Autonomous UAV Safety by Visual Human Crowd Detection Using Multi-Task Deep Neural Networks

Christos Papaioannidis[1], Ioannis Mademlis[1], Ioannis Pitas[1]

***Abstract*— Camera-equipped UAVs, or drones, are increasingly employed in a wide range of applications. Thus, ensuring their safe flight in areas containing people is a top priority. In this paper, a deep neural network-based method is proposed for the task of visual human crowd detection from UAV footage, allowing a drone to rapidly extract semantic segmentation maps from captured video frames during flight. These maps can be exploited (e.g., by a path planner) to define no-fly zones over, or near human crowds and, hence, enhance UAV flight safety. To this end, a novel neural architecture for binary (crowd/non-crowd) semantic segmentation from single RGB images is proposed, based on Convolutional Neural Networks (CNNs). It consists of a semantic segmentation and an image-to-image translation (I2I) neural branch. The overall network is trained using a novel multi-task loss function that addresses both tasks by processing the output of the corresponding branch. During inference, information flows across branches through additional skip synapses to further assist the crowd detection task. In order to evaluate the proposed method, we introduce a real and a synthetic human crowd RGB image dataset. The proposed method outperforms previous aerial crowd detection methods by a large margin and without any post-processing. Moreover, it demonstrates increased generalization ability, while running at real-time and near-real-time speeds on a ground computer and on embedded AI hardware, respectively.**

I. Introduction

Over the last few years, Unmanned Aerial Vehicles (UAVs) have been utilized in various applications such as surveillance [1], area mapping [2] or search and rescue operations [3]. In similar scenarios, UAVs might be required to operate near groups of people, raising significant safety and legal issues due to possible malfunctions and/or regulations that forbid flight in the vicinity of human crowds. Relevant examples include infrastructure inspection in populated areas, or cinematography/media production applications [4]–[8], where it is typical to find crowds within the flight area (e.g., spectators of an outdoors sports event, etc.). Under such conditions, autonomous UAV operation requires special precautions.

Improved safety can be achieved by defining no-fly zones, in order to avoid operation near/over people. Human crowd detection on video frames captured from UAV cameras offers an effective solution, as safety can be ensured by visually recognizing crowded areas on-frame and, subsequently, actively avoiding them in 3D space (e.g., by back-projecting them onto the 3D area map [31] and correspondingly constraining the path planner). While a strict definition of human crowd is not commonly accepted, the national legislation of Germany prohibits UAV operation at a distance of less than 100 m from assemblages of more than 12 individuals, which is the crowd definition adopted in this work.

Human crowd detection entails detecting *crowd* and *non-crowd* regions on the 2D image/video frame. Previous methods approached the crowd detection problem either by applying a probabilistic model on extracted image features [9], [10], or by training a Fully Convolutional Network (FCN) [11] to classify video frame patches in two classes, *crowd* and *non-crowd* [12]–[14], [28]. Alternatively, Convolutional Neural Networks (CNNs) [16] were trained to perform crowd counting [20], [23]–[25] or directly regress crowd density maps [15], [17]–[19], from which human crowd regions may be obtained by applying image processing methods. Although these methods were able to predict heatmaps or density maps that indeed capture visible crowd regions, the region boundaries are not strictly delineated on the 2D video frame. Therefore, an extra post-processing step needs to be applied on the output heatmaps or density maps to obtain the final *crowd* and *non-crowd* regions. This extra post-processing step, which usually consists of simple image processing methods (e.g., thresholding/binarization, Gaussian blur, etc.), is not at all robust to distribution shifts between the training and the test set, while it adds undesirable computational complexity. The latter point is especially problematic in embedded systems with limited computational capabilities, as is typically the case with drones. In autonomous UAV flight, sluggish prediction of visible 2D crowd regions raises safety issues: when slow inference is combined with increased vehicle flight speed, crowd regions that need to be avoided may easily be missed.

To overcome the above issues, we propose transforming the crowd detection problem into a binary semantic image segmentation one, where each pixel of the input video frame is classified as belonging to either the *crowd* or the *non-crowd* class. Thus, more accurate (pixel-level) crowd region boundaries can be obtained, while a post-processing step is no longer necessary. Following this direction, the proposed method introduces a novel CNN architecture for rapid crowd segmentation in single RGB images. It utilizes a real-time semantic image segmentation CNN as the main neural branch and an Image-to-Image Translation (I2I) [22] network as the auxiliary branch to aid the main branch in the crowd segmentation task. This is accomplished through skip synapses that are added between them, in order to allow information flow

[1]Christos Papaioannidis, Ioannis Mademlis and Ioannis Pitas are with Department of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece. email: cpapaionn@csd.auth.gr, imademlis@csd.auth.gr, pitas@csd.auth.gr

This work has received funding from the European Union's Seventh Horizon 2020 research and innovation programme under grant agreement numbers 731667 (MULTIDRONE) and 871479 (AERIAL-CORE).

