Crowd Counting from Unmanned Aerial Vehicles with Fully-Convolutional Neural Networks

Giovanna Castellano
Dept. of Computer Science
University of Bari
Bari, Italy
giovanna.castellano@uniba.it

Ciro Castiello
Dept. of Computer Science
University of Bari
Bari, Italy
ciro.castiello@uniba.it

Corrado Mencar
Dept. of Computer Science
University of Bari
Bari, Italy
corrado.mencar@uniba.it

Gennaro Vessio
Dept. of Computer Science
University of Bari
Bari, Italy
gennaro.vessio@uniba.it

***Abstract*—Crowd analysis is receiving an increasing attention in the last years because of its social and public safety implications. One of the building blocks of crowd analysis is crowd counting and the associated crowd density estimation. Several commercially available drones are equipped with on-board cameras and embed powerful GPUs, making them an excellent platform for real-time crowd counting tools. This paper proposes a light-weight and fast fully-convolutional neural network to learn a regression model for crowd counting in images acquired from drones. A robust model is derived by training the network from scratch on a subset of the very challenging VisDrone dataset, which is characterized by a high variety of locations, environments, perspectives and lighting conditions. The derived model achieves an MAE of** 8.86 **and an RMSE of** 15.07 **on the test images, outperforming models developed by state-of-the-art light-weight architectures, that are MobileNetV2 and YOLOv3.**

***Index Terms*—unmanned aerial vehicles, crowd counting, computer vision, convolutional neural networks**

I. Introduction

Crowd analysis is by nature an interdisciplinary research topic which in the last years has been drawing the increasing attention of sociologists and psychologists, as well as engineers and computer scientists [1]. The exponential increase of world population and the growing urbanization, in fact, have led to a higher incidence of unusual concentrations of people. They are due to a number of reasons, including social activities such as sport events and political rallies. Critical applications of crowd analysis include: video-surveillance for security purposes; overcrowding detection for disaster management; public safety design and traffic monitoring; simulation studies for a better understanding of crowd phenomena; and so on (e.g., [2]–[5]).

Crowd counting and its associated crowd density estimation are among the most crucial crowd analysis related tasks [1]. Crowd counting refers to the task of counting the number of people in the scene; whereas, crowd density estimation refers to the prediction of the corresponding density map. These are fundamental tasks for the application of any subsequent processing pipeline. Over the last years, researchers have addressed these issues by applying pattern recognition and computer vision strategies. Such problems are difficult because of several challenges posed by crowded images: non-uniform distribution of people; variable lighting conditions; heavy occlusion; etc. While several attempts were made with models based on hand-crafted features (e.g., [6]–[8]), the recent advancements in Convolutional Neural Network (CNN)-based methods have led to improved performance, thanks to their ability to approximate complex nonlinear relationships and to learn automatically meaningful representations from the low-level pixel features (e.g., [9]–[11]). Several successful applications of CNNs, in fact, based on images acquired from traditional cameras, have been reported in the literature (e.g., [12]–[14]).

An alternative way to acquire crowd images is to use unmanned aerial vehicles (UAVs), most commonly known as drones. They are increasingly used for crowd analysis because of their fast, real-time and low-cost image acquisition capability [15]. Several commercially available drones, in fact, are equipped with on-board cameras and inexpensive, yet powerful embedded GPUs, which make them excellent platforms for decision making tools. In addition, the acquired images can be geo-referenced using positioning sensors, such as GPS, and can be transmitted to base stations, for example via wireless, for further processing [16], [17].

However, additional difficulties must be faced when dealing with crowd counting in images captured from drones [18]. On one hand, the computer vision algorithms applied to aerial images are burdened with further difficulties, because scale and perspective issues are taken to an extreme. On the other hand, the methods commonly applied in this field, which are sophisticated and computational intensive, do not meet the strict computational requirements imposed by the UAV.

In order to address these issues, the present paper proposes a light-weight fully-convolutional neural network (FCN) model for crowd counting in images captured from drones. The model is used as a regressor for estimating the global count of the crowd, starting from an aerial image. Although it is generally recognized that regression-based methods suffer from the limited size and variance of currently available datasets [1], in our work the model is made more robust by using the recently published VisDrone dataset [18]. VisDrone is a very large benchmark database which collects images

This work was supported by the Italian Ministry of Education, University and Research within the RPASInAir Project under Grant PON ARS01_00820.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

covering a wide spectrum of locations, environments, objects and density, in different scenarios and under various weather and lighting conditions. We show that the proposed method outperforms the popular MobileNetV2 architecture [19], which is tailored for mobile and embedded applications. Moreover, the proposed model is able to outperform, on the same dataset, the well-known YOLOv3 model for object detection [20], which can be used for crowd counting as people detector. Finally, the proposed method can be used to output heatmaps that semantically enrich the flight maps. These heatmaps may be used for further tasks, such as detection of crowd areas for autonomous landing.

