IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Crowd Detection in Aerial Images using Spatial Graphs and Fully-Convolutional Neural Networks

GIOVANNA CASTELLANO, CIRO CASTIELLO, CORRADO MENCAR AND GENNARO VESSIO

Department of Computer Science, University of Bari "Aldo Moro", Bari, Italy

Corresponding author: Gennaro Vessio (e-mail: gennaro.vessio@uniba.it).

This work was supported by the Italian Ministry of Education, University and Research within the RPASInAir Project under Grant PON ARS01_00820. GC, CC, and CM are members of the INdAM Research group GNCS. The Jetson TX2 used for this research was donated by the NVIDIA Corporation.

ABSTRACT Unmanned aerial vehicles (UAVs) also known as drones are increasingly populating our skies. This represents a relevant issue both for the legislator and the researcher. While the regulation plans often assume precautionary approaches, stating restrictive conditions of use for the sake of public safety, the applied research is exploring novel strategies to develop autonomous vehicles endowed with trusty operating mechanisms. The challenge is to let drones overflying even populated areas while keeping a steady control of the situation on the ground, thus enabling the possibility of safe landing with no harm for people. This can be done employing on-board cameras and embedded GPUs which allow for the execution in real-time of computer vision applications. In this paper, we introduce a crowd detection method for drone safe landing. The pivotal points of our proposal are related to the computational limitations imposed by the currently available hardware resources of UAVs. In this sense, our method is based on the light-weight scheme of a fully-convolutional neural network which conjugates nimble computations and effectiveness. We propose a two-loss model where a classification task (oriented to distinguish between crowded/non-crowded scenes) is supported by a regression task (aimed at better focusing the agglomeration tendency of the persons). This latter job is realized by resorting to the construction of a spatial graph for each analysed image and to the evaluation of the corresponding clustering coefficient. As a further element, our model is endowed with the capability to produce class activation heatmaps which contribute to the semantic enrichment of the flight maps. We tested our model on a large dataset of aerial images and we observed how it compares favorably with other approaches proposed in literature.

INDEX TERMS Deep learning, computer vision, crowd detection, unmanned aerial vehicles, safe landing.

I. INTRODUCTION

THE last few years have been witnessing a widespread availability of Remotely Piloted Aircrafts (RPAs), also known as Unmanned Aerial Vehicles (UAVs) or drones. Their versatility and their reduced cost contributed to a boost in their commercial popularity and facilitated their application in several scenarios [1], [2]. At the same time, drones being maneuvered by an increasing number of non-professional operators call for suitable regulations. For example, in Italy RPAs are commonly forbidden from overflying "gathering of persons during parades, sports events or different forms of entertainment or [...] areas where there is an unusual concentration of people",[1] as can be read in the Italian Regulation issued in 2018 by ENAC, which is the national aviation authority of Italy. It is straightforward that this basic form of public safety implies the identification of "restricted areas", where the aircraft flight is prohibited, in accordance with certain specified conditions. However, this kind of determination cannot be welcomed as a conclusive resolution. In fact, unpredictable occurrences and states of emergency may lead to hazardous operations, including the possibility to attempt a landing in areas where crowds of people are gathered. In

[1] https://www.enac.gov.it/sites/default/files/allegati/2018-Lug/Regulation_RPAS_Issue_2_Rev_4_eng.pdf

addition, it may be useful to release the vehicles from strict prohibitions in their flight plans, while continuing to keep track of the situation of the ground below. In other words, automatic mechanisms would be useful to endow drones with the capability to distinguish between "safe" and "risky" routes, so that their flight-plans can be properly adjusted even while overflying populated zones (such as, urban areas) [3]. Such an equipment would be worthwhile for RPAs, but would prove to bring even more benefits when applied to another category of vehicles, i.e. autonomous UAVs. Those aircrafts must be able to automatically follow a flight plan and possibly to adapt it (identification of "safe" way-points on geo-referenced maps may be a principle of operation) and are currently regarded as a kind of advanced drones [4].

In order to provide the drone with a real-time, decision-making tool, an on-board intelligent system is required. Several commercially available drones are equipped with relatively cheap cameras and GPUs. The latter are powerful enough to address the problem of crowd detection from drones by using computer vision algorithms. In particular, Convolutional Neural Networks (CNNs) have been successfully applied in the realm of object recognition [5] and recently proved their effectiveness in a wide range of image classification tasks (see, for example, [6]–[9]). Nevertheless, the treatment of images shot by drones is somewhat more complex: additional difficulties (such as scale and viewpoint modifications) translate this kind of task into a real challenge [10].

When we turn to consider the specific problem related to crowd detection in images captured from UAVs, the literature panorama is scarce. On the one hand, the computational burden implied by the neural paradigm is remarkable when potentialities of drones are taken into account. On the other hand, there are not so many datasets involving images purposely captured to perform crowd detection. For those reasons, the state-of-the-art of computer vision approaches devoted to such kind of problems is not so populated. Among the few examples proposed in literature, the work by Tzelepi and Tefas [11], [12] is based on a Fully-Convolutional Network (FCN) to be implanted on drones. The FCN has been adopted as an undemanding tool which is able to analyse images and discriminate between those including people gatherings and those which don't. Also, it is employed to provide estimated heatmaps to semantically enrich the flight maps. To validate their proposal, the authors used their own annotated dataset, i.e. the *Crowd-Drone* dataset. This means that the proposed approach has not been tested on datasets of images which could stand as better test-beds both in terms of dimension and heterogeneity of depicted scenarios. On top of that, the work presented in [11], [12] confines the problem to a binary discrimination (scenes with people/scenes without people), leaving room for the unaccomplished analysis of more ambiguous cases.

