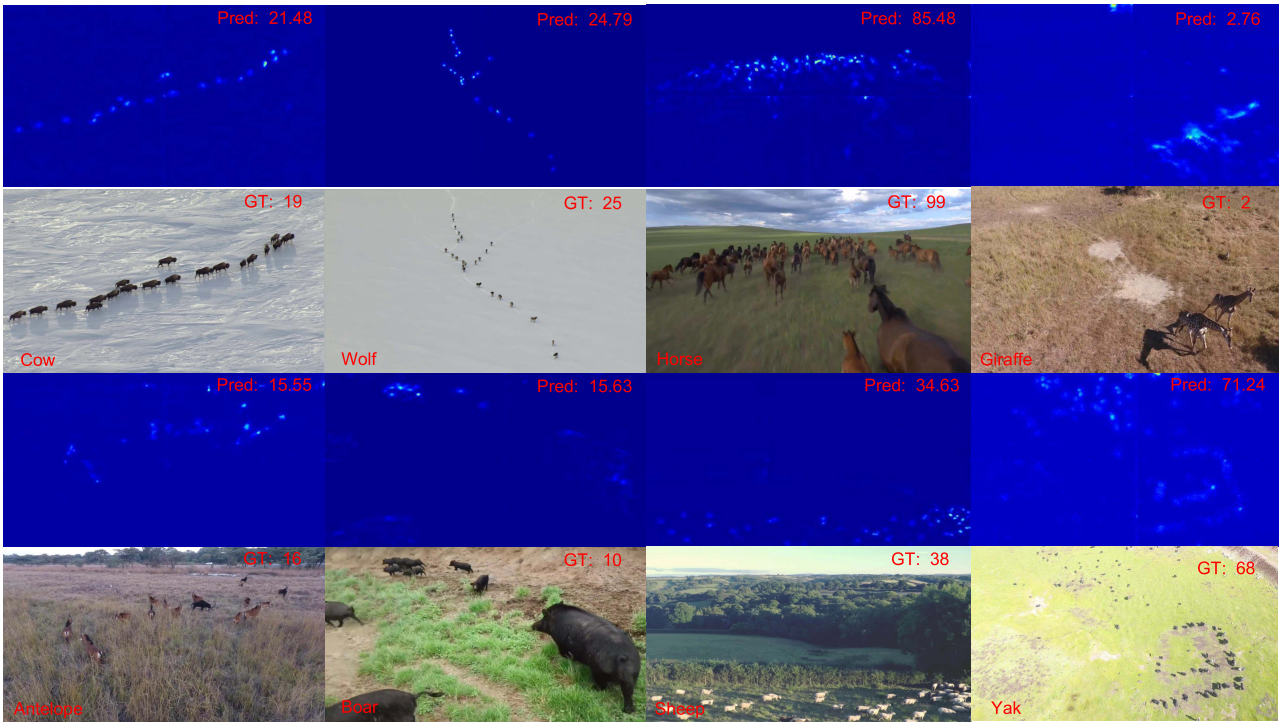


Fig. 8. Visual counting results in terms of animal species on the AnimalCount dataset.

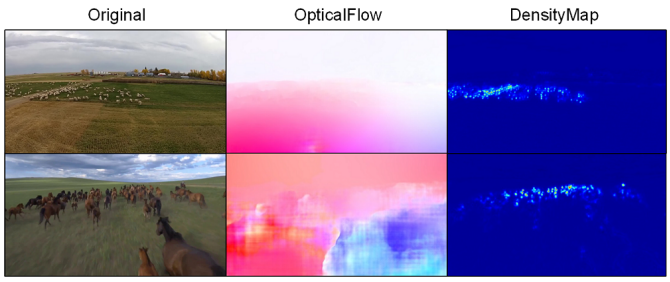


Fig. 9. Some example results of failure cases on the AnimalDrone dataset.