The rest of the paper is structured as follows. Section 2 discusses the related work. Section 3 presents the proposed method. Section 4 describes the data used for the present study and reports the obtained results. Section 5 concludes the work.

II. Related Work

Most of the early attempts to perform crowd counting were made with detection-based methods, where sliding window detectors were used to detect people in the scenes. Various learning approaches based on hand-crafted features were experimented to this purpose (e.g., [21], [22]). While these approaches provided successful results on low dense crowds, they proved to be ineffective in the presence of highly dense crowds. To address this issue, research started to focus on regression-based methods, aimed at learning a direct mapping between the features extracted from the input images and their global people count (e.g., [23], [24]). Although the use of a regressor makes the approach independent of the precise localization of the individuals in the crowd, which is a very complex task, it ignores spatial information which can be indeed very useful for the prediction task. To avoid the difficulty of detecting and precisely localizing people in the scene, several works (e.g., [25], [26]) proposed to learn object density maps, thus incorporating spatial information directly within the learning process.

In the last years, motivated by the unprecedented success of CNN-based methods in a number of learning tasks, researchers began to use this methodology for the purposes of crowd counting and crowd density estimation. Zhang et al. [14] introduced an iterative switching process where the density estimation and the count estimation tasks are alternately optimized, through backpropagation: in this way, the two related tasks help each other and are able to achieve a lower loss. Moreover, since a model trained on a specific scene can have difficulties when used in other scenes, the authors proposed a data-driven method to select samples from the training set to fine-tune the pre-trained CNN: the model is thus more apt to the unseen target scenes it is asked to estimate. The proposed crowd CNN model outperformed classic approaches based on hand-crafted features on a challenging dataset. In [12], Boominathan et al. proposed CrowdNet: a deep CNN-based framework for estimating crowd density from images of highly dense crowds (more than one thousand people). Highly dense crowds typically suffer from severe occlusion and are characterized by non-uniform scaling: for instance, an individual near the camera is captured in great detail, while an individual away from the camera could be represented as a head blob. To address this issue, CrowdNet uses a combination of a shallow and a deep architecture which simultaneously operate at a high semantic level, i.e. face detection, and at the head blob low-level. Moreover, the model is made robust to scale variations by using a data augmentation technique based on patches cropped from a multi-scale pyramidal representation of each training image. In [13], Sindagi et al. presented an end-to-end cascaded CNN that jointly learns the crowd density map and a high-level global prior which is conceived to aid the prediction of density maps from images with large variations in scale and appearance. The high-level prior consists in a crowd count classification, where crowds are categorized in several groups depending on the people count. Unfortunately, these works did not consider aerial images taken from drones. Correspondingly, they proposed complex methods which can be too expensive for the real-time and computational requirements of the application deployed on UAVs.

Only a few recent works have focused their attention specifically on drones. In [27], motivated by the inability of regression counters to generate precise object positions, Hsieh et al. proposed to inject spatial layout information into the deep network model to improve localization accuracy. To evaluate the effectiveness of their method, the authors addressed the problem of car counting in images of car parkings. However, having precise spatial layout information is an assumption which is not met in the non-uniform and casual human crowded scenes.

In [28], Liu et al. showed that providing a deep network with an explicit model of perspective distortions effects, along with enforcing physics-based spatio-temporal constraints, can improve prediction performance on video frames acquired from moving drones. To this end, they fed the network not only with the original image but also with an identically-sized image which contains the local scale as a function of the camera orientation with respect to the ground plane. In addition, they imposed temporal consistency by forcing the densities in consecutive images to correspond to physically possible people flows. Conversely, in contrast to the recent trend, Küchold et al. [29] used features based on the luminance channel and kernel density estimation, showing that this approach can be faster and more accurate than a CNN-based method. Unfortunately, both methods were tested on very little data, thus more experiments are needed to estimate their generalization capability.

