Semantic Map Annotation through UAV Video Analysis using Deep Learning Models in ROS

Efstratios Kakaletsis, Maria Tzelepi, Pantelis I. Kaplanoglou, Charalampos Symeonidis, Nikos Nikolaidis, Anastasios Tefas, Ioannis Pitas *

Aristotle University of Thessaloniki, Greece
{nikolaid, tefas, pitas}@aiia.csd.auth.gr

Abstract. Enriching the map of the flight environment with semantic knowledge is a common need for several UAV applications. Safety legislations require no-fly zones near crowded areas that can be indicated by semantic annotations on a geometric map. This work proposes an automatic annotation of 3D maps with crowded areas, by projecting 2D annotations that are derived through visual analysis of UAV video frames. To this aim, a fully convolutional neural network is proposed, in order to comply with the computational restrictions of the application, that can effectively distinguish between crowded and non-crowded scenes based on a regularized multiple-loss training method, and provide semantic heatmaps that are projected on the 3D occupancy grid of Octomap. The projection is based on raycasting and leads to polygonal areas that are geo-localized on the map and could be exported in KML format. Initial qualitative evaluation using both synthetic and real world drone scenes, proves the applicability of the method.

Keywords: Drone Imaging · Crowd Detection · Deep Learning · FCNN · Semantic Mapping · Octomap · ROS

1 Introduction

A 3D map of the UAV flight environment with annotated regions that relate to safety, such as crowd gathering locations or no-fly zones in general, is crucial for drone path planning and navigation. Recently, imposed legislations for drones, forbid the flight in vicinity of crowds, for drone flight safety purposes. For example, the drone flight regulation rules for UK[1] define that drones should not be flown within 50m of people and within 150m of a crowd of over 1000 people, while in Italy[2] it is not allowed for the drone to operate at a distance less than 50m of

* The research leading to these results has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement number 731667 (MULTIDRONE). This publication reflects only the authors views. The European Union is not liable for any use that may be made of the information contained therein.

[1] http://publicapps.caa.co.uk/docs/33/CAP393_E5A3_MAR2018(p).pdf

[2] https://www.enac.gov.it/repository/ContentManagement/information/N1220929004/Regulation_RPAS_Issue_2_Rev 2_eng.pdf

human crowds. Therefore, it is crucial for the drone to be capable of detecting crowds in order to define no-fly zones and proceed to re-planning during the flying operation. Towards this end, in this work we utilize deep Convolutional Neural Networks (CNN) [13]. In particular, we propose a fully convolutional architecture in order to comply with the computational limitations of the application.

During the recent years deep CNNs, have been established as one of the most efficient Deep Learning architectures in computer vision, accomplishing outstanding results in a plethora of computer vision tasks. More specifically, deep CNNs have been successfully applied in image classification [25], object detection [14], semantic segmentation [7], image retrieval [28], and pose estimation [26]. The main reasons behind their success are the availability of large annotated datasets, and the GPUs computational power and affordability.

Thus, in this paper we propose a fully convolutional neural model for crowd detection in drone-captured high-definition (HD) video frames. The fully convolutional nature of the model is crucial in handling input images with arbitrary dimension, and estimating pixel-level probability heatmaps, which in turn are projected on the 3D occupancy grid of Octomap [10] . The projection is based on raycasting and leads to polygonal areas that are geo-localized on the map and could be exported in Keyhole Markup Language (KML) format. Finally, a primary contribution of this paper is a reusable software architecture for Robotic Operating System (ROS) [20] and the implementation of a system that annotates maps with regions of crowd, that are recognized in video frames. That is, utilizing the generated heatmaps we describe the task of map projection that uses the heatmaps together with other sensor data that constitute the set of extrinsic and intrinsic camera parameters for the scene. The detection can be performed offline or during the flight depending on the architecture of the drone and the wireless network connectivity. The prototype implementation of our system demonstrates its applicability for annotating maps with regions of human crowds and exporting them in KML format, used by Google Earth API [9].