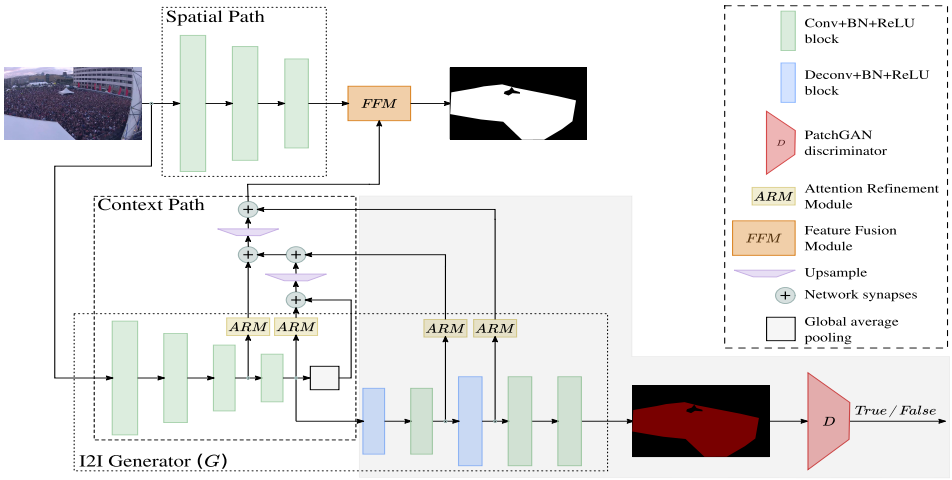

Fig. 1. Overall network architecture of the proposed crowd detection method during training. The semantic image segmentation branch consists of the Spatial Path and the Context Path, while the I2I branch consists of the generator G, which is followed by the discriminator D. The backbone CNN (ResNet-18) is shared between the two branches. The proposed additions to the baseline semantic segmentation network lie in the shaded area.

from the I2I branch to the segmentation branch, thus providing extra context for crowd detection. The overall network is trained using a novel multi-task objective function that involves both semantic segmentation and I2I. Finally, we also introduce two human crowd segmentation image datasets, DroneCrowd and AirSimCrowd, which consist of real and synthetic UAV crowd images, respectively, along with their annotated segmentation maps. The proposed method was evaluated on both datasets, outperforming previous visual crowd detection methods while being significantly faster. Note that the proposed method is a generic, visual-based one requiring only an RGB camera, thus, it is directly applicable to any camera-equipped UAV.

In summary, the contributions of this paper are threefold:

- A novel composite CNN architecture for human crowd detection is introduced, combining in parallel two neural building blocks (a semantic segmentation and a I2I branch) that utilize a common feature extraction backbone and additional skip synapses between them.
- A novel multi-task loss function is employed for training the proposed architecture.
- Two new human crowd segmentation image datasets are introduced for evaluating the proposed method.

II. Crowd Segmentation

In this work, crowd detection is approached as a semantic image segmentation problem, where each pixel of the input UAV video frame is assigned a per-class probability for each of the two object classes (*crowd/non-crowd*). Thus, for an input resolution of $M \times N$ pixels, the output is a $M \times N \times 2$ crowd *segmentation map*. With this goal in mind, a novel deep CNN architecture for crowd segmentation is proposed, which combines a semantic image segmentation network with an I2I network to accurately predict crowd segmentation maps. he I2I neural branch is used to provide extra semantic information to the segmentation neural branch through skip synapses that connect the two branches, further assisting the crowd segmentation task. The two networks share a single backbone/feature extraction CNN and are jointly trained using a multi-task objective function.

A. Semantic Image Segmentation Branch

Given an input image/video frame $\mathbf{x}$, semantic image segmentation assigns object class probabilities to each input pixel. Since human crowd positions in the 3D world might change dynamically during UAV flight, regular and frequent semantic video feed analysis is a necessity. Thus, BiSeNet [21] was employed as the baseline semantic segmentation neural branch, due to its real-time processing capabilities, and thus is briefly described below. *Note, however, that any fast CNN for semantic image segmentation could be utilized in its place*.