All things considered, we feel that the employment of deep learning techniques for automatic detection of crowds still deserves further investigation. In this sense, the research we present in this paper stands as a contribution to the state-of-the-art on crowd detection from drones. Moving from the above observations, we founded our work on a number of key-points. Concerning the data, we considered the VisDrone dataset [10] including a great amount of aerial images taken from drones. Those shots pertain to a broader set of scenarios, which is manifold under any aspect: depicted scenes, included elements, object and people density, lighting conditions, size scales, and so on. Concerning the method, we propose a new light-weight FCN architecture and we train and evaluate different models developed by the proposed architecture. In a preliminary investigation, we experimented a couple of light-weight models: a classic cross-entropy loss model and a multi-output model. The latter implements a joint loss combining the cross-entropy to a regression loss, based on the people count [13]. In contrast to traditional approaches, where multi-output models are meant to provide different outputs from the same input, in our case the regression task is used to "assist" the classification task in order to learn more meaningful features.

However, this preliminary investigation was bounded to a simple characterization of "crowdedness", based only on people count. The present paper, besides reporting that kind of results, significantly extends our previous work by introducing a more refined characterization of "crowdedness", based on the spatial clustering tendency of the crowd. More precisely, from each input image we extract the spatial graph having people as vertexes and we derive a clustering coefficient aimed at evaluating the clustering tendency of the crowd. The rationale behind this approach is injecting additional information about what a crowd is, so that the model can learn a better mapping between images and crowded scenes. We show that this approach outperforms our previous models. Moreover, when applied to the same data, our light-weight models are collectively able to provide better results than the FCN architecture proposed in [12]. In addition, they also outperform MobileNet, which is a popular pre-trained FCN [14].

The rest of the paper is organized as follows. Section II discusses related works. Section III presents the proposed method. Section IV describes the data used for the present study and provides experimental results. Section V concludes the work.

II. RELATED WORK

To the best of our knowledge, there are not so many proposals in literature concerning the task of crowd detection from drones. In the following we expose the contributions coming from a number of works which are differently related to our study.

To a broader extent, camera devices installed on UAVs may be exploited to drive the vehicles toward safe landing. Some approaches can be reported which address the problem to spot a marker on the ground, intended to supply guidance for drone landing. To this end, Lin et al. [15] and Polvara et al. [16] proposed a classification method based on classic

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2984768, IEEE Access

hand-crafted features. Conversely, Nguyen et al. [4] used features automatically learned by a light CNN implementation (namely, lightDenseYOLO) to direct the vehicle toward the marker.

Some other approaches aim at identifying "safe" areas for possible landing. This is the case, for instance, of the work by Marcu et al. [17] where an embeddable CNN is used to estimate depth from in-flight images and segment them into "safe-landing" and "obstacle" regions. Mukadam et al. in [18] follow a more conventional approach and make use of SVM algorithms to identify suitable landing areas by analyzing color features extracted from satellite images.

When the crowd detection task is more specifically considered, the number of contributions in literature is even lesser. We already mentioned the pioneering works by Tzelepi and Tefas: in [11] they employed an FCN as a light-weight model to distinguish between crowded and non-crowded scenes captured from drones. The FCN model derives from a pre-trained CNN where the fully-connected layer has been discarded and a final convolutional layer has been added before conducting the retraining of all the convolutional layers. Also, they proposed a novel two-loss-training procedure, which aims at enhancing the separability of crowd and non-crowd classes. It is worthwhile to observe that the authors remedy the scarcity of suitable data by constructing their own dataset. It has been built retrieving videos from Youtube: some of them are related to keywords describing crowded events (e.g., parades, festivals, marathons, protests, political rallies); others non-crowded videos have been gathered by searching for unspecified drone videos. The adopted model was able to produce both relevant results in terms of classification accuracy and heatmaps for crowded areas. The latter allow for semantic annotation of the flight-maps, leading to the definition of no-fly zones. In [12] the authors engaged in the enhancement of their previous work by introducing a regularization scheme (drawing inspiration from the Graph Embedding framework), which produced a slight improvement of the obtained results.

Within the context of crowd analysis, a closely related issue is attracting growing interest due to its reverberation on a number of practical applications, that is the problem of crowd counting and crowd density estimation. In [19] a CNN is proposed to perform cross-scene crowd counting. The model is trained by a switchable learning process with two learning objectives (crowd density maps and crowd counts) which can assist each other to obtain better local optima. Since each scene has its unique properties (view angles, scales, density, etc.), the authors bridged the distribution gap between the training and test scenes through a nonparametric fine-tuning scheme which adapts the pre-trained CNN model to unseen target scenes.

Another example of deep CNN applied to the crowd counting problem can be found in [20] where CrowdNet is introduced. CrowdNet is a deep learning framework to be applied for crowd density estimation in scenarios characterized by high density of people (a few thousands of persons). The images involved in this kind of scenes pose a variety of challenges ranging from severe occlusion of single persons to the non-uniform scaling of the crowd. CrowdNet faces those problems by using a combination of deep and shallow convolutional neural networks operating at different semantic levels during the image analysis. The authors had to deal also with the limited amount of training data available: extensive data augmentation has been performed by sampling patches from the multi-scale image representation.

In [21], 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.

The aforementioned works propose methods which can be mostly regarded as too expensive when we consider the real-time requirement and the computational limits of the applications deployed on UAVs. Moreover, all of the previous proposals do not take into account the analysis of images shot from drones. Actually, scenes of that kind have been considered to tackle the crowd counting problem in [22], where features are extracted from images to compute a density map (by means of kernel density estimation). However, also in that work the presence of a crowd into the image is implicitly assumed. By contrast, in our research activity we are interested in determining the existence of a crowd for the sake of drone safe landing.