Finally, a different way to look at the problem of crowd analysis from drones is crowd detection. In [30] and [31], Tzelepi and Tefas adapted a pre-trained CNN model by discarding the fully-connected higher layers in favor of an extra convolutional layer, making it an FCN: this approach was conceived to reduce the parameters to be learnt and thus the computational cost. Due to the lack of available datasets, experiments were performed on the *Crowd-Drone*

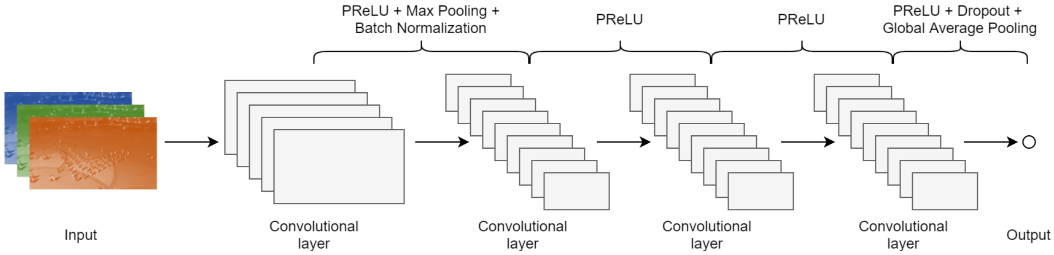

Fig. 1. Proposed FCN architecture.

dataset, purposely designed by the authors. Specifically, the dataset was created by querying YouTube using keywords describing crowded scenes captured from drones (e.g., festival, parade, political rally, etc.). The proposed approach achieved successful results in the binary discrimination crowded vs. non-crowded scenes. This approach is suitable for several applications as it is able to output heatmaps that semantically enrich the flight maps, for example by defining "fly" and "no-fly" zones for autonomous landing. Each heatmap is obtained by feeding the network with the corresponding image labeled as "crowd" and by extracting the feature map of the last convolutional layer. Unfortunately, *Crowd-Drone* is not provided with annotations of people count. Inspired by the work of Tzelepi and Tefas, in [32] we have proposed a crowd detector for drone-captured images based on an FCN trained on a subset of the very challenging VisDrone dataset. The proposed method is based on a two-loss model in which the main classification task, aimed at distinguishing between crowded and noncrowded scenes, is simultaneously assisted by a regression task, aimed at people counting. In [33], we improved upon the proposed model by replacing the auxiliary loss based on crowd counting with a loss based on the agglomeration tendency of the crowd.

III. Proposed Method

In the context of crowd analysis from aerial images, a light-weight model is required to meet the computational limitations imposed by the UAVs' hardware. To this end, we propose an FCN architecture. Relying only on convolutional layers for feature extraction reduces considerably the amounts of parameters to be learnt, as the fully connected layers typically stacked on top of the convolutional base contribute the most to the overall computational cost. Another advantage is that the network can be fed with images of arbitrary dimensions, as only the fully connected layers expect inputs having a fixed size. Finally, the convolutional layers preserve the spatial information which is destroyed by the fully connected layers, because of their connection to all input neurons.

The proposed FCN architecture is depicted in Fig. 1. To speed up calculation, without sacrificing too much capacity, the input to our model are 128×128 three-channel images, normalized in the range $[0, 1]$ before training. Each image is then propagated through a convolutional layer having 32 filters, with kernel size 5×5 and stride 1. This configuration is intended to preserve the initial input information. The first convolutional layer is followed by a Parametric ReLU (PReLU) non-linearity, in which a parameter a is adaptively learned during backpropagation to avoid zero gradient when a unit is not active:

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ ax & \text{otherwise,} \end{cases}$$

where x is the input to a neuron. This modification can slightly improve performance in large datasets [34]. Then, the output of PReLU is down-sampled by a max pooling layer, which divides each spatial dimension by a factor of 2. The output of the max pooling layer is propagated through a batch normalization layer, with momentum of 0.99 and ϵ of 0.001, to aid generalization [35]. Next, three consecutive convolutional layers follow, each having 64 filters with kernel size 3×3. The number of filters in these layers is higher mainly because the number of low level features (i.e., circles, edges, lines, etc.) is typically low, but the number of ways to combine them to obtain higher level features can be high. Each of the three convolutional layers is followed by a PReLU activation. These layers are not interleaved by pooling layers, which would further reduce the resolution of the feature maps, thus preventing the network from learning globally relevant and discriminating features from images that are characterized by large variations of scale. Finally, the output layer is preceded by a dropout layer with dropout rate of 50%, which is introduced to mitigate overfitting, and by a global average pooling layer, which calculates the average of each feature map in the previous layer and thus reduces considerably the number of features to be used for the final regression. We conceived this architecture mainly because it provides a very light-weight model which meets the strict computational requirements of the UAV; moreover, it is complex enough to avoid underfitting the data.