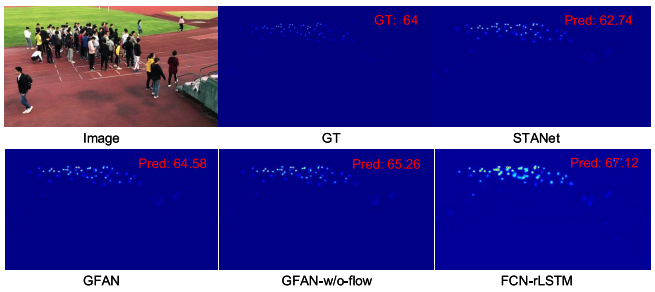


Fig. 10. Visual results of three video based methods on the FDST dataset [8].

TABLE V
RESULTS ON THE FDST AND UCSD DATASETS

Method	FDST [8]		UCSD [9]	
	MAE	MSE	MAE	MSE
MCNN [10]	3.77	4.88	1.07	1.35
CSRNet [37]	3.69	4.82	1.16	1.47
BL [50]	2.85	3.74	1.01	1.27
ConvLSTM [40]	4.48	5.82	1.13	1.43
LSTN [8]	3.35	4.45	1.07	1.39
FCN-rLSTM [41]	3.57	5.30	1.54	3.02
STANet [19]	**2.21**	**3.00**	1.05	1.29
GFAN	2.45	3.22	**0.97**	**1.24**

we construct another GFAN variant by removing the optical flow module in the network, named as GFAN-w/o-flow. Moreover, we compare other crowd counting methods except ConvLSTM [40] and LSTN [8] without publicly available codes. Note that the resolution of testing images is 1920×1080. As presented in Table VI, three video based methods (STANet [19], FCN-rLSTM [41], and our GFAN) have higher FLOPs and slightly inferior running speed than image based methods. This is because extracting temporal information introduces extra computation. With the optical flow module, our GFAN method achieves better performance (2.45 vs. 3.06 MAE score) with one half speed (2.25 vs. 5.56 FPS). Compared with the best performer STANet [19] on the FDST dataset [8], our method performs similarly, *i.e.*, 2.45 vs. 2.21 MAE score. In Fig. 10, we also show some visual results of our method and two compared video based methods including STANet [19] and FCN-rLSTM [41].

E. Ablation Study

To study the influence of the loss function in the proposed network, we construct three variants of the proposed GFAN method, *i.e.*, GFAN-w/o-graph, GFAN-w/o-warp and GFAN-w/o-cnt. For a fair comparison, it is worth mentioning that