The main contributions of this work can be summarized as follows:

- We propose a lightweight fully convolutional model for crowd detection towards drone flight safety
- We propose a generic multiple-loss regularized training method in deep CNNs
- We propose a method that implements the projection of the crowded heatmaps, derived from the crowd detection convolutional model, onto the 3D occupancy grid of Octomap.
- We propose a software architecture for ROS and the implementation of a system that annotates maps with regions of semantic classes (i.e. crowded scene)

The remainder of the manuscript is structured as follows. In Section 1.1, related work is described. In Section 2, we propose the crowd detection method for drone flight safety and in Section 3 we describe the proposed system architecture that implements the UAV mapping. In Section 4, we describe the acquisition of drone data and present results of our crowd detection scenario in both synthetic and real-world drone imagery. Conclusions follow.

1.1 Related Work

Although several works utilize deep CNNs for crowd analysis and understanding, e.g.[3,24,2], research in the topic of crowd detection is rather limited. Furthermore, to the best of our knowledge, crowd detection in drone-captured images, which bears additional challenges (e.g. small person size, occlusions etc.), is an uncharted territory. Since the crowd first needs to be detected, this emphasizes the demand for algorithms capable of efficiently distinguishing between crowded and non-crowded scenes in drone-captured images. A first attempt utilizing state-of-art deep CNNs is presented in [27], where a pretrained model is finetuned for the task of crowd detection.

As we negotiate about flying robots, namely UAVs, in the last years, several approaches have been followed to augment topological maps [21] with semantic information [29], [6], allowing robots to reason about more expressive concepts and to execute more sophisticated tasks. The goal of these techniques is to learn how to split the environment into regions that have a coherent semantic meaning to humans. The combination of semantics with topology is an important step towards closing the gap between the traditional robotic representation of the world and human cognitive maps, making it easier for robots and humans to communicate and cooperate. Recently, the focus of the robotics community has shifted towards semantic representations [19], [16], [15] and object relation modeling in semantic maps [18], [1], [17] to develop autonomous interactive robots that are capable of understanding the semantics and relationships between the objects in the environment, besides exploiting occupancy grid maps for navigation.

2 Proposed Crowd Detection Model

In this work, we propose a crowd detection method for drone flight safety, using deep CNNs. A main focus is to provide a lightweight CNN model, which, satisfying the computational and memory limitations of our application, can distinguish between crowded and non-crowded scenes, in drone-captured images. To achieve this goal, a fully convolutional model is proposed. The fully convolutional nature of the model is crucial in handling input images with arbitrary dimension, and estimating a heatmap of the probability of crowd existence in each location of the input image, that can be used to semantically augment the flying zones. Furthermore, this will allow for handling low computational and memory resources on-drone whenever other processes occur (*e.g.*, re-planning, SLAM, etc.), and only low-dimensional images can be processed on the fly for crowd avoidance.

We should also note that the fully convolutional architectures are accompanied by a series of benefits. For example, the convolutional neural layers preserve spatial information due to the spatial arrangement of the activations, as opposed to the fully connected layers that discard it since they are connected to all the input neurons. That is, the convolutional layers inherently produce feature maps with spatial information. Additionally, an architecture without fully connected layers drastically decreases the amount of the model parameters, and therefore the computational cost is restricted, since the fully connected layers of deep CNNs usually occupy the most of the model parameters. For example, in VGG the fully connected layers comprise 102M parameters out of a total of 138M parameters. Finally, we should note that state-of-the-art object detectors, like SSD, also use fully convolutional architectures.

2.1 CNN Architecture

The proposed CNN model contains six learned convolutional layers. The network accepts RGB images of size $128 \times 128 \times 3$. The output of the last convolutional layer is fed to a Softmax layer which produces a distribution over the 2 classes of *Crowd* and *Non-Crowd*. Each convolutional layer except for the last one is followed by a Parametric Rectified Linear Unit (PReLU) activation layer which learns the parameters of the rectifiers, since it has been proven to enhance the classification performance, while max-pooling layers follow the first and the fifth convolutional layers.

2.2 Multiple-Loss Training

In order to enhance the generalization ability of the proposed crowd detection model, we propose a multiple-loss training method. That is, motivated by the Linear Discriminant Analysis (LDA) method, which aims at best separating samples of different classes, by projecting them into a new low-dimensional space, which maximizes the between-class separability while minimizing their within-class variability, we also propose a new model architecture. The new model, apart from the softmax loss layer which preserves the between class separability, includes an additional loss layer that aims to bring the samples of the same class closer to each other.