In order to predict the global count of the crowd, the FCN model is asked to minimize a classic mean absolute error loss function:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left| y_i - h_\theta(x_i) \right| ,$$

where N is the number of training examples, while y_i and $h_\theta(x_i)$ represent the actual and predicted crowd count, respectively. We preferred this loss over other regression loss functions, because of its precise physical meaning.

Finally, it is worth noting that the last convolutional layer can be used to obtain heatmaps for semantically enriching the flight maps. To do this, inspired by the class activation map method described in [36], we propose to use a regression activation map (RAM) which is essentially a weighted sum of the feature maps in the last convolutional layer. This layer retains the last available information maintaining a correspondence with a given original input image, before the drastic reduction caused by the global average pooling. More formally, let $A^k \in \mathbb{R}^{u \times v}$ be the k-th feature map from the last convolutional layer, being u and v its height and width. The information in these feature maps can be used to localize the "most active" regions in the original image with respect to the final regression prediction y'. A summary of the overall feature maps, i.e. a regression activation map L_{RAM}, can be obtained as a linear combination, followed by a ReLU:

$$L_{RAM} = ReLU\left(\sum_k \alpha_k A^k\right).$$

Since some feature maps would be more important than others to make the final decision, as in [36] we propose to use the averaging pooling of the gradient of y' with respect to the k-th feature map as a weight for the feature map:

$$\alpha_k = \frac{1}{uv}\sum_{i=1}^{u}\sum_{j=1}^{v}\frac{\partial y'}{\partial A_{i,j}^k}.$$

In practice, $\frac{\partial y'}{\partial A_{i,j}^k}$ measures the effect of the (i,j)-th pixel in the k-th feature map on the y' score. Differently from [36], we are interested not only in the features that have a positive influence on a certain class, but we are interested in the influence of the features on the overall regression score. Upsampling the RAM to the size of the input image enables the identification of the regions that are most relevant for the final prediction. This approach allows one to obtain a kind of crowd density map without explicitly learning it.

IV. Experiment

To evaluate the effectiveness of the proposed method in correctly estimating the crowd count, we re-arranged the VisDrone dataset, as described in the next subsection. As a baseline to compare our method against, we employed the popular MobileNetV2 architecture [19], pre-trained on ImageNet [37] and fine-tuned to our data. MobileNet is a light architecture which is well suited to mobile and embedded computer vision applications [38]. This architecture introduced the so-called depthwise separable convolutions, which perform a single convolution over each colour channel rather than combining all of them. This significantly reduces the numbers of parameters to be learned. MobileNetV2 still uses depthwise separable convolutions as efficient building blocks; however, it introduces linear bottlenecks between layers and shortcut connections between bottlenecks to improve the efficacy and effectiveness of the network. To perform transfer learning on the VisDrone dataset, we used the common practice to remove the top level classifier, which is very specific for the original classification problem, and to stack a custom layer to be trained on our task.

Similarly, since the task of counting people can be accomplished also through object/pedestrian detection, we fairly compared our method to the well-known YOLOv3 object detector [20]. The "You Only Look Once" (YOLO) family is a family of models designed for fast object detection [39]. The approach involves a single deep convolutional network that splits the input image into a grid of cells, where each cell is responsible for predicting the bounding box and object category of the object it contains. The resulting outcome is a number of candidate boxes which are consolidated into a final detection through a following non-maximum suppression. There are three main variations of the originally proposed architecture: the third version is currently the last one. In contrast to MobileNetV2, we used a version of YOLOv3 pre-trained on the large-scale MS COCO object detection dataset [40]. Clearly, to speed up calculations, we forced the network to detect only persons, ignoring the other existing categorizations, such as animals and vehicles.

A. Dataset Preparation

Developing a large crowd dataset from a drone perspective is a very time consuming and expensive process. To overcome this issue, we used an adaptation of the VisDrone benchmark dataset,[1] collected by the AISKYEYE team at the Laboratory of Machine Learning and Data Mining, Tianjin University, China. The data have been used for the VisDrone 2018 and 2019 challenge. To date, VisDrone is the largest dataset of aerial images from drones ever published.