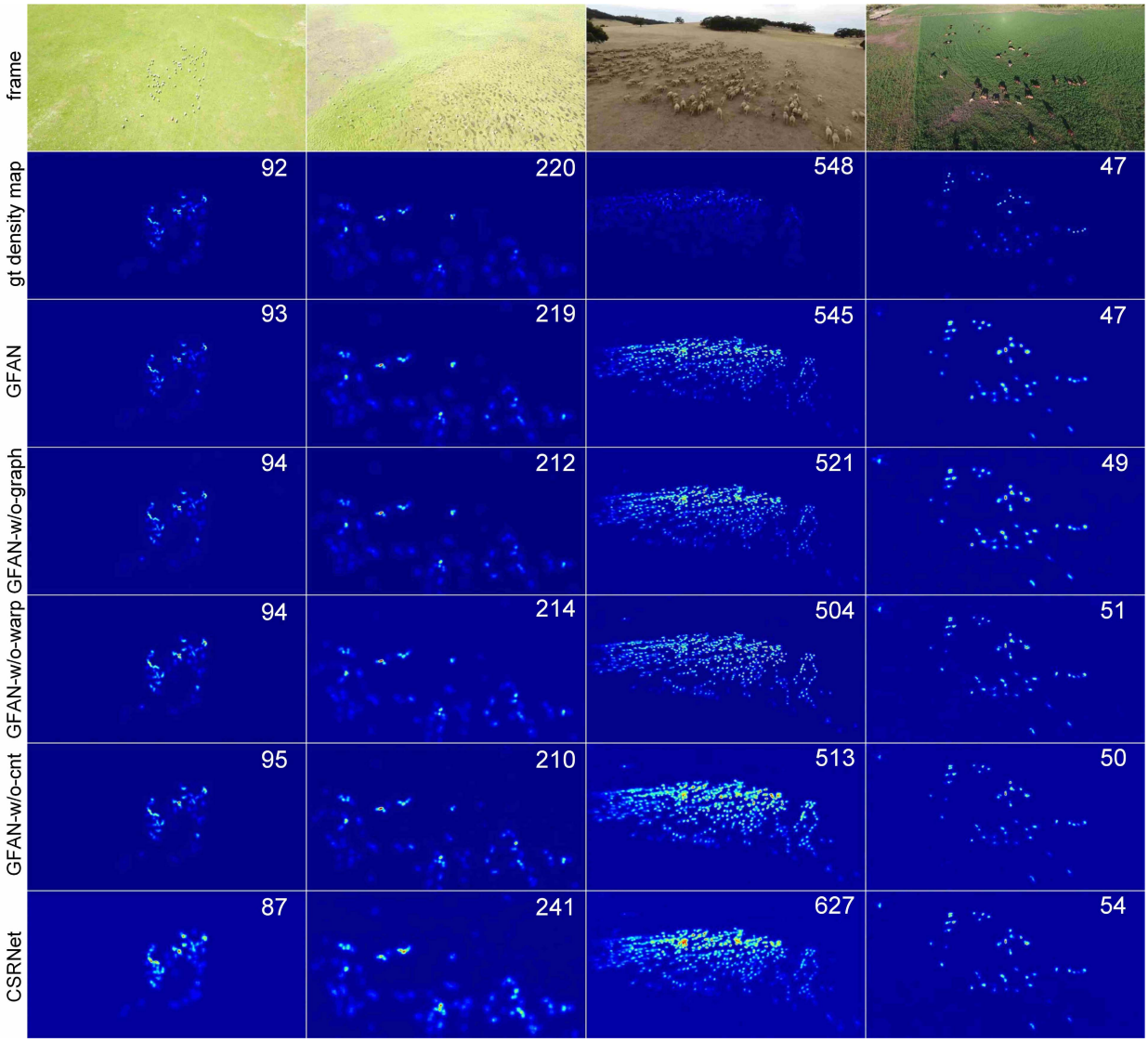


Fig. 11. Some example results of GFAN variants (*i.e.*, GFAN, GFAN-w/o-graph, GFAN-w/o-warp, GFAN-w/o-cnt) and CSRNet [37] on the AnimalDrone dataset. The ground-truth and estimated counts of the frame are presented at the top-right corner.

TABLE VI

TRADE-OFF BETWEEN PERFORMANCES AND RUNNING SPEEDS OF COMPARED METHODS ON THE FDST DATASET [8]

Methods	MAE	MSE	Parameter	FLOPs	FPS
MCNN [10]	3.77	4.88	0.15M	24G	31.26
CSRNet [37]	3.69	4.82	16.26M	815G	4.12
BL [50]	2.85	3.74	21.50M	809G	3.98
FCN-rLSTM [41]	3.57	5.30	10.86M	2,615G	3.07
STANet [19]	2.21	3.00	11.02M	3,302G	2.69
GFAN-w/o-flow	3.06	4.09	10.26M	1,193G	5.56
GFAN	2.45	3.22	18.32M	3,022G	2.25

all variants of GFAN are trained on the AnimalDrone training set and evaluated on the AnimalDrone testing set with the same parameter settings and input size. Specifically, the GFAN-w/o-graph method corresponds to the GFAN method without the graph regularization loss. The GFAN-w/o-warp method removes the warping loss from the GFAN-w/o-graph method and fixes the parameters in the optical flow network. The GFAN-w/o-cnt method denotes that we remove the region-wise count loss from the GFAN-w/o-warp method. As presented in Table II and Table III, our GFAN outperforms its three variants on both PartA and PartB, which demonstrates the effectiveness of our proposed modules.