That is, considering a labeled representation $\mathbf{z}_i$, we aim to minimize the squared distance between $\mathbf{z}_i$ and the mean representation of its class.

Let $\mathcal{I} = \{\mathbf{I}_i, i = 1, \ldots, N\}$ be the set of N images of the training set, $\mathcal{Z} = \{\mathbf{z}_i, i = 1, \ldots, N\}$ the set of N feature representations emerged in a certain layer of a deep neural model, and $\mathcal{C}^i = \{\mathbf{c}_k, k = 1, \ldots, K^i\}$ the set of K^i representations of the i-th image, belonging to the same class. We compute the mean vector of the K^i representations of C^i to the image representation $\mathbf{z}_i$, and we denote it by $\boldsymbol{\mu}_c^i$. That is, $\boldsymbol{\mu}_c^i = \frac{1}{K^i} \sum_k \mathbf{c}_k$.

Then our goal is defined by the following optimization problem:

$$\min_{\mathbf{z}_i \in \mathcal{Z}} \mathcal{J} = \min_{\mathbf{z}_i \in \mathcal{Z}} \sum_{i=1}^{N} \|\mathbf{z}_i - \boldsymbol{\mu}_c^i\|_2^2, \tag{1}$$

The Euclidean Loss (Sum of Squares) is utilized for implementing the additional formulated regression task in eq(1). We should note that the additional

Euclidean Loss layer can be attached, either to a certain convolutional layer (e.g. last one) or to multiple layers. The proposed multiple-loss training method can be considered as an extra regularization layer that exploits information from the data samples that are relevant to the input image. Generally, multitask-learning [4] constitutes a way of improving the generalization performance of a model. Furthermore, the proposed regularization technique can be applied for generic classification purposes, and also in various deep architectures, which is of utmost importance since deep neural networks are prone to over-fitting due to their high capacity.

2.3 Crowd-Drone Dataset

Since there is no publicly available crowd dataset of drone-captured videos and images, we have constructed a *Crowd-Drone* dataset. The new dataset has been created by querying with specific keywords the Youtube video search engine. More specifically, we collected 57 drone videos using keywords that describe crowded events (*e.g.* marathon, festival, parade, political rally, protests, etc). We also selected non-crowded videos by searching for generic drone videos. Non-crowd images (e.g. cars, buildings, bikes, etc.) were also randomly gathered from the senseFly-Example-drone[3] and the UAV123[4] datasets. Subsequently, we manually annotated crowded regions from the extracted frames. A total number of 5,920 crowded regions and an equal number of non-crowded images formulated the *Crowd-Drone* dataset. Sample regions of crowded and non-crowded scenes are shown in Fig. 1.

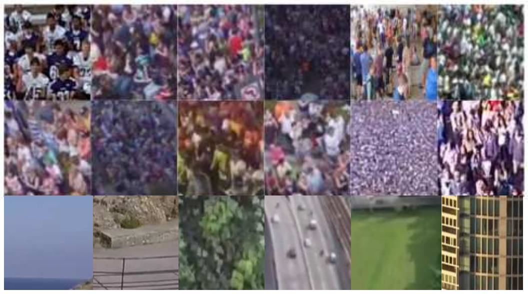


Fig. 1. Sample regions of the *Crowd-Drone* dataset.

We have trained the proposed crowd detection model on the aforementioned dataset utilizing the proposed multiple-loss regularized training method on all the convolutional layers of the model.

3 Proposed System Architecture

3.1 Architecture Overview

The proposed ROS-based architecture (Fig. 2) that implements the UAV mapping is based on image analysis and consists of the Visual Semantics Analyser,

[3] `https://www.sensefly.com/drones/example-datasets.html`

[4] `https://ivul.kaust.edu.sa/Pages/Dataset-UAV123.aspx`

Semantic Map Region Projector and the Semantic Map Manager that are described below.

For each drone a video stream from an on-board camera is published into ROS as a sequence of consecutive ROS image messages, each one corresponding to a grabbed video frame. During flight these messages are transmitted over a wireless network to the processing server which runs our software. Our system requires additional ROS messages for sensor data for the projection into the three-dimensional space of the flight environment. These include the position of the vehicle that is provided by the GPS sensor and pose of the gimbal on which the camera is mounted from the corresponding MCU and camera intrinsic parameters from the camera controller, e.g. the focal length of each input frame. In case of drones that are designed for ROS, these data are published as messages by specialized nodes that run on-board. In scenarios without ROS, sensor data can be received through other media and are published as ROS messages by our software.