The original dataset consists of 288 video clips, with $261,908$ frames and $10,209$ additional static images: they were acquired by various drone platforms, across 14 different cities separated by thousands of kilometers in China [18]. The captured scenes cover various weather and lighting conditions, environment (urban and country), objects (pedestrians, vehicles, etc.) and density (sparse and crowded scenes). The maximum resolutions of video clips and static images are 3840×2160 and 2000×1500, respectively. Sample images are shown in Fig. 2. Frames and images were manually annotated with more than 2.6 million bounding boxes of targets. The manually annotated ground truth is available only for the training and validation sets, but not for the test sets to avoid the overfitting of the algorithms proposed by the challenge participants. The object categories involve human and vehicles of the daily life: pedestrians, persons, cars, vans, buses, and so on. If an individual maintains a standing pose or is walking, it is classified as a pedestrian, otherwise as a person. For our purposes, we considered both pedestrians and persons as a unique people category.

The benchmark data embedded in VisDrone have been originally conceived to tackle different kinds of tasks, ranging from object detection in images/videos to single or multi-object tracking. Since object categories are not provided for the

[1] http://aiskyeye.com

Fig. 2. Sample images from VisDrone.

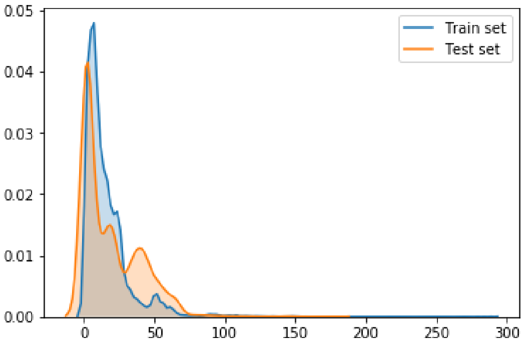

Fig. 3. Distributions of the crowd counts in the training and test set estimated with a kernel density estimation. For a better visualization, we removed from the training set a single image with a count of 888.

tracking tasks, we used only the data for the object detection tasks. Then, starting from the provided annotations of pedestrians and persons, in particular their count, we developed our own crowd dataset: it is composed by $30,672$ images as training set and $3,394$ images as test set. We used the challenge training sets as our training data, while the validation sets formed our test data. The distributions of the crowd counts in the training and test set are illustrated in Fig. 3. As it can be seen, VisDrone is characterized by a prevalence of sparse scenes with few tens of people. Artificially augmenting the dataset with overcrowded scenes calls for future research.

B. Implementation Details

Experiments were run on an Intel Core i5 equipped with the NVIDIA GeForce MX110, with dedicated memory of 2GB. As deep learning framework, we used TensorFlow 2.0 and the Keras API.

The proposed FCN model was trained from scratch by performing stochastic gradient descent with randomly sampled mini-batches of 64 images, learning rate of 10^{-5} and momentum of 0.9. As previously mentioned, to reduce the computational cost the input images were resized to 128×128 and normalized within the range $[0, 1]$.

In order to assess the effectiveness of our model, we made a comparison with state-of-the-art approaches: MobileNetV2 and YOLOv3. Concerning the MobileNetV2 model, we used a low learning rate of 10^{-4} in order to prevent the weights previously learned on ImageNet from being destroyed. Moreover, it is worth remarking that we used larger input images of shape 224×224, so as to address the higher capacity of the network, and each input channel was re-scaled to the range $[-1, 1]$, as this is the input expected by the network. Concerning YOLOv3, the network expects inputs with a square shape of 416×416 and pixel values scaled between 0 and 1. As for any object detection system, we needed to set thresholds for the confidence score, i.e. the confidence for a bounding box to accurately describe an object, and the amount of overlap between bounding boxes referring to the same objects under which they are filtered out during non-maximum suppression. We used a confidence score of 0.50 and an overlapping threshold of 0.50: these values are typically used in pedestrian detection as they represent a good compromise between the precision and recall of the detections.

For all the models we employed, we used early stopping with patience of 1 to avoid over-training. This technique was applied by monitoring the loss value on a validation set randomly held out as a fraction of 10% of the training set. As for the training time, each model was trained for few tens of epochs before reaching convergence, requiring a few hours of time. This was expected, as VisDrone is characterized by high variance, thus a model starts overfit soon.

C. Experimental Results

Experimental results are provided in Table I. For the purposes of the evaluation, we used standard metrics used by many existing methods for crowd counting: mean absolute error (MAE) and root mean squared error (RSME). Analogously to the loss function we used, MAE is the average absolute difference between the ground truth and the predicted count for all test scenes:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - y'_i| \,,$$

where N is the number of test examples, while y_i and y'_i are the actual and estimated crowd count, respectively. Similarly, RMSE is the square root of the averaged squared difference between the ground truth and the predicted count over all test scenes:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2},$$

where N, y_i and y_i' have the same meaning as before. The main difference between the two metrics is that RMSE is more sensitive to large errors. In addition, we also provide measures of size (MB in HDF5 format) and speed (frames per second) of the experimented models.