BiSeNet adopts a two-column network architecture consisting of two neural streams, namely, the Spatial Path and the Context Path. The Spatial Path is composed of a shallow CNN in order to learn high-resolution features that encode spatial information. In contrast, pre-trained state-of-the-art CNN architectures are utilized in the Context Path to encode high level semantic context information. Moreover, the features of each stage of the Context Path are refined using an Attention Refinement Module (ARM) to guide the learning process. As features from the Spatial and the Context Path encode different information, a Feature Fusion Module (FFM) was also utilized to effectively fuse the learned features. The final segmentation map is, then, obtained by upsampling the combined feature map to the output resolution. The loss function employed for training is the following one:

$$\mathcal{L}_{segm} = \mathcal{L}_p + \alpha \sum_{i=2}^{3} \mathcal{L}_{a_i}, \tag{1}$$

where $\mathcal{L}_p$ is the principal loss used to supervise the whole network and $\mathcal{L}_{a_i}$ is an auxiliary loss for stage i of the Context Path. α is used to weight the contribution of the auxiliary losses in the total loss. Both $\mathcal{L}_p$ and $\mathcal{L}_{a_i}$ are standard Softmax loss functions. Finally, note that for an input video frame resolution of $M \times N$ pixels, the output of the semantic

segmentation branch, as well as the corresponding ground-truth during training, is a $M \times N \times 2$ tensor.

B. Image-to-Image Translation Branch

Given paired training samples $\{\mathbf{x}_i, \mathbf{y}_i\}, i = 1 \dots N$, where $\mathbf{x}_i \in \mathcal{X}$ are images belonging to a source domain $\mathcal{X}$ and $\mathbf{y}_i \in \mathcal{Y}$ images belonging to a target domain $\mathcal{Y}$, I2I methods [22] aim to learn a mapping, $G : \mathcal{X} \mapsto \mathcal{Y}$. G is typically represented by an encoder-decoder CNN architecture, trained under the conditional Generative Adversarial Network (GAN) [27], [38] framework. Conditional GANs consist of two competing networks, the generator and the discriminator. Given samples originating from the source domain, the generator aims to produce outputs that are similar to target domain samples and cannot be distinguished by the discriminator, which is adversarially trained to detect the generator's "fake" outputs.

In the proposed method, I2I is employed as an auxiliary task to aid semantic segmentation; the underlying intuition is that adversarial learning can complement typical supervised learning. Thus, during training, RGB images of resolution $M \times N$ containing crowds serve as source domain data, while their corresponding ground-truth RGB *segmentation images* (tensors of size $M \times N \times 3$, derived by trivially processing the corresponding segmentation maps) are utilized as target domain data. RGB segmentation images constitute simply an alternative representation of the segmentation map ground-truth, one necessary for training the I2I neural branch of the proposed architecture. This branch corresponds to G, serving as the generator network whose objective is to learn the underlying mapping from real crowd images (source domain) to RGB segmentation images (target domain), while the objective of the employed discriminator D is to distinguish samples produced by G from ground-truth RGB segmentation images. As in typical conditional GANs, both G and D are trained in a supervised manner via the minimax game, $min_G max_D \mathcal{L}_{cGAN}(G, D)$, where the objective function $\mathcal{L}_{cGAN}(G, D)$ is given by [22]:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{(\mathbf{x},\mathbf{y})}[\log D(\mathbf{x}, \mathbf{y})] + \mathbb{E}_{\mathbf{x}}[\log(1 - D(\mathbf{x}, G(\mathbf{x})))]. \quad (2)$$

In alignment with previous methods [22], [32], we also train G to not only fool D, but also to generate RGB segmentation images that are "close" to the corresponding ground-truth target domain images. As in [22], we utilize the $L1$ distance in the employed similarity loss function:

$$\mathcal{L}_s(G) = \mathbb{E}_{(\mathbf{x},\mathbf{y})}[\|\mathbf{y} - G(\mathbf{x})\|_1]. \quad (3)$$

Apart from $\mathcal{L}_{cGAN}(G, D)$ and $\mathcal{L}_s(G)$, which are typically used for training I2I networks [22], we also train the generator G to predict regular crowd segmentation maps, in order to prevent the backbone network from losing focus from our main task (crowd semantic segmentation). Therefore, the final objective function used to train the I2I neural branch of the proposed network is defined as follows:

$$\mathcal{L}_{i2i} = \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \mathcal{L}_s(G) + \mathcal{L}_a(G), \quad (4)$$

where $\mathcal{L}_a(G)$ is an auxiliary semantic segmentation loss function for the penultimate convolutional layer of the decoder part of G, which is similar to the ones ($\mathcal{L}_p$, $\mathcal{L}_{a_i}$) used in Eq. (1).