1) Effectiveness of Graph Regularization Loss: As presented in Table II and Table III, the GFAN-w/o-graph method achieves 7.1 MAE score and 9.2 MSE score in PartA and 16.1 MAE score and 30.3 MSE score in PartB, respectively. That is, if we remove the graph regularization loss, the overall MAE scores increase 0.2 and 1.1 in PartA and PartB. We notice that the GFAN-w/o-graph method performs inferior than the GFAN method in PartB especially in `high-density` (*i.e.*, 33.2 MAE score vs. 30.4 MAE score). This is maybe because the graph regularization can help maintain the temporal consistence in the density map space.

2) Effectiveness of Warping Loss: To verify the effect of warping Loss, we jointly train the optical flow network and counting network through self-supervised manner. It can be

seen than the GFAN-w/o-graph method performs better than the GFAN-w/o-warp method. When the warping loss is used, the MAE score decreases 0.4 and 1.5 in PartA and PartB respectively. It indicates that the warping loss can generate more accurate counting results by aggregating motion information in consecutive frames.

3) Effectiveness of Multi-Granularity Loss: Besides, we evaluate the influence of multi-granularity loss on the counting accuracy. If we further remove the region-wise count loss term in Eq. (1) from the GFAN-w/o-warp method, the MAE score is increased from 7.5 to 8.3 in PartA and from 17.6 to 18.7 in PartB, respectively. In terms of the `low-density` subset, the GFAN-w/o-warp method achieves better performance (*i.e.*, 8.2 MAE score and 11.0 MSE score in PartA, and 5.7 MAE score and 7.3 MSE score in PartB) than the GFAN-w/o-cnt method (*i.e.*, 9.4 MAE score and 11.3 MSE score in PartA, and 6.1 MAE score and 8.3 MSE score in PartB). The decline in counting accuracy of low density frames shows the region-wise count loss facilitates improving the counting performance.

VI. Conclusion

In this paper, we propose the largest video based crowd counting dataset to date. The AnimalDrone dataset includes 162 sequences with 53, 644 frames and more than 4 million annotations in unconstrained wild scenes. Moreover, we develop the Graph regularized Flow Attention Network (GFAN) as a strong baseline to deal with scale variations and temporal coherence in different complex scenarios. The proposed GFAN method outperforms state-of-the-art crowd counting methods on the AnimalDrone dataset in terms of several attributes. We hope this benchmark and the GFAN method can boost the research in animal crowd counting for agriculture and wildlife protection, especially on the drone platform.

References

[1] Z. Wu, N. Fuller, D. Theriault, and M. Betke, "A thermal infrared video benchmark for visual analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 201–208.

[2] C. Arteta, V. S. Lempitsky, and A. Zisserman, "Counting in the wild," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 483–498.

[3] J. Naude and D. Joubert, "The aerial elephant dataset: A new public benchmark for aerial object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2019, pp. 48–55.

[4] Z. Zhao, T. Han, J. Gao, Q. Wang, and X. Li, "A flow base bi-path network for cross-scene video crowd understanding in aerial view," in *Proc. Eur. Conf. Comput. Vis. Workshops*, vol. 12538, 2020, pp. 574–587.

[5] M. A. Hossain, K. Cannons, D. Jang, F. Cuzzolin, and Z. Xu, "Video-based crowd counting using a multi-scale optical flow pyramid network," in *Proc. Asian Conf. Comput. Vis.*, vol. 12626, 2020, pp. 3–20.

[6] D. Sun, X. Yang, M. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.

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

[8] Y. Fang, B. Zhan, W. Cai, S. Gao, and B. Hu, "Locality-constrained spatial transformer network for video crowd counting," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 814–819.