As it can be observed, the worst results both in terms of efficacy and efficiency have been obtained by YOLOv3. The worst results in terms of prediction accuracy were expected, since it is well-known the difficulty of running object detection algorithms on the very challenging VisDrone dataset [41]. It is worth noting that, for each scene, we did not consider only the true positive detections, but also the false positive ones, as their sum represents the overall number of people the model "believed" were in the scene. Also the lower recognition speed was expected, as YOLOv3 was asked to predict not only the presence of people, but also to localize precisely their position by estimating the corresponding bounding boxes. Better results were obtained by MobileNetV2 which drastically reduced the size of the predictive model, while improving speed. This finding confirms that a regression-based method may be preferred over an object detector for the task of crowd counting. Finally, the overall best results were achieved with the proposed FCN model trained from scratch. The better results with respect to MobileNetV2 can be explained considering that the proposed FCN has lower capacity, thus it may have suffered less from overfitting. In addition, it should be considered that, although MobileNetV2 was fine-tuned to VisDrone, the ImageNet dataset the model was originally trained on is characterized by a number of scenes which are very different from aerial images captured from drones. In other words, a perspective problem arises.

Unfortunately, the averaged results provided by MSE and RMSE do not allow the evaluation of the model behaviour depending on the sparseness or on the crowdedness of the people in the scenes. However, to perform a finer evaluation, it is worth to note that the test images can be divided into categories and then the model can be evaluated by reformulating the task as a classic classification problem. In particular, we made two distinct evaluations. In the first one, test images were divided into two classes: sparse scenes, with less than 10 people in the scene; and crowded scenes, with more than 10 people in the scene. In the second one, we considered three classes, in which the "crowd" class was further divided depending on whether the number of people exceeded 30 individuals. Accordingly, we adjusted the actual and predicted counts as category labels. In this way, we were able to measure standard classification metrics such a precision and recall. Precision is calculated as follows:

$$Precision = \frac{TP}{TP + FP},$$

TABLE I
CROWD COUNTING RESULTS.

Model	MSE	RMSE	Size (MB)	Speed (fps)
MobileNetV2	10.84	15.40	$\sim$ 16.5	53.87
YOLOv3	12.14	19.12	$\sim$ 242.8	3.4
Proposed FCN	8.86	15.07	$\sim$ 10.3	57.52

TABLE II
TWO-CLASS CROWD CLASSIFICATION RESULTS OF THE PROPOSED FCN.

Class	Precision	Recall
< 10	0.89	0.80
≥ 10	0.83	0.91
Average	0.86	0.85

TABLE III
THREE-CLASS CROWD CLASSIFICATION RESULTS OF THE PROPOSED FCN.

Class	Precision	Recall
< 10	0.89	0.80
≥ 10 and < 30	0.46	0.86
≥ 30	1.00	0.48
Average	0.78	0.71

where TP and FP stand for the number of true positives and false positives, respectively. Intuitively, precision is the ability of the model not to label as positive a sample that is negative. Similarly, recall is calculated as the following ratio:

$$Recall = \frac{TP}{TP + FN},$$

where FN is the number of false negatives. Intuitively, recall is the ability of the model to find all the positive instances.

As it can be seen in Table II, which reports the results of the proposed FCN model in discriminating between sparse (less then 10 people) and crowded (more than 10 people) scenes, the model was, on average, pretty good in correctly detecting the presence or absence of a crowd in the test images. In the more refined evaluation based on three classes (see Table III), we found that the model had difficulties in finding all the highly dense crowds (more than 30 people), which were often mistakenly categorized as low dense crowds (i.e., with a number of individuals between 10 and 30).

Finally, from a qualitative point of view, the proposed method can be used to output regression activation maps that can semantically augment the flight maps. Examples of test images and corresponding heatmaps are depicted in Fig. 4. It can be seen that the model was able to some extent to distinguish the zones where people were in the scene from the areas with no people. This discriminating ability is desirable, as it can be beneficial to several tasks, for example during autonomous landing operations in order to prevent the drone from landing on a "risky" zone.