C. Combining Semantic Segmentation with Image-to-Image Translation

The proposed method combines the semantic image segmentation branch and the I2I branch in a novel, unified network architecture for crowd segmentation, which is illustrated in Fig. 1. The employed semantic image segmentation neural branch consists of the Context Path and the Spatial Path, while the I2I neural branch consists of the generator G followed by the discriminator D. The backbone network (ResNet-18 [26]) of the Context Path is shared between the two branches, serving also as the encoder of the generator G. The decoding network of G is a CNN that uses both convolutional and deconvolutional layers and D is a standard PatchGAN [22] classifier, similar to the one used in [22]. Moreover, in order to allow information flow from the I2I branch to the semantic segmentation branch to enrich the extracted semantic features, skip synapses between neurons of the two intermediate stages of G decoder (I2I branch) and the segmenter's Context Path were added, conjoining the two branches. Importantly, as the final crowd segmentation maps are obtained from the main segmentation branch, the discriminator D and the two last convolutional layers of the generator G are necessary only during training, thus, they are discarded in deployment-time to avoid extra computational cost during inference.

The overall network is trained using the proposed multi-task loss function that combines semantic image segmentation with image-to-image translation:

$$\mathcal{L} = \lambda \mathcal{L}_{segm} + (1 - \lambda)\mathcal{L}_{i2i}, \quad (5)$$

where λ is a hyper-parameter used to adjust focus between the two tasks.

The advantages of the proposed method are threefold. First, the proposed multi-task loss function assists the main crowd segmentation task, by effectively complementing typical supervised learning with adversarial learning. Since it is well-known that GANs are inherently resistant to overfitting [38], enhancing supervised training with an adversarial objective seems to have a regularizing effect. Second, the backbone CNN (ResNet-18) serves both as the encoder of the I2I branch and as the feature extractor of the crowd segmentation branch, thus saving significant computational cost and introducing additional regularization. Finally, the proposed parallel network architecture facilitates multi-task learning and allows the auxiliary I2I neural branch to further assist the main crowd segmentor through skip synapses.

III. Empirical Evaluation

This Section provides a detailed description of the two human crowd image datasets we are introducing, along with the metrics used to evaluate the proposed crowd detection method. In addition, quantitative and qualitative performance evaluations of the proposed method are presented.

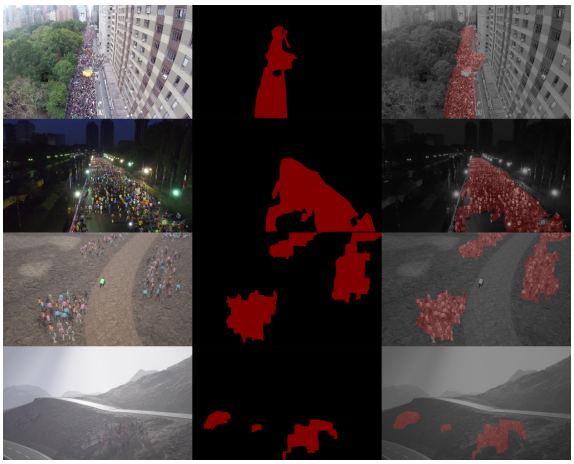

Fig. 2. Example samples of the manually annotated DroneCrowd (rows 1 and 2) and the synthetic AirSimCrowd (rows 3 and 4) datasets. Left column: RGB image, middle column: RGB segmentation image, right column: visualization of annotation.

A. Datasets and Metrics

Although there are existing aerial-footage datasets for crowd counting (e.g., VisDrone [37]), they cannot be used for dense crowd detection since people appear scattered across the image, thus not forming crowd. Therefore, two suitable datasets were created and annotated to evaluate the proposed method: DroneCrowd and AirSimCrowd[1].

The *DroneCrowd* dataset consists of RGB images depicting human crowds in a wide range of scenes (urban, countryside, day, night), captured at varying altitudes (from low to very high altitudes) and with varying crowd density (from tens to thousands people). In order to induce this diversity in the dataset, we included images from the Crowd-Drone dataset [13], the dataset used in [29], as well as newly-captured relevant aerial images[2]. For the latter, in total five separate UAV flight missions were performed over two different terrains using a custom drone equipped with an RGB camera mounted on a gimbal. In total 1700 images were manually annotated with their ground-truth segmentation maps using annotation software [36], resulting in a very diverse and challenging human crowd detection dataset. The image resolution varies from 480 $\times$ 360 to 1920 $\times$ 1080 pixels, rendering the dataset even more challenging. From these images, 1199 are used for training and 591 for testing. The train set consists of the train images from the Crowd-Drone dataset, *Sequence3*, *Sequence8*, *Sequence9*, *Sequence10*, *Sequence11*, *Sequence16* from [29] and images captured during the three of the five performed missions, including both terrains. In a complementary manner, the test set includes images from the the Crowd-Drone test set and images captured during the remaining two UAV flight missions, ensuring that the train and test sets are mutually exclusive. Example samples from the DroneCrowd dataset can be seen in rows 1 and 2 of Fig. 2.