III. PROPOSED METHOD

In order to perform crowd detection in video frames acquired from drones, we propose a detection model based on a light-weight FCN. To construct such a model, i.e. to learn a mapping from each input image to the presence or absence of a crowd, a dataset of labeled images including examples of crowds is needed. Specifically, a proper characterization of the concept of "crowdedness" is necessary. Unfortunately, a precise definition of "crowdedness" does not exist. The Italian regulation, as mentioned in the introductory section, rests on an ambiguous definition of crowd intended as an "unusual concentration of people".

In a preliminary version of our work [13], we proposed a simple concept of crowdedness based on people counting. In particular, we labeled an image as containing a crowd only if it contained at least 10 persons in the captured scene. Although simple, this concept led to an effective crowd detector. However, such a characterization does not capture how the individuals in the scene are effectively aggregated. In this work, we inject additional information based on the clustering tendency of the crowd. This may help the FCN learn a better model for crowd detection.

A. LIGHT-WEIGHT NETWORK MODELS

Drones are characterized by limited capabilities in terms of computational power: this calls for the definition of light-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2984768, IEEE Access

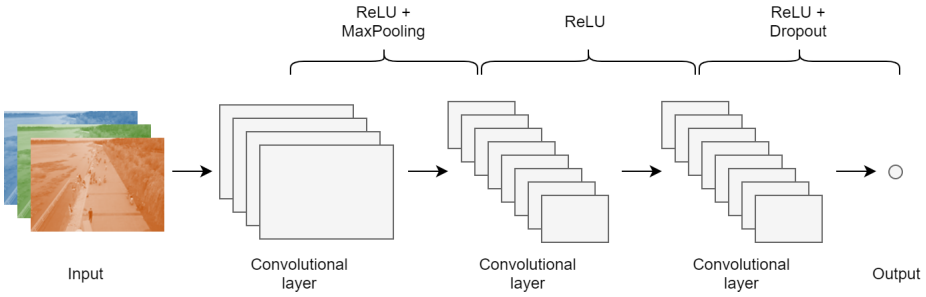

FIGURE 1: The general scheme of the proposed FCN architecture.

weight models to be installed on UAVs for tackling the crowd detection task. When deep learning architectures are considered, we observe that the fully connected (FC) layers bring about some handicaps. In fact, an FC layer, which is typically set up on top of the last convolutional/pooling layer, requires the injection of a fixed-size input. Also, that is the layer where the spatial information integrated with the images are thrown away. Most importantly for our scopes, the FC layer absorbs much of the computational effort. By contrast, resorting to FCNs may result beneficial for reducing the computational costs, managing input of any size and preserving spatial information. Those features are appreciated in computer vision tasks; for instance, some well-known object detectors, such as R-FCN [23], are fully-convolutional.

Moving from the above considerations, the architecture we set up to carry on our experiments is modelled on the scheme illustrated in Fig. 1. As for input, we consider 128×128 images assuming that three channels are involved and the values are normalized in $[0, 1]$. Each image goes through a configuration preserving the initial information, composed by a convolutional layer with 32 filters (values of kernel size and stride are 5×5 and 1, respectively) and then a common rectified linear unit (ReLU) non-linearity. A max pooling layer is employed to down-sample the ReLU output (spatial dimensions are divided by a factor of 2). The addition of pooling layers is a common practice to obtain down-sampling, which is useful to reduce the computational cost, while achieving invariance to small translations. This technique consists in partitioning the input feature map into a set of non-overlapping regions, then applying a pooling operation; in the case of max pooling, the maximum value for each sub-region is pooled. A few filters are initially considered due to the comparatively reduced number of low level features in the images. Such features can be variously combined giving rise to a greater number of high-level features. Therefore, the following two convolutional layers are characterized by 64 filters (kernel size 3×3): the increased computational burden is mitigated by the feature map reduction previously operated by the pooling layer. Both the convolutional layers are followed by a ReLU activation, and a dropout layer (with dropout rate of 0.5) is introduced to reduce overfitting prior to the final output layer. The proposed architecture represents a light-weight model suitable to be implanted in a UAV. However, it is still complex enough to avoid data underfitting.

The scheme depicted in Fig. 1 has been implemented in a number of versions. Firstly, we were interested in discriminating images on the basis of presence or absence of crowd. This kind of binary classification calls for a single output layer with a sigmoid activation function involved. Being N the samples cardinality, y_i the actual class label and $h_\theta(x_i)$ the predicted class label, the network is asked to minimize the cross-entropy loss function:

$$\mathcal{H}(\theta) = \sum_{i=1}^{N} y_i^c \log(h_\theta^c(x_i)) + (1 - y_i^c) \log(1 - h_\theta^c(x_i)),$$

where the superscript c indicates the classification task, θ collectively indicates the weight parameters of the network and x_i is a single training sample. Class distinction is performed on the basis of people count which is a kind of ill-defined information. For that reason, a variant of the previous implementation has been considered to set up a network predicting both the class label (referred to the whole image) and the count of persons (referred to the depicted crowd). To this aim, we conceived a regression task as an auxiliary output to support the classification task. In this sense, the loss function to be considered is the mean absolute error involving the actual and estimated people count:

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

where the superscript r indicates the regression task. Therefore, this specific version of the FCN is asked to minimize a joint loss function composed by the cross-entropy and the mean absolute error loss:

$$\mathcal{J}(\theta) = \mathcal{H}(\theta) + \mathcal{L}(\theta).$$

In this way, although the main task is still the classification one, the network can learn features from the data that may be useful across tasks. The features learned for the regression on the people count may then improve the discriminating ability of the features learned for the classification task.