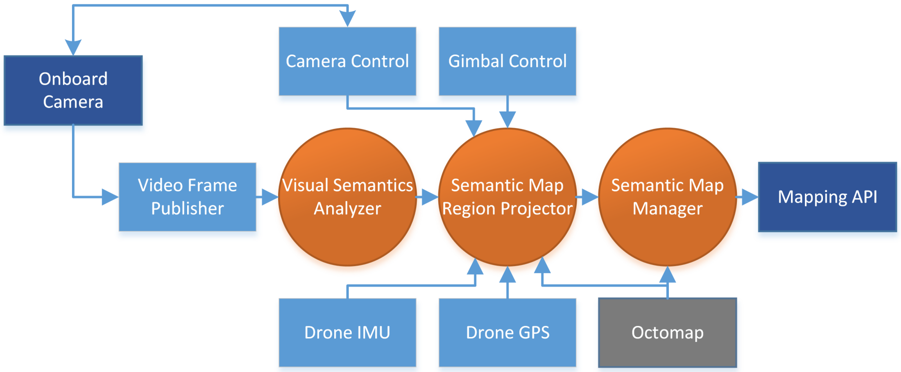


Fig. 2. Outline of the proposed system architecture.

3.2 Visual Semantics Analyzer (VSA)

The Visual Semantics Analyzer receives a single frame that belongs to a video sequence and provides an output in the form of numerical 2D annotations for each input pixel. The values can be either class identifiers (labels) or probabilities for occurrence of a specific class that can be subsequently thresholded for its discrimination. It subscribes to a multiple number of ROS topics and publishes an equal number of output topics. Each incoming ROS image message is tagged with an ID of its origin, e.g. 1 to indicate a frame from drone 1, and placed at a processing queue.

This part of the system uses the deep neural networks to analyze the incoming video frames and derive their visual semantics. Using an enlarged input size, compared to that of training, expands the single class prediction into a heatmap that contains a probability for each patch of the input image. The neural network expects input image with square dimensions and the source frame resolution is 1920x1080 pixels, thus the vertical dimension is padded with zeros and the input size is set to 1920x1920. The spatial dimensions of the FCNN's output activation tensor are significantly smaller due to down-sampling performed by max-pooling layers. It is resized to 1920x1920 using linear interpolation and then cropped to the original input size, providing pixel-level probabilities for the existence of crowds. The numerical annotations of the scene semantics are published as ROS image messages keeping the same timestamp with the source frame.

In addition, a compatibility layer was designed so that any UAV platform or synthetic data can be used with our system. Given a record of sensor data from the same moment in time it publishes messages in three topics that are required for 3D projection. The drone telemetry message contains the GPS coordinates provided by the onboard GPS. If the camera is mounted on a gimbal, a set of pitch, roll, yaw is published as gimbal status. The width and height of the camera sensor and the focal length in millimeters are published in ROS as the camera status. The data can be received offline in the form of a log or data file, online using TCP/UDP sockets or through the web using HTTP requests by implementing interoperability with a web application.

Finally, a visualization helper that runs on its own thread, allows the visual inspection of the FCNN input/output during analysis of live video streams.

3.3 Semantic Map Region Projector (SRP)

The Semantic Map Region Projector (SRP) comprises the processes that conduct projection of the produced heatmap, e.g. the crowd existence heatmap, onto the 3D volumetric map handled by Octomap. This is accomplished by the following stages:

The first stage of the process is responsible of gathering all the appropriate ROS messages and synchronizing them with the current processed heatmap based on their accompanying timestamps. The sensor data contained in these messages must be synchronized so that the camera extrinsic and intrinsic parameters match the moment that the frame was captured. These data include the drone position, gimbal orientation and camera intrinsic parameters that may vary like focal length. By applying thresholding on the heatmap in order to retain only image locations with high probabilities of crowd existence, we convert the image into a binary image where groups of adjacent pixels with value 1 (white) represent 2D regions occupied by crowd. Next we apply a contour following algorithm in order to find the contours of this image, resulting in a new binary image indicating the boundaries (white pixels) of the aforementioned crowd regions 2D polygons. If needed, the polylines are simplified maintaining their shape according to the Ramer-Douglas-Peucker algorithm [5], which takes a curve composed of line segments and finds a similar one with fewer points. By traversing the points (pixels) of the regions' boundaries in a counter clockwise manner, we conduct ray casting [8], [22]. More specifically, this contour image lies on the focal plane of the drone camera, for which we know the following parameters: a) the location of the center of projection (COP) in the 3D world (derived from the drone location), b) the camera orientation (derived from the gimbal state) c) the distance of the focal plane from the COP (the camera focal length). Thus one can cast a ray from each of the boundary contour points towards the voxels of the Octomap. This results in finding the occupied voxel hit by each ray, leading to the evaluation of the X,Y,Z terrain coordinates where each of the contours' points is projected, as the Octomap is coordinates-referenced. Since the 2D boundary contour points are traversed sequentially, so are the points of the 3D boundary contour (polyline).

3.4 Semantic Map Manager (SMM)

The final stage of our pipeline is the Semantic Map Manager whose functionality can be summarized as follows: Firstly, the polygonal lines are fused and delineate crowd gathering locations (see Section 2) on the 3D map. As the drone moves, and its camera sees new areas of the terrain, the newly generated polygonal lines are merged with previous ones using the union operator.

Subsequently, the constantly updated geometric annotations are stored in an internal data layer as ROS messages that will be exported as KML files. These will be used for drone navigation and control purposes as well as for visual inspection by the flight/safety personnel. The KML is a file format used to display geographic data and to overlay annotations on a map such as Google Earth. KML uses a tag-based structure with nested elements and attributes and is based on the XML standard. In our case we use the polygon entity to store the coordinates of the earth surface locations that form the points of the polyline, delineating for example a crowd area.

4 Experiments and Qualitative Results

4.1 Data Creation and Acquisition

Synthetic Scenes In order to test our system for the generation of automatic crowd annotations, we needed aerial footage of crowds at known positions on the 2D plane of the map. Setting up such data acquisition scenarios in the real world, would be very cumbersome. As an alternative, we have generated scenes that contain synthetic crowds in a virtual 3D world environment using Unreal Engine 4 (UE4) [12] and Microsoft AirSim [23].

UE4 is a game engine developed by Epic Games that can achieve high-quality photorealistic graphics, includes a physics engine to simulate real-world physics, supports development in C++ and provides flexible world and asset editors. The AirSim simulator includes a plugin for UE4 that can be used to navigate a virtual UAV inside any 3D world model. In our case various assets, such as crowd and landscape assets, were combined to produce crowd scenes and were programmed to interact and look as realistic as possible. The AirSim plugin was used for controlling the virtual drone and extracting high-definition (HD) images along with simulated sensor data, i.e. camera pose, camera intrinsic parameters, etc. Our setup allows the export of the synthetic crowd positions that are predefined in the 3D space as ground truth annotations for each rendered scene. We have used these annotations to verify the correctness of our system's output along with Full HD (1920x1080) synthetic frames.

Real-World Scenes To test the applicability of our system in real-world environments we gathered video footage and sensor data using an off-the-shelf commercial quadrocopter. The DJI Phantom 4[5] can record video at various resolutions up to DCI 4K (4096x2160) [11], accompanied with a log file that contains values of several internal sensors and microcontrollers. The log file begins at the moment when the engines of the drone start, before take-off. To emulate a drone that operates with ROS we have paired the records from the DJI log file with the frames recorded from the video camera, giving them the same timestamp.

The events in the DJI log are recorded at a specific frequency of 10Hz and the frequency of the published frames in ROS was adjusted accordingly from the 25FPS video stream. We have used the DJI drone to record video of a real crowd that has gathered to attend an open event inside the AUTH campus. The video footage was captured at DCI 4K resolution and has been resized to Full HD to reduce computational complexity of the deep neural network inference.