[1]DroneCrowd and AirSimCrowd datasets are available at `https://aiia.csd.auth.gr/open-multidrone-datasets`.

[2]The MULTIDRONE project experimental media productions are the corresponding source.

The *AirSimCrowd* dataset is a synthetic crowd detection dataset obtained from the UAV simulation software AirSim [30]. AirSim is a photorealistic UAV simulation environment, built on top of the advanced Unreal 4 (UE4) real-time 3D graphics/physics engine, which allows programmatic interaction with the simulated UAVs via Remote Procedural Call (RPC)-based communication and offers tools for RGB image and ground truth annotation data extraction. In order to create the AirSimCrowd dataset, we simulated two scenarios. First, a cycling scenario on a mountainous environment, where a cyclist is set to traverse a pre-defined route with crowd gathered at random locations along the route. During the simulation, a UAV was set to follow the cyclist at a relatively constant speed while recording video with a RGB camera pointing at the cyclist. The second scenario involved a UAV following a predefined, random trajectory near crowds placed at random locations in the scene. The scene used in this second scenario is different from the one used in the cycling scenario, in order to induce diversity in the AirSimCrowd dataset samples. Overall, 602 RGB images at a resolution of 640 $\times$ 360 pixels along with their corresponding ground-truth segmentation maps were obtained from both simulated scenarios. Example video frames of the footage captured by the simulated UAV in both scenarios are depicted in rows 3 and 4 in Fig. 2. Note that all images in the AirsimCrowd dataset are used only for testing purposes and not for training.

The crowd detection performance of all methods was evaluated using the commonly adopted Intersection-over-Union (IoU) metric:

$$IoU = \frac{TP}{TP + FP + FN}, \tag{6}$$

where TP, FP and FN are the number of true positives, false positives and false negatives at pixel level, respectively. In addition, inference speed is measured both in ms and FPS.

B. Evaluation procedure

In all experimental sessions, all neural models were trained using the DroneCrowd train set. The proposed crowd detection method is compared to the baseline methods of [13], [15], [21]. The model of [13] consists of a simple FCN, which was trained by first extracting 128 $\times$ 128 pixel image patches depicting both crowd and non-crowd and subsequently training the FCN as a binary classifier, similarly to [13]. During inference, the trained model is able to predict crowd heatmaps by assigning crowd probabilities to each 128 $\times$ 128 pixel patch of the test image. Two variants of the method are used: a) the vanilla version from [13], denoted by FCN_t, where the 2D crowd regions are obtained by thresholding the predicted crowd heatmap, resulting in a binary crowd map, and b) a variant slightly improved by us and denoted by FCN_p, containing a final post-processing step to further refine the detected crowd regions. The post-processing step consists in convolving the obtained binary crowd map with a Gaussian kernel, in order to fill potential gaps in the binary maps. Moreover, a state-of-the-art crowd analysis network using a simple encoder-decoder architecture

TABLE I

SPEED COMPARISON OF THE PROPOSED METHOD AGAINST FCN [13], FCN_t [13], FCN_p [13], $CSRNet$ [15] AND $BiSeNet$ [21] AT VARIOUS INPUT IMAGE RESOLUTIONS (M×N).

	GTX 1080 Ti						**Jetson Xavier**					
	640×360		1280×720		1920×1080		640×360		1280×720		1920×1080	
	ms	*FPS*	*ms*	*FPS*	*ms*	*FPS*	*ms*	*FPS*	*ms*	*FPS*	*ms*	*FPS*
FCN [13]	7.3	140	20.84	48	47.11	21.2	77.12	13	181.21	5.5	234.57	4.3
FCN_t ⋆ [13]	10.13	98.7	32.7	30.6	77.68	12.9	98.32	10.2	228.35	4.4	297.88	3.4
FCN_p † [13]	12.9	77.5	47.65	21	118.39	8.4	126.9	7.9	315.3	3.2	516.78	1.9
$CSRNet$ ⋆ [15]	28.33	35.3	90.38	11.1	207.2	4.8	173	5.8	682.87	1.5	1523.8	0.65
$BiSeNet$ [21]	5.29	189	13.83	72.3	29.96	33.4	30.8	32.5	35.77	28	45.39	22
Proposed	8	124.9	17.54	57	36.58	27.3	40.12	24.9	46.23	21.6	59.57	16.8

⋆ Simple thresholding was applied to the network output.
† Thresholding and Gaussian blur was applied to the network output.