These implementations represent the preliminary stage of our research [13]. As a further improvement, we propose a different two-loss variant. In this case, $\mathcal{L}(\theta)$ is intended to estimate the clustering tendency of the crowd, in place of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2984768, IEEE Access

FIGURE 2: Spatial graphs superimposed to sample images: the clustering coefficients evaluated for the left and right images are equal to 0.72 and 0.86, respectively.

crowd cardinality, as described in the following subsection. By doing so, we plan to enhance the prediction accuracy.

It is worth noting that the last convolutional layer of the FCN model can be exploited to derive heatmaps of class activation over the input images. In turn, the heatmaps may prove their usefulness to semantically enrich the flight maps. To derive the heatmaps, we resort to the class activation map (Grad-CAM) method illustrated in [24]. Given an input image, this technique extracts the output feature maps of the last convolutional layer and weights every channel by the gradient of the class with respect to that channel. 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 prediction h_θ^c. A summary of the overall feature maps, i.e. a class activation map L_{CAM}, can be obtained as a linear combination, followed by a ReLU:

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

Since some feature maps could be more important than others to make the final decision, as in [24] we propose to use the averaging pooling of the gradient of h_θ^c with respect to the k-th feature map as a weight for the feature map:

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

In practice, $\frac{\partial h_\theta^c}{\partial A_{i,j}^k}$ measures the effect of the (i,j)-th pixel in the k-th feature map on the prediction h_θ^c for the given class. Upsampling the Grad-CAM to the size of the input image enables the identification of the regions that are most relevant for the final prediction. In this way, a heatmap is obtained, indicating how intensely the input image activates the class.

B. CHARACTERIZATION OF CROWDEDNESS

To better characterize the concept of crowd in an image, we propose a graph-based approach. We assume the availability of an annotated image dataset with information on where people are in each scene. Usual bounding boxes conceived for pedestrian detection can be used for the purpose. Given an annotated dataset, for each image I we calculate the precise pixel location of the individuals by computing the middle point of the corresponding annotated bounding boxes. Then, we build a graph G, whose nodes are the middle points of all the individuals in the scenes. Edges are established by the classic Euclidean distance between nodes: if the distance between two middle points is smaller than a given threshold δ, then the corresponding node pair is considered to be connected. Since calculating effective geographical distances between points is impractical, we rely on pixel distances to compute the graph edges. In this way, a so-called *spatial network* can be obtained [25].

Given the graph G associated to an image I, we propose to use the clustering coefficient [26] to derive a characterization of crowdedness in terms of aggregation tendency. The clustering coefficient can provide meaningful insights in several real-world complex networks; for example, it has been successfully applied for analyzing brain networks for neurodegenerative disease investigation [27].

The clustering coefficient for a graph G is computed as:

$$C = \frac{1}{M}\sum_{v \in G} c_v,$$

where M is the number of nodes in G and c_v is the clustering coefficient of a node v. The coefficient c_v of node v is defined as the fraction of possible triangles existing through that node:

$$c_v = \frac{2T(v)}{k_v(k_v-1)},$$

where $T(v)$ is the number of triangles through v and k_v is the degree of v, i.e. the number of edges that are incident to v. A *triangle* is simply a triplet of nodes connected together. Clearly, for images with no people, we have $C = 0$. Figure 2 shows spatial graphs superimposed to sample images.

It is worth noting that, due to the intrinsic nature of the information it captures, the clustering coefficient C can be somewhat high even when few people are in the scene. That is so since C, according to its formulation, is independent

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2984768, IEEE Access

of the actual size of the graph. This may have a twofold effect. On one hand, it may be beneficial to the FCN model to learn the concept of aggregation independently of the people count. On the other hand, it may impose a bias toward the prediction of crowd even when a crowd is not present in the scene. Nevertheless, it should be remarked that the proposed FCN is mainly trained to discriminate between crowd/no-crowd based on the people count (similarly to what we did in our preliminary investigation [13]), while the prediction of C represents an auxiliary output. In other words, C is used as the target to be predicted in place of the crowd cardinality when computing the joint loss $\mathcal{J}$. In this way, the classification task is also helpful to mitigate the chance of misinterpretation in those cases where the value of C is high but the number of people is low.

IV. EXPERIMENT

Our experimental session is oriented to prove the effectiveness of the proposed FCN architecture when applied to the analysis of complex real-world data. For the sake of comparison, we considered two baselines:

a) An implementation of the model proposed in [12] we purposely realized;
b) The popular MobileNet model [14] pre-trained on ImageNet [28].

Concerning the model a), it is characterized by a scheme which is larger than our own. In fact, it includes six convolutional layers, with a parametric ReLU as activation function which follows each layer but the last one. The output of the last convolutional layer is fed to an output layer with a softmax activation. The first and fifth convolutional layers are followed by max pooling layers to reduce their input size. The first pooling layer is followed by a response-normalization layer to improve generalization. Finally, a dropout layer, with dropout rate of 0.5, follows the fifth convolutional layer to reduce overfitting. In our implementation of the model proposed in [12] the main difference concerns the use of a classic ℓ_2 regularization term, applied to every convolutional layer to further mitigate overfitting. Actually, this regularization technique has been applied also in [12] and its performance was only slightly lower than the one ultimately proposed by the authors.

Concerning the model b), it is characterized by a light architecture so that it can be adopted for computer vision applications in mobile and embedded computer scenarios. The MobileNet model is based on depthwise separable convolutions which are a form of factorized convolutions applying a single filter to each color channel. This factorization has the effect of reducing computation and model size. To perform transfer learning on the dataset employed in our experiment, we relied on the common practice to remove the top level classifier (quite specific for the original classification problem) and to stack a custom layer to be trained for our task.

The experiments have been conducted on the complex VisDrone dataset. Section IV-A provides a description of this data together with some details about the re-arrangement we operated to fit the dataset for our purposes. A report of the obtained results is given in Sec. IV-C, including the crowd heatmaps representing the qualitative outcomes provided by the proposed method.