4.2 Results

Figure 3 shows a simulation depicting crowds that are gathered in front of a road presumably to watch an outdoor sports event, e.g. a bike race. In this simulation, a drone with a cinematographic camera flies above the 3D scene that contains the synthetic crowds and captures a video which is then fed to the VSA, thus producing crowd heatmaps. These are then projected on the 3D terrain and the obtained crowd polygons are exported in KML and visualized in Google Earth. The correctness of the crowd polylines created by the projection was verified by comparing them with ground truth boundaries of the synthetic crowd region on the terrain. A flat terrain was used in Octomap for the specific application of our system.

In the real world scenario, the DJI Phantom 4 drone flies and captures video footage of a crowd that was gathered in the AUTH chemistry square (Figure 4a). The sequence of video frames was fed in the VSA module that had hosted the crowd detector FCNN which produced heatmaps (Figure 4b). This crowd prediction was subsequently projected and led to the simplified polygon depicted as green in the Google Earth environment (Figure 4c), used to visualize the results. The projection task used the Octomap terrain that is presented in Figure 4d. The camera intrinsic and extrinsic parameters as well as the drone position are obtained by subscribing to ROS topics, which have been emulated to publish the contents of the DJI log file.

5 Conclusion and Future Work

In this paper we present a system that annotates maps automatically with semantic knowledge that has been extracted from video frames using a deep neural network. We have implemented the system in ROS for the scenario of discovering crowds through analysis of UAV video frames and projecting them as regions on its navigation map. Our proposed three stage software pipeline can be reused for additional semantic classes like landing zones, water, roads and tree ranges.

[5] `https://www.dji.com/phantom-4/info`

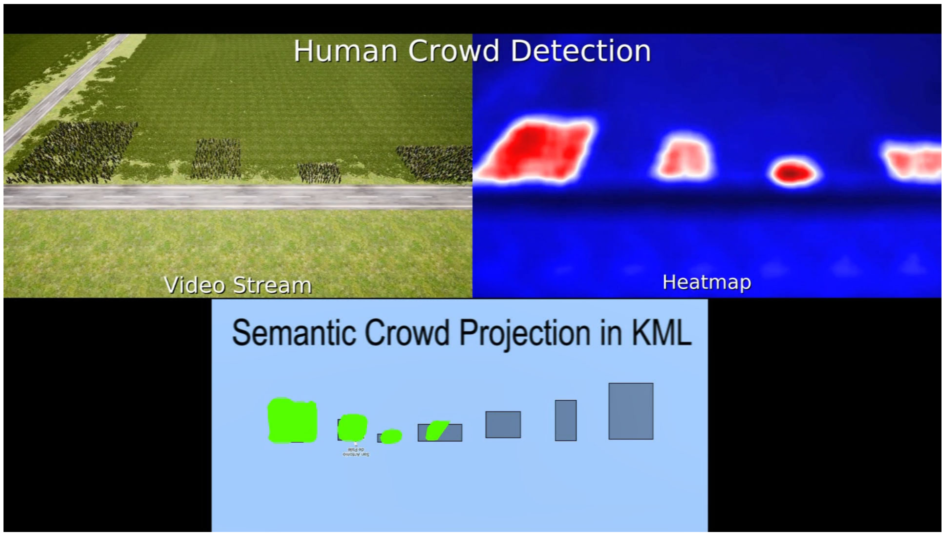
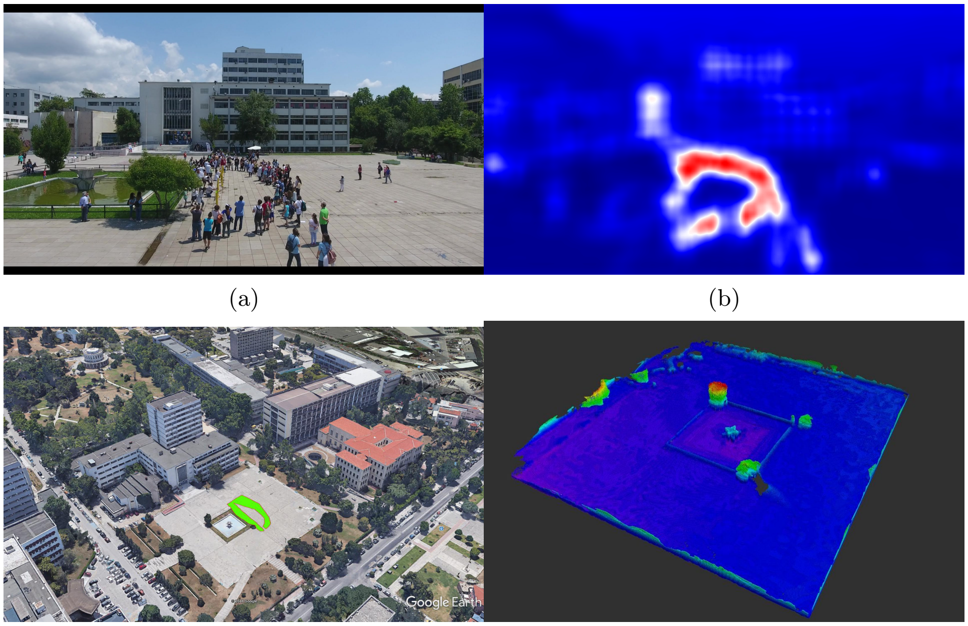