TABLE II

CROWD DETECTION PERFORMANCE OF BOTH *crowd* AND *non-crowd* CLASSES ON THE MANUALLY ANNOTATED DRONECROWD DATASET.

	IoU (%)			
	640 × 360		1280 × 720	
	crowd	*non-crowd*	*crowd*	*non-crowd*
FCN_t ⋆ [13]	49.46	92.21	61.81	95.02
FCN_p † [13]	50.62	92.56	64.94	95.34
$CSRNet$ ⋆ [15]	78.62	97.91	79.42	97.92
$BiSeNet$ [21]	80.57	97.96	83.51	98.13
Proposed	**85.63**	**98.51**	**85.90**	**98.75**

⋆ Simple thresholding was applied to the network output.
† Thresholding and Gaussian blur was applied to the network output.

TABLE III

CROWD DETECTION PERFORMANCE OF BOTH *crowd* AND *non-crowd* CLASSES ON THE SYNTHETIC AIRSIMCROWD DATASET.

	IoU (%)			
	640 × 360		1280 × 720	
	crowd	*non-crowd*	*crowd*	*non-crowd*
FCN_t ⋆ [13]	36.4	88.21	44.76	93.22
FCN_p † [13]	37.72	88.29	53.39	93.76
$CSRNet$ ⋆ [15]	63.65	96.2	68.29	96.53
$BiSeNet$ [21]	62.38	95.59	63.07	95.76
Proposed	**70.61**	**96.7**	**75.84**	**97.15**

⋆ Simple thresholding was applied to the network output.
† Thresholding and Gaussian blur was applied to the network output.

with dilated convolutions [33], i.e., $CSRNet$ [15], was also adapted to our case and trained to predict grayscale segmentation images instead of crowd density maps (since we only care for detecting crowds and not counting them). During testing, similarly to FCN_t, the final 2D crowd regions are obtained by simply thresholding the predicted output maps. Finally, the proposed method is compared against the semantic segmentation network $BiSeNet$ [21] with a ResNet-18 as backbone, which was trained to directly predict crowd segmentation maps. Note that in all experiments, the best performing post-processing hyperparameters (threshold value and Gaussian kernel size) were selected for FCN_t, $CSRNet$ and FCN_p.

The proposed network was simultaneously trained for both crowd segmentation and image-to-image translation tasks using Eq. (5), up to 200 epochs. The Adam [34] optimizer was used with batch size 16 and initial learning rate 0.001, which is reduced in each epoch using the "poly" learning rate strategy [21]. Similar to $BiSeNet$ and $CSRNet$, our backbone network (ResNet-18) is pretrained on ImageNet [35], while λ in (5) was empirically set to 0.7, a value most beneficial for crowd detection. Moreover, the train set images were augmented online during training using random scale, cropping and horizontal flipping.

Experiments were performed for several input resolutions to demonstrate the performance-speed ratio offered by all competing methods. Typically, higher input resolutions facilitate crowd detection as people are more distinguishable in the image. However, in these cases, inference speed can be considerably decreased, especially when embedded hardware is used.

Detailed inference speed comparisons were made across all competing models (Proposed, FCN_t, FCN_p, $CSRNet$ and $BiSeNet$), due to the crucial importance of fast (ideally real-time) execution in robotics applications. Notably, if human crowd areas are identified at a slow rate when the UAV is flying at a high speed, regions that need to be avoided may easily be missed, thus raising important safety concerns. Results in terms of inference speed (in msec) and FPS are presented in Table I. We also report the crowd heatmap prediction speed of [13] (without either thresholding or Gaussian blur) (FCN) to evaluate purely the network's forward pass speed. However, in this case, accurately delineated 2D crowd regions can not be obtained. Moreover, experiments using both a *Nvidia GTX 1080 Ti* GPU and a *Nvidia Jetson Xavier* embedded AI computing board were conducted, in case the crowd detection algorithm/model is running on a ground computer or on-board the UAV hardware, respectively. Different input image sizes of 640 × 360, 1280 × 720 and 1920 × 1080 pixels ($M \times N$) were used to test inference speed at low, medium and high input resolution, respectively. The results indicate that the proposed method runs significantly faster than FCN, FCN_t, FCN_p and $CSRNet$, even achieving double the speed, or faster, for the highest resolution. When compared to the $BiSeNet$ baseline, the proposed network architecture is slower only by 7.5 FPS in the worst-case embedded-execution scenario (640 × 360 input resolution on a *Nvidia Jetson Xavier*). This is not critical, as running speed remains real-time[3].

In order to evaluate the crowd detection performance of the proposed and all competing methods in real-world aerial

[3]We assume 25 FPS to be the real-time execution barrier, since this is a typical camera filming rate.