A. DATASET PREPARATION

There is a scarce availability of datasets for crowd detection from drones. For our experimental purposes we resorted to VisDrone [10], a dataset compiled by the AISKYEYE team at Lab of Machine Learning and Data Mining (Tianjin University, China) which has been employed for the annual VisDrone Challenge since 2018. As a basic illustration of VisDrone, we report the presentation provided by the very same team on the landing page of their Website[2] (retrieved March 30, 2020):

> We [...] present a large-scale benchmark with carefully annotated ground-truth for various important computer vision tasks, named VisDrone, to make vision meet drones. [...] The benchmark dataset consists of 288 video clips formed by 261,908 frames and 10,209 static images, captured by various drone-mounted cameras [...] Note that, the dataset was collected using various drone platforms [...], in different scenarios, and under various weather and lighting conditions. These frames are manually annotated with more than 2.6 million bounding boxes of targets of frequent interests, such as pedestrians, cars, bicycles, and tricycles.

VisDrone is the largest dataset of aerial images from drones ever published; some images from the dataset are depicted in Fig. 3 for the sake of illustration.

The manually annotated ground truth is put at user's disposal for the training and validation sets, while it has been deliberately made unavailable for the test sets (to avoid fitting of algorithms during the challenges).

As anticipated by its authors, VisDrone has been proposed to be employed in a number of different tasks, including object recognition and object tracking. Our work deals with crowd detection, therefore we are interested in some particular categories of (human) items annotated into the images, i.e. pedestrians and persons. In particular, we resolved to set a threshold of at least 10 persons to imply the presence of a crowd inside an image. By doing so, we focused our attention on a subset of VisDrone where all the included images are properly labelled as "crowd" or "non-crowd". Such a subset represents the crowd dataset collecting the images we used in our experimental session. The crowd dataset is described in Table 1: it can be noted how the involved training and test sets are well-balanced in the number of instances belonging to different classes.

In addition, we derived the clustering coefficient associated to each image of the crowd dataset. To build the graph, we used a pixel threshold δ which is independent of the proportions of the particular input image and that simply

[2]http://aiskyeye.com

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2984768, IEEE Access

FIGURE 3: Sample images from the VisDrone dataset.

TABLE 1: Description of the crowd dataset.

Class	Training set (size)	Test set (size)
Non-crowd (< 10)	$15,591$	$1,634$
Crowd (≥ 10)	$15,081$	$1,760$
Total	$30,672$	$3,394$

corresponds to $1/10$ of the image width. This threshold was chosen after several preliminary trials on sample images, as it represented a good trade-off between a higher value, which would have resulted in complete graphs, and a lower value, which would have resulted in disconnected graphs. The choice of a single general threshold has been also supported by a factual observation concerning VisDrone: even though the involved scenes are very different, all of them have been shot approximately from the same altitude ($\sim$ 15-30m). The obtained graphs were composed by clusters separated enough to allow a drone to land between them even in the riskiest situations.

B. IMPLEMENTATION DETAILS

In our experiments we used TensorFlow and the Keras API.[3] As for hardware equipment, the training was run offline on an Intel Core i5, running a Windows 10 Operating System on a 8GB RAM, with the NVIDIA GeForce MX110 (2GB of dedicated memory). Instead, the tests were performed on two computational platforms commonly mounted on drones for several applications. The first one was a Raspberry Pi 3, running Raspbian 4.19 Operating System on 1GB RAM, featuring a Quad Core 1.2GHz CPU. The second was an NVIDIA Jetson TX2, running Ubuntu 18.04 Operating System on 8GB RAM. The Jetson TX2 implements a Pascal GPU architecture with 256 cores. This allowed a feasible estimation of the real-time capacity of our models either on a single-board CPU and an embedded GPU.

Training has been performed by applying stochastic gradient descent on the basis of randomly sampled mini-batches of 64 images: they have been resized to 128×128 (thus contributing to a reduced computational cost) and normalized in $[0, 1]$. Learning rate was set at 0.01.

Our models have been tested against different FCN architectures proposed in literature, namely the pre-trained MobileNet [14] and the FCN introduced by Tzelepi and Tefas [12]. Concerning the latter, we realized an implementation for it that we tested setting the same ensemble of parameters mentioned in the original paper. Again, images have been resized to 128×128 (batch size set to 64); the values of learning rate and momentum have been set to 10^{-5} and 0.9, respectively. When we turned to consider the MobileNet model, larger images (224×224) have been considered in input, in line with the higher capacity of that network. Also, MobileNet accepts by default input expressed in the range $[-1, 1]$, therefore each input channel has been re-scaled accordingly. As for the remaining parameters, learning rate has been set to 10^{-5} (to prevent the previously learned weights from being destroyed) and the parameter α is initialized to 0.5 (thus proportionally decreasing the number of filters in each layer, for the sake of the model lightness). We did not perform fine tuning of the MobileNet model, as we noticed that this was detrimental to prediction accuracy.

In fact, during the training phase of the models involved in the experimental session we observed that some irrelevant patterns happened to be learnt quite soon, due to the complexity of the crowd dataset employed. In this sense, overfitting represented a major issue in our tests, therefore all the models have been trained for a few numbers of epochs which demanded for a few hours of processing time. For

[3]The proposed crowd detector is available at https://github.com/gvessio/uav-crowd-detection.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2984768, IEEE Access

TABLE 2: Results. Input is expressed as the number of pixels along the horizontal and vertical dimensions. Accuracy, precision and recall are expressed in percentages according to well-known formulas. Size is measured in MB and speed in fps.