[9] A. B. Chan, Z. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–7.

[10] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 589–597.

[11] H. Idrees *et al.*, "Composition loss for counting, density map estimation and localization in dense crowds," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 544–559.

[12] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4165–4173.

[13] Q. Wang, J. Gao, W. Lin, and X. Li, "NWPU-crowd: A large-scale benchmark for crowd counting and localization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 2141–2149, Jun. 2021.

[14] Z. Yan *et al.*, "Perspective-guided convolution networks for crowd counting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 952–961.

[15] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 8198–8207.

[16] V. A. Sindagi, R. Yasarla, and V. M. Patel, "JHU-CROWD++: Large-scale crowd counting dataset and A benchmark method," 2020, *arXiv:2004.03597*. [Online]. Available: https://arxiv.org/abs/2004.03597

[17] C. C. Loy, S. Gong, and T. Xiang, "From semi-supervised to transfer counting of crowds," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2256–2263.

[18] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 833–841.

[19] L. Wen *et al.*, "Drone-based joint density map estimation, localization and tracking with space-time multi-scale attention network," 2019, *arXiv:1912.01811*. [Online]. Available: https://arxiv.org/abs/1912.01811

[20] D. Du *et al.*, "VisDrone-CC2020: The vision meets drone crowd counting challenge results," in *Proc. Eur. Conf. Comput. Vis. Workshops*, vol. 12538, 2020, pp. 675–691.

[21] J. C. van Gemert, C. R. Verschoor, P. Mettes, K. Epema, L. P. Koh, and S. A. Wich, "Nature conservation drones for automatic localization and counting of animals," in *Proc. European Conf. Comput. Vis. Workshops*, vol. 8925, 2014, pp. 255–270.

[22] N. Rey, M. Volpi, S. Joost, and D. Tuia, "Detecting animals in African savanna with UAVs and the crowds," *Remote Sens. Environ.*, vol. 200, pp. 341–351, Oct. 2017.

[23] B. Kellenberger, D. Marcos, and D. Tuia, "Detecting mammals in UAV images: Best practices to address a substantially imbalanced dataset with deep learning," *Remote Sens. Environ.*, vol. 216, pp. 139–153, Oct. 2018.

[24] W. Shao, R. Kawakami, R. Yoshihashi, S. You, H. Kawase, and T. Naemura, "Cattle detection and counting in UAV images based on convolutional neural networks," *Int. J. Remote Sens.*, vol. 41, no. 1, pp. 31–52, Jan. 2020.

[25] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2547–2554.

[26] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 878–885.

[27] M. Li, Z. Zhang, K. Huang, and T. Tan, "Estimating the number of people in crowded scenes by MID based foreground segmentation and head-shoulder detection," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.

[28] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[29] C. YuanQiang *et al.*, "Guided attention network for object detection and counting on drones," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 709–717.

[30] B. Tan, J. Zhang, and L. Wang, "Semi-supervised elastic net for pedestrian counting," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2297–2304, Oct. 2011.

[31] A. B. Chan and N. Vasconcelos, "Counting people with low-level features and Bayesian regression," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2160–2177, Apr. 2012.

[32] K. Chen, S. Gong, T. Xiang, and C. C. Loy, "Cumulative attribute space for age and crowd density estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2467–2474.

[33] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 1299–1302.

[34] L. Boominathan, S. S. S. Kruthiventi, and R. V. Babu, "CrowdNet: A deep convolutional network for dense crowd counting," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 640–644.