V. CONCLUSION

Today, unmanned aerial vehicles are increasingly used in a plethora of domains, from fast delivery to precision agriculture. With the recent breakthroughs in deep learning and

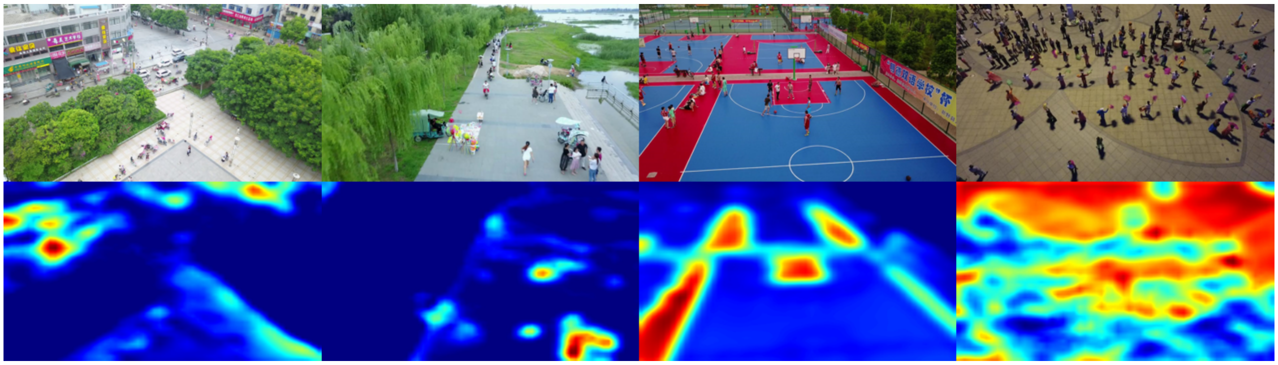

Fig. 4. On the top, four test images; on the bottom, the corresponding heatmaps provided by the method. For a better visualization, they have been re-scaled to the original proportions.

computer vision, these platforms can now be empowered with real-time and accurate decision making tools. One of the most promising applications of computer vision on aerial images from drones is crowd counting and its associated crowd density estimation. The present paper addressed this problem by proposing a light-weight fully-convolutional network regression model, which can cope with the strict computational requirements of the UAVs' hardware and can provide real-time responses. While regression methods have shown non-optimal performance because of the limited size and variability of most currently available datasets, we have proven that by relying on a sufficiently large and general dataset, they can achieve successful performance. To this end, we employed the large VisDrone benchmark dataset, characterized by a large variety of aerial scenes from drones. The proposed method is able not only to regress on the global people count, but can also provide heatmaps that can be used to semantically enrich the flight maps for several applications, e.g. for autonomous landing. These heatmaps provide a kind of crowd density maps that, in contrast to traditional approaches, are not required to be directly learned by the deep model.

The proposed method was able to provide better results than a more complex model based on the MobileNetV2 architecture. A deep network pre-trained on ImageNet can be less tailored to distinguish among aerial images, mainly because of their different perspective against traditional photographic scenes. In addition, both models outperformed the YOLOv3 state-of-the-art real-time object detector. This finding confirms that the use of a regression-based method may be preferred over the use of an object detection strategy, for the purposes of crowd counting, even if based on a CNN-based solution.

Finally, a major limitation of the proposed approach should be remarked. Probably because of the dataset's characteristics, the proposed model is biased in favor of the correct prediction of the less sparse scenes instead of the overcrowded ones. Future developments of the present research should address this issue, for example by enlarging the data at disposal with data augmentation [42] or synthetic data generation through generative adversarial networks [43]. Another future direction is to consider the video captured by the drone naturally as a data stream [44], instead of a collection of still images.

REFERENCES

[1] V. A. Sindagi and V. M. Patel, "A survey of recent advances in CNN-based single image crowd counting and density estimation," *Pattern Recognition Letters*, vol. 107, pp. 3–16, 2018.

[2] R. Chaker, Z. Al Aghbari, and I. N. Junejo, "Social network model for crowd anomaly detection and localization," *Pattern Recognition*, vol. 61, pp. 266–281, 2017.

[3] V. J. Kok, M. K. Lim, and C. S. Chan, "Crowd behavior analysis: A review where physics meets biology," *Neurocomputing*, vol. 177, pp. 342–362, 2016.

[4] A. Bianchi, S. Pizzutilo, and G. Vessio, "Applying predicate abstraction to abstract state machines," in *Enterprise, Business-Process and Information Systems Modeling*. Springer, 2015, pp. 283–292.

[5] K. Lloyd, P. L. Rosin, D. Marshall, and S. C. Moore, "Detecting violent and abnormal crowd activity using temporal analysis of grey level co-occurrence matrix (GLCM)-based texture measures," *Machine Vision and Applications*, vol. 28, no. 3-4, pp. 361–371, 2017.