Model	Input	Accuracy	Precision	Recall	Size	Speed (Pi 3)	Speed (TX2)
Replica of [12]	128×128	79.13%	83.76%	79.82%	~ 17.3	1.88	12.47
Pre-trained MobileNet	224×224	83.11%	83.94%	82.82%	~ 3.6	2.01	12.59
Proposed one-loss FCN (classification)	128×128	84.03%	85.42%	83.66%	~ 1.1	2.63	13.00
Proposed two-loss FCN (classification + people count)	128×128	86.80%	86.79%	86.77%	~ 2.0	2.61	12.75
Proposed two-loss FCN (classification + clustering tendency)	128×128	87.95%	88.37%	87.76%	~ 2.0	2.61	12.85

the same reason, we used early stopping with a patience of only 1 epoch, by monitoring the loss value on a validation set randomly held out as a 10% fraction of the training set.

C. EXPERIMENTAL RESULTS

To evaluate the results of our experiments we relied upon the following metrics:

- Percentage of prediction accuracy;
- Percentage of average precision;
- Percentage of average recall;
- Size of the resulting HDF5 file;
- Processing speed.

The first three measures are related to the capability of the model to tackle the classification task (crowd/non-crowd). Size and speed pertain to technical facets of the working process of each model, and are evaluated in terms of MB and frames per second, respectively. Such metrics have been derived during the tests conducted over each model and are reported in Table 2. The table includes the results of the different versions of the proposed method (one loss function; joint loss function involving the crowd cardinality; joint loss function involving the clustering coefficient) as well as those related to the baselines selected for comparison: the FCN proposed in [12] and the pre-trained MobileNet.

The lower value of classification performance has been exhibited by the first baseline which is also characterized by the largest size. It may be argued that these values are correlated, so that this model is hampered by too much capacity for tackling the problem at hand. MobileNet was able to perform better in terms of prediction performance. A couple of reasons may be advanced to explain such a behaviour. On the one hand, MobileNet is still a complex model notwithstanding the halving of the filters which is applied at each layer (indeed, its size is comparatively contained). On the other hand, we should recall that a pre-training was performed for this model on data coming from ImageNet which are quite dissimilar from scenes shot from UAVs (especially concerning shot perspectives which are totally different). It should be noted also that the results reported in table for MobileNet correspond to the pre-selected α value (i.e., 0.5): different values have been tested aside providing worse results.

Among the models which pertain to the proposed FCN architecture, the scheme embedding the one-loss function was able to reach better accuracy, precision and recall than baselines which have been outperformed also in terms of size

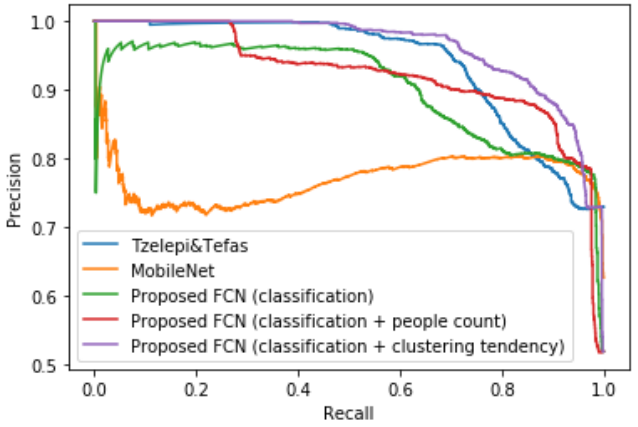

FIGURE 4: Precision-recall curves. They summarize the trade-off between precision and recall for the predictive models at different probability thresholds. A high area under the curve represents both high recall and high precision, where high precision indicates a low false positive rate, while high recall indicates a low false negative rate.

and speed. Actually, the one-loss model exhibits the smallest size and the fastest speed overall: such values characterize it as a qualified candidate to be mounted on drones. Concerning the two-loss models, when the classifier is coupled with the regressor on crowd cardinality, a further enhancement is registered in accuracy (86.80%), precision (86.79%) and recall (86.77%). As expected, adding the regression task effectively improved the prediction performance. Of course, this is paid in terms of bigger size ($\sim$ 2.0MB) of the model and a slightly lower speed (2.61 and 12.75 fps, for the Raspberry and NVIDIA platform, respectively). The best classification results come from the two-loss model based on the estimation of the clustering tendency of the crowd. In fact, it outperforms all the other models not only in terms of accuracy (87.95%) but also in terms of precision (88.37%) and recall (87.76%). At the same time, this model exhibits size and speed which are almost equivalent to the alternative two-loss variant (i.e., $\sim$ 2.0MB and 2.61 and 12.85 fps, respectively). As expected, for all the models we employed, NVIDIA Jetson TX2 outperformed Raspberry Pi 3 in terms of processing speed.

A wider comparison of our results with the state-of-the-art (other than the considered baselines) is not an easy task. In fact, research in this field is still in its infancy; also, different