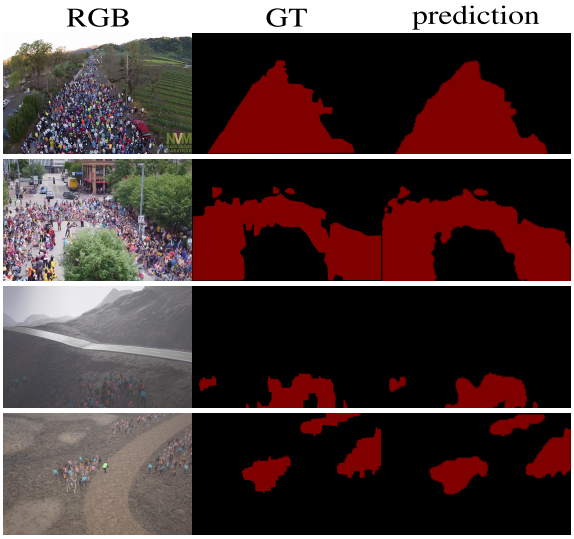

Fig. 3. Crowd detection results of the proposed method for real test crowd images from DroneCrowd dataset (rows 1 and 2) and unseen synthetic crowd images from AirSimCrowd dataset (rows 3 and 4). Each row depicts a triplet of corresponding input, ground-truth segmentation and predicted segmentation images.

images, the DroneCrowd test set was used. The IoU for both *crowd* and *non-crowd* classes of all models are reported in Table II. Results are reported at 640 $\times$ 360 and 1280 $\times$ 720 input resolution (by training and testing all models accordingly), where the running speed of the proposed method is over 20 FPS on a *Nvidia Jetson Xavier*, simulating a real-world autonomous UAV flight scenario. As shown in Table II, the proposed method significantly outperforms FCN_t and FCN_p, improving *crowd* class IoU up to 36% and 24% at low and medium input resolution, respectively. Moreover, the proposed method increased crowd detection performance, when compared to $CSRNet$ and $BiSeNet$ baselines, by a margin of up to 7% and 5%, respectively. Apart from increased detection performance of the *crowd* class, the proposed method was able to detect *non-crowd* regions more efficiently too. This is also important for autonomous UAV flight, as unnecessary actions or dangerous maneuvers can be avoided.

In real-world applications, however, the UAVs might operate in scenes that highly differ from the ones depicted in the train dataset, rendering generalization a necessary crowd detection model feature. In order to evaluate the generalization ability of all competing methods, the corresponding models that were trained on the DroneCrowd train set were tested on the AirSimCrowd set images. The IoU results of all models for both classes (*crowd*, *non-crowd*) at low and medium input resolution are presented in Table III. The proposed method demonstrates the highest generalization ability, increasing the crowd detection performance up to 34%, 12% and 7%, when compared to the models of [13] (FCN_t, FCN_p), $BiSeNet$ and $CSRNet$, respectively.

As shown in Tables I - III, the proposed method manages to outperform all competing methods both in crowd detection accuracy and in generalization, without sacrificing execution speed. This is due to its parallel network architecture, which

Fig. 4. Example video frame of a video demonstrating the proposed crowd detection method results on a previously unseen UAV-captured video. URL: `http://bit.ly/crowd_det_results`.

simultaneously saves computational cost and allows multi-task training and information exchange between the two neural branches, resulting in richer feature maps for crowd detection when compared to typical network architectures (FCN, $CSRNet$, $BiSeNet$).

Apart from the crowd detection performance reported in Tables I - III, a qualitative evaluation of the proposed model was also performed. The crowd detection results of the proposed method can be seen in Fig. 3, where random DroneCrowd and AirSimCrowd test images are depicted along with their corresponding ground-truth and predicted 2D crowd regions. The proposed method accurately predicts human crowds in 2D pixel space, with negligible false negatives and false positives, both for real and synthetic test images. In addition, the crowd detection results of the proposed method on a previously unseen, real UAV-captured video can be found in the following URL: `http://bit.ly/crowd_det_results`. An example video frame is depicted in Fig. 4. The proposed crowd detection method successfully detects crowd regions, even though its input is a completely unknown scene.

IV. Conclusion

In this paper, a deep neural network-based human crowd detection method from UAV video feed was presented. It is based on transforming the crowd detection problem into a semantic segmentation one. To this end, a novel neural architecture is proposed that combines a CNN-based real-time semantic image segmentation network branch with an image-to-image translation neural branch. The two neural pathways share the same feature extraction CNN and are jointly trained using a novel multi-task loss function that considers both tasks. Additionally, skip synapses were added between neurons of the two branches, allowing semantic information to flow from the I2I branch to the segmentation branch during inference. The proposed crowd detection method was evaluated using two newly introduced aerial crowd detection image datasets, DroneCrowd and AirSimCrowd. The proposed method significantly outperformed all competing methods, while running at real-time and near-real-time speeds on a ground computer and on an embedded AI system, respectively.