[6] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2547–2554.

[7] R. Liang, Y. Zhu, and H. Wang, "Counting crowd flow based on feature points," *Neurocomputing*, vol. 133, pp. 377–384, 2014.

[8] J. Xing, H. Ai, L. Liu, and S. Lao, "Robust crowd counting using detection flow," in *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 2061–2064.

[9] Z. Deng, Z. Wang, and S. Wang, "Stochastic area pooling for generic convolutional neural network," in *Proceedings of the Twenty-second European Conference on Artificial Intelligence*. IOS Press, 2016, pp. 1760–1761.

[10] M. Diaz, M. A. Ferrer, D. Impedovo, G. Pirlo, and G. Vessio, "Dynamically enhanced static handwriting representation for Parkinson's disease detection," *Pattern Recognition Letters*, vol. 128, pp. 204–210, 2019.

[11] G. Castellano and G. Vessio, "Towards a tool for visual link retrieval and knowledge discovery in painting datasets," in *Italian Research Conference on Digital Libraries*. Springer, 2020, pp. 105–110.

[12] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in *Proceedings of the 24th ACM International Conference on Multimedia*. ACM, 2016, pp. 640–644.

[13] V. A. Sindagi and V. M. Patel, "CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017, pp. 1–6.

[14] 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*, 2015, pp. 833–841.

[15] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of unmanned aerial vehicles*. Springer, 2015.

[16] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: Opportunities and challenges,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[17] G. D’Amato, G. Avitabile, G. Coviello, and C. Talarico, “A beam steering unit for active phased-array antennas based on FPGA synthesized delay-lines and PLLs,” in *2015 International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE, 2015, pp. 1–4.

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

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[20] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[21] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, “Hough forests for object detection, tracking, and action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2188–2202, 2011.

[22] P. Viola, M. J. Jones, and D. Snow, “Detecting pedestrians using patterns of motion and appearance,” *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.

[23] A. B. Chan and N. Vasconcelos, “Bayesian poisson regression for crowd counting,” in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 545–551.

[24] K. Chen, C. C. Loy, S. Gong, and T. Xiang, “Feature mining for localised crowd counting.” in *BMVC*, vol. 1, no. 2, 2012, p. 3.

[25] V. Lempitsky and A. Zisserman, “Learning to count objects in images,” in *Advances in Neural Information Processing Systems*, 2010, pp. 1324–1332.

[26] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada, “Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3253–3261.

[27] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, “Drone-based object counting by spatially regularized regional proposal network,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4145–4153.

[28] W. Liu, K. M. Lis, M. Salzmann, and P. Fua, “Geometric and physical constraints for drone-based head plane crowd density estimation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, no. CONF. IEEE/RSJ, 2019.

[29] M. Küchhold, M. Simon, V. Eiselein, and T. Sikora, “Scale-adaptive real-time crowd detection and counting for drone images,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 943–947.

[30] M. Tzelepi and A. Tefas, “Human crowd detection for drone flight safety using convolutional neural networks,” in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 743–747.

[31] ——, “Graph embedded convolutional neural networks in human crowd detection for drone flight safety,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2019.

[32] G. Castellano, C. Castiello, C. Mencar, and G. Vessio, “Crowd detection for drone safe landing through fully-convolutional neural networks,” in *International Conference on Current Trends in Theory and Practice of Informatics*. Springer, 2020, pp. 301–312.

[33] ——, “Crowd detection in aerial images using spatial graphs and fully-convolutional neural networks,” *IEEE Access*, vol. 8, pp. 64 534–64 544, 2020.

[34] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[35] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[36] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[38] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[39] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[40] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in *European Conference on Computer Vision*. Springer, 2014, pp. 740–755.

[41] P. Zhu, D. Du, L. Wen, X. Bian, H. Ling, Q. Hu, T. Peng, J. Zheng, X. Wang, Y. Zhang *et al.*, “VisDrone-VID2019: The vision meets drone object detection in video challenge results,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[42] L. Taylor and G. Nitschke, “Improving deep learning using generic data augmentation,” *arXiv preprint arXiv:1708.06020*, 2017.

[43] A. Antoniou, A. Storkey, and H. Edwards, “Data augmentation generative adversarial networks,” *arXiv preprint arXiv:1711.04340*, 2017.

[44] G. Casalino, G. Castellano, and C. Mencar, “Data stream classification by dynamic incremental semi-supervised fuzzy clustering,” *International Journal on Artificial Intelligence Tools*, vol. 28, no. 08, p. 1960009, 2019.