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2984768, IEEE Access

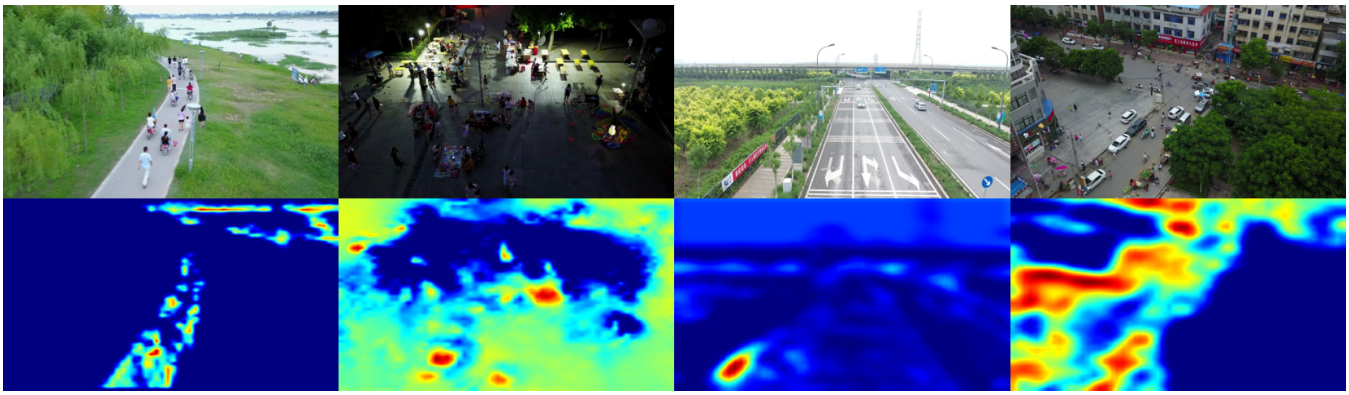

FIGURE 5: Illustrative examples of the heatmaps derived through the application of the proposed method. (For a better visualization, input images have been re-scaled to the original proportions, while heatmaps have been upscaled to fit these proportions).

researchers often work on different data. By referring again to [12], we could argue that our best accuracy value (87.95%) still does not match the best performance reported in those tests (95.46%). However, the crowd dataset we have been working on is much larger than the one adopted in that experimentation. Most of all, it is characterized by a greater variance in its contents. This is the reason why the application of our implementation of that baseline on the crowd dataset failed to reply analogous results in terms of generalization.

Figure 4 illustrates the precision-recall curves evaluated for each model while lowering the confidence threshold. We observe how the leaning of the curves for our two-loss models happens in the top-right corner of the figure. In particular, when the confidence threshold is lowered to a value around 0.4, the two-loss model based on the estimation of the clustering tendency shows a high recall value (98.75%), while keeping a satisfactory performance in terms of precision (which is around 73%). This can be read as a further confirmation of the feasibility of the proposed method in the specific context of our research. In fact, security reasons dictate the necessity to recognize as much people gatherings as possible, even if some amount of precision must be sacrificed. The proposed method proved its efficacy in detecting almost every crowd, without suffering too much from false positive cases.

Heatmaps are useful to visually enhance the analysis of aerial images. Their contribution is also beneficial to better appreciate the results of the experimental results, thanks to the additional pieces of information provided to the observer. That is highlighted in Fig. 5 where some sample images are reported together with the heatmaps resulting from the application of our two-loss method involving clustering tendency. The figure shows the capability of the model in detecting to some extent the presence of people inside the scene, thus offering indication about safe and risky zones for overflying. The work presented in [12] comes to definition of crowd heatmaps too. However, in that study heatmaps have been derived starting from the analysis of high resolution images (1024×1024). This contributed to the definition of notable results, but affected the computational burden. Our approach, instead, produces heatmaps starting from the analysis of re-sized images (128×128) and heatmaps are provided together with the class prediction. Obtaining higher quality heatmaps with the proposed method calls for further research.

V. CONCLUSION

In this paper, we have proposed a novel human crowd detector for aerial images shot by drones. Crowd detection is a crucial task in several applications, particularly whenever drones are brought (or obliged) to overfly zones which are possibly occupied by people. Crowd detection may enable safety mechanisms conceived to automatically adapt the flight plan on a contingency basis and to discriminate between safe and risky regions in case of emergency landing. Several commercially available drones are equipped with on-board cameras and GPUs, therefore computer vision algorithms may be operated to tackle the problem of human crowd detection from drones. However, that can be accomplished only providing (nearly) real-time responses in full compliance with the computational requirements of the UAV's hardware (which sometimes is quite limiting). Moving from such premises, we have proposed a very light-weight Fully-Convolutional Network architecture, trained to distinguish between crowded and non-crowded scenes. The general scheme we conceived is useful to produce different network variants which have been trained from scratch on the very challenging VisDrone benchmark dataset, characterized by a large variety of aerial scenes. In particular, we proposed a method based on the minimization of a joint loss function which combines two terms, respectively related to classification and regression. The regression task aims at predicting the agglomeration tendency of the crowd, based on the clustering coefficient of the corresponding spatial graph. By doing so, it supports the classification task during the analysis of the aerial images and their final discrimination. The proposed model is able to outperform the other variants of the FCN

architecture. One of them is based on a single loss function (conceived for the binary discrimination); the other one is based on a two-loss function where the regressor is aimed at predicting the cardinality of the crowd. All the proposed models are able to provide better predictions when compared to other approaches proposed in literature. Among them, we proposed a comparison with a more complex method based on the well-known MobileNet architecture. However, 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, we compared our method against an FCN purposely designed for crowd detection. All in all, our approach compares favorably with the state-of-the-art, providing an effective, light-weight model.

Future developments of the present research may attempt to further improve prediction accuracy. Promising ways appears to be the application of data augmentation techniques [29] or the generation of synthetic aerial data through generative adversarial networks [30]. In addition, a real-world case study is necessary to confirm the applicability of the proposed system: a practical application is currently the topic of future research. This may also serve to measure energy consumption. Other future directions may concern the extension of the proposed method to application domains other than drone safe landing. For example, human crowd detection from drones can be used for crowd density estimation for the purposes of video-surveillance [31], or to enable the investigation of human crowd behaviour [32]. Developing safe UAV applications can increase the trust on this technology, hopefully making some strict regulations more relaxed.

REFERENCES

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

[2] 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.

[3] T. Castelli, A. Sharghi, D. Harper, A. Tremeau, and M. Shah, "Autonomous navigation for low-altitude UAVs in urban areas," arXiv preprint arXiv:1602.08141, 2016.