REFERENCES

[1] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous UAV surveillance in complex urban environments," in *Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 2009.

[2] F. Nex and F. Remondino, "UAV for 3D mapping applications: a review," *Applied Geomatics*, vol. 6, no. 1, pp. 1–15, 2014.

[3] P. Doherty and P. Rudol, "A UAV search and rescue scenario with human body detection and geolocalization," in *Proceedings of the Australasian Joint Conference on Artificial Intelligence (AJCAI)*, 2007.

[4] I. Mademlis, V. Mygdalis, N. Nikolaidis, M. Montagnuolo, F. Negro, A. Messina, and I. Pitas, "High-level multiple-UAV cinematography tools for covering outdoor events," *IEEE Transactions on Broadcasting*, vol. 65, no. 3, pp. 627–635, 2019.

[5] I. Mademlis, N. Nikolaidis, A. Tefas, I. Pitas, T. Wagner, and A. Messina, "Autonomous UAV cinematography: A tutorial and a formalized shot type taxonomy," *ACM Computing Surveys*, vol. 52, no. 5, pp. 105, 2019.

[6] I. Karakostas, I. Mademlis, N. Nikolaidis, and I. Pitas, "UAV cinematography constraints imposed by visual target tracking," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2018.

[7] I. Karakostas, I. Mademlis, N. Nikolaidis, and I. Pitas, "Shot type feasibility in autonomous UAV cinematography," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[8] I. Karakostas, I. Mademlis, N. Nikolaidis, and I. Pitas, "Shot type constraints in UAV cinematography for autonomous target tracking," *Information Sciences*, vol. 506, pp. 273–294, 2020.

[9] B. Sirmacek and P. Reinartz, "Automatic crowd analysis from airborne images," in *Proceedings of the IEEE International Conference on Recent Advances in Space Technologies (RAST)*, 2011.

[10] M. Küchhold, M. Simon, V. Eiselein, and T. Sikora, "Scale-adaptive real-time crowd detection and counting for drone images," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2018.

[11] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[12] K. Kang and X. Wang, "Fully convolutional neural networks for crowd segmentation," *arXiv preprint arXiv:1411.4464*, 2014.

[13] M. Tzelepi and A. Tefas, "Graph embedded convolutional neural networks in human crowd detection for drone flight safety," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–14, 2019.

[14] M. Tzelepi and A. Tefas, "Human crowd detection for drone flight safety using convolutional neural networks," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2017.

[15] Y. Li, X. Zhang and D. Chen, "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[17] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[18] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] J. Wan and A. Chan, "Adaptive density map generation for crowd counting," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[20] L. Boominathan, S. Kruthiventi, and V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in *Proceedings of the ACM International Conference on Multimedia (ACMMM)*, 2016.

[21] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[22] P. Isola, J. Zhu, T. Zhou, and A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[23] X. Liu, J. van de Weijer, and A. Bagdanov, "Leveraging unlabeled data for crowd counting by learning to rank," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[24] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[27] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[28] G. Castellano, C. Castiello, C. Mencar, and G. Vessio, "Crowd detection for drone safe landing through fully-convolutional neural networks," in *Proceedings of the International Conference on Current Trends in Theory and Practice of Informatics (SOFSEM)*, 2020.

[29] M. K. Lim, V. J. Kok, C. C. Loy, and C. S. Chan, "Crowd saliency detection via global similarity structure," in *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, 2014.

[30] S. Shah, D. Day, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, pp. 621–635, 2018.

[31] E. Kakaletsis, M. Tzelepi, P. Kaplanoglou, C. Symeonidis, N. Nikolaidis, A. Tefas, and I. Pitas, "Semantic map annotation through UAV video analysis using deep learning models in ROS," in *Proceedings of the International Conference on Multimedia Modeling (MMM)*, 2019.

[32] C. Papaioannidis, V. Mygdalis, and I. Pitas, "Domain-translated 3D object pose estimation," *IEEE Transactions on Image Processing*, vol. 29, pp. 9279–9291, 2020.

[33] F. Yu, and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[34] D. Kingma, and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[35] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[36] W. Kentaro, "Labelme: Image polygonal annotation with Python," `https://github.com/wkentaro/labelme`, 2016.

[37] P. Zhu, L. Wen, X. Bian, H. Ling and Q. Hu, "Vision meets drones: A challenge," *arXiv preprint arXiv:1804.07437*, 2018.

[38] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.