[35] Z. Zhao, H. Li, R. Zhao, and X. Wang, "Crossing-line crowd counting with two-phase deep neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 712–726.
[36] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4031–4039.
[37] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1091–1100.
[38] Z. Shi, P. Mettes, and C. Snoek, "Counting with focus for free," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4199–4208.
[39] L. Liu, Z. Qiu, G. Li, S. Liu, W. Ouyang, and L. Lin, "Crowd counting with deep structured scale integration network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1774–1783.
[40] F. Xiong, X. Shi, and D.-Y. Yeung, "Spatiotemporal modeling for crowd counting in videos," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5161–5169.
[41] S. Zhang, G. Wu, J. P. Costeira, and J. M. F. Moura, "FCN-rLSTM: Deep spatio-temporal neural networks for vehicle counting in city cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 3687–3696.
[42] Y. Yang, G. Li, D. Du, Q. Huang, and N. Sebe, "Embedding perspective analysis into multi-column convolutional neural network for crowd counting," *IEEE Trans. Image Process.*, vol. 30, pp. 1395–1407, 2021.
[43] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently scaling up crowdsourced video annotation: A set of best practices for high quality, economical video labeling," *Int. J. Comput. Vis.*, vol. 101, no. 1, pp. 184–204, Jan. 2013.
[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
[45] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 153–160.
[46] M. Wang, W. Fu, S. Hao, D. Tao, and X. Wu, "Scalable semi-supervised learning by efficient anchor graph regularization," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1864–1877, Jul. 2016.
[47] L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang, "Multi-scale convolutional neural networks for crowd counting," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 465–469.
[48] V. A. Sindagi and V. M. Patel, "CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.
[49] Z. Zou, X. Su, X. Qu, and P. Zhou, "DA-Net: Learning the fine-grained density distribution with deformation aggregation network," *IEEE Access*, vol. 6, pp. 60745–60756, 2018.
[50] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6141–6150.
[51] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.
[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

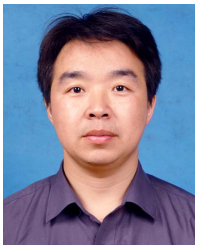
Pengfei Zhu received the B.S. and M.S. degrees from the Harbin Institute of Technology, Harbin, China, in 2009 and 2011, respectively, and the Ph.D. degree from The Hong Kong Polytechnic University, Hong Kong, SAR, China, in 2015. He is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University. His research interests include machine learning and computer vision.

Tao Peng received the bachelor's degree from Tianjin University, Tianjin, China, in 2014. He is currently pursuing the master's degree with the College of Intelligence and Computing, Tianjin University. His current research interests include object counting and object detection.

Dawei Du received the B.S. and M.S. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2010 and 2013, respectively, and the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2018. He is currently a Postdoctoral Researcher with the University at Albany, State University of New York, Albany, NY, USA. His current research interests include computer vision and machine learning, with a focus on visual tracking, object detection, and video segmentation.

Hongtao Yu received the bachelor's degree from the Jiangsu University of Science and Technology, Zhenjiang, China, in 2015. He is currently pursuing the master's degree with the College of Intelligence and Computing, Tianjin University. His current research interests include object tracking and object detection.

Libo Zhang received the Ph.D. degree in computer software and theory from the University of Chinese Academy of Sciences, Beijing, China, in 2017. He is currently an Associate Research Professor with the Institute of Software Chinese Academy of Sciences, Beijing. His current research interests include image processing and pattern recognition. He is selected as a member of the Youth Innovation Promotion Association, the Chinese Academy of Sciences, and the Outstanding Youth Scientist of Institute of Software Chinese Academy of Sciences.



Qinghua Hu (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively. He was a Postdoctoral Fellow with the Department of Computing, The Hong Kong Polytechnic University, from 2009 to 2011. He is currently the Dean of the School of Artificial Intelligence, the Vice Chairman of the Tianjin Branch of China Computer Federation, and the Vice Director of the SIG Granular Computing and Knowledge Discovery and the Chinese Association of Artificial Intelligence. He is also supported by the Key Program, National Natural Science Foundation of China. He has published over 200 peer-reviewed articles. His current research interests include uncertainty modeling in big data, machine learning with multi-modality data, and intelligent unmanned systems. He is also an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Acta Automatica Sinica*, and *Energies*.