[4] P. Nguyen, M. Arsalan, J. Koo, R. Naqvi, N. Truong, and K. Park, "LightDenseYOLO: A fast and accurate marker tracker for autonomous UAV landing by visible light camera sensor on drone," Sensors, vol. 18, no. 6, p. 1703, 2018.

[5] 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.

[6] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," Neurocomputing, vol. 234, pp. 11–26, 2017.

[7] D. Bardou, K. Zhang, and S. M. Ahmad, "Classification of breast cancer based on histology images using convolutional neural networks," IEEE Access, vol. 6, pp. 24 680–24 693, 2018.

[8] G. Liang, H. Hong, W. Xie, and L. Zheng, "Combining convolutional neural network with recursive neural network for blood cell image classification," IEEE Access, vol. 6, pp. 36 188–36 197, 2018.

[9] 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, 2019.

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

[11] 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.

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

[13] 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.

[14] 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.

[15] S. Lin, M. A. Garratt, and A. J. Lambert, "Monocular vision-based real-time target recognition and tracking for autonomously landing an UAV in a cluttered shipboard environment," Autonomous Robots, vol. 41, no. 4, pp. 881–901, 2017.

[16] R. Polvara, S. Sharma, J. Wan, A. Manning, and R. Sutton, "Towards autonomous landing on a moving vessel through fiducial markers," in 2017 European Conference on Mobile Robots (ECMR). IEEE, 2017, pp. 1–6.

[17] A. Marcu, D. Costea, V. Licaret, M. Pirvu, E. Slusanschi, and M. Leordeanu, "SafeUAV: Learning to estimate depth and safe landing areas for UAVs from synthetic data," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 0–0.

[18] K. Mukadam, A. Sinh, and R. Karani, "Detection of landing areas for unmanned aerial vehicles," in 2016 International Conference on Computing Communication Control and automation (ICCUBEA). IEEE, 2016, pp. 1–5.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in Advances in Neural Information Processing Systems, 2016, pp. 379–387.

[24] 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.

[25] M. Barthélemy, Spatial networks. Springer, 2014.

[26] M. Kaiser, "Mean clustering coefficients: the role of isolated nodes and leafs on clustering measures for small-world networks," New Journal of Physics, vol. 10, no. 8, p. 083042, 2008.

[27] E. Lella, N. Amoroso, A. Lombardi, T. Maggipinto, S. Tangaro, and R. Bellotti, "Communicability disruption in Alzheimer's disease connectivity networks," Journal of Complex Networks, 2018.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE Conference on Computer Vision and Pattern Recognition. Ieee, 2009, pp. 248–255.

[29] L. Taylor and G. Nitschke, "Improving deep learning using generic data augmentation," arXiv preprint arXiv:1708.06020, 2017.

[30] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," arXiv preprint arXiv:1711.04340, 2017.

[31] 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.

[32] M. Haghani and M. Sarvi, "Crowd behaviour and motion: Empirical methods," Transportation research part B: methodological, vol. 107, pp. 253–294, 2018.

GIOVANNA CASTELLANO is Associate Professor in Computer Science. She is the coordinator of the CILAB (Computational Intelligence Lab) at the Computer Science Department of the University of Bari. Her research interests are in the area of computational intelligence and include fuzzy systems, fuzzy image processing, fuzzy clustering, image processing and retrieval. She is the co-author of the book "Fuzzy Logic for Image Processing: A Gentle Introduction using Java" (Springer, ISBN 978-3-319-44130-6) and co-editor of the book "Web Personalization in Intelligent Environments" (Springer, ISBN 978-3-642-02793-2). She is Associate Editor of Information Sciences (Elsevier) and International Journal of Systems, Control and Communications (InderScience). She is on the editorial board of Journal of Knowledge-Based and Intelligent Engineering Systems and International Journal of Knowledge and Web Intelligence. She has served as a PC member for several refereed conferences or workshops in the field of computational intelligence, such as FUZZ-IEEE, ICANN, IJCCI, WILF, FCTA.

CIRO CASTIELLO graduated in Computer Science in 2001 and received his Ph.D. in Computer Science in 2005. Currently, he is an Assistant Professor at the Department of Computer Science of the University of Bari, Italy. His research interests include: fuzzy logic, soft computing techniques, inductive learning mechanisms, interpretability of fuzzy systems, explainable artificial intelligence. He participated in several research projects and published more than seventy peer-reviewed papers. He is also regularly involved in the teaching activities of his department. He is member of the European Society for Fuzzy Logic and Technology (EUSFLAT) and of the INdAM Research group GNCS (Italian National Group of Scientific Computing).

CORRADO MENCAR received the Ph.D. degree in Computer Science from the University of Bari, Italy in 2001. Currently, he is Associate Professor in Computer Science at the Computer Science Department of the University of Bari. His research interests are in the area of computational intelligence, granular computing and explainable artificial intelligence. He published more than a hundred scientific papers in international journals, conference proceedings and book collections. He is Associate Editor of IEEE Access, Granular Computing (Springer) and International Journal of Artificial Intelligence. He also serves as PC member in several international conferences.

GENNARO VESSIO received the M.Sc. degree (Hons.) in Computer Science and the Ph.D. degree in Computer Science and Mathematics from the Computer Science Department, University of Bari, Italy, in 2013 and 2017, respectively, where he is currently an Assistant Professor. His current research interests include machine learning and pattern recognition, computer vision and health informatics. He has coauthored articles in these fields, appeared in both international journals and conferences, such as Pattern Recognition Letters, IEEE Access and Cognitive Computation. He served as a Reviewer for many international journals, including Information Sciences and Information and Software Technology, and has been a part of the Program Committee of international conferences, such as IJCAI and SEKE.