Fig. 3. Application of our system using synthetic crowd scenes and simulated drone sensor data in UE4 and AirSim. Left: Source video frame. Right: Respective heatmap at the output of the FCNN. Bottom: Visualization of the KML annotations (green) over ground truth locations of crowd (gray)



(a) (b)

(c) (d)

Fig. 4. Application of our system using real crowd scenes: (a) Source video frame, (b) Respective crowd heatmap, (c) 3D crowd region projection as depicted in Google Earth, (d) Respective 3D geometric map of the location in Octomap

Moreover it can interface with any geographic information system or web application through the implementation of appropriate interchange formats. Future work will include the creation of a dataset that will contain real crowd images and corresponding sensor data. The dataset will have ground truth annotations of the crowd regions defined by GPS coordinates. These will be used to evaluate the projection accuracy using common metrics like intersection-over-union (IoU). Furthermore we plan to use deep learning models for semantic image segmentation that can provide multiple heatmaps for a given scene, assisting the UAV navigation through semantic understanding of dynamic real-world environments.

References

1. Anand, A., Koppula, H.S., Joachims, T., Saxena, A.: Contextually guided semantic labeling and search for three-dimensional point clouds. The International Journal of Robotics Research **32**(1), 19–34 (2013)
2. Babu Sam, D., Surya, S., Venkatesh Babu, R.: Switching convolutional neural network for crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5744–5752 (2017)
3. Boominathan, L., Kruthiventi, S.S., Babu, R.V.: Crowdnet: a deep convolutional network for dense crowd counting. In: Proceedings of the 2016 ACM on Multimedia Conference. pp. 640–644. ACM (2016)
4. Caruana, R.: Multitask learning. Machine learning **28**(1), 41–75 (1997)
5. Douglas, D.H., Peucker, T.K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: The International Journal for Geographic Information and Geovisualization **10**(2), 112–122 (1973)
6. Friedman, S., Pasula, H., Fox, D.: Voronoi random fields: Extracting topological structure of indoor environments via place labeling. In: IJCAI. vol. 7, pp. 2109–2114 (2007)
7. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Garcia-Rodriguez, J.: A review on deep learning techniques applied to semantic segmentation. arXiv preprint arXiv:1704.06857 (2017)
8. Glassner, A.S.: An introduction to ray tracing. Elsevier (1989)
9. Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R.: Google earth engine: Planetary-scale geospatial analysis for everyone. Remote Sensing of Environment **202**, 18–27 (2017)
10. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: Octomap: An efficient probabilistic 3d mapping framework based on octrees. Autonomous Robots **34**(3), 189–206 (2013)
11. Kaneko, K., Ohta, N.: 4k applications beyond digital cinema. In: Virtual Systems and Multimedia (VSMM), 2010 16th International Conference on. pp. 133–136. IEEE (2010)
12. Karis, B., Games, E.: Real shading in unreal engine 4. Proc. Physically Based Shading Theory Practice pp. 621–635 (2013)
13. Le Cun, B.B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Handwritten digit recognition with a back-propagation network. In: Advances in neural information processing systems 2. pp. 396–404. Morgan Kaufmann Publishers Inc. (1990)

14. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
15. Mitsou, N., de Nijs, R., Lenz, D., Frimberger, J., Wollherr, D., Kühnlenz, K., Tzafestas, C.: Online semantic mapping of urban environments. In: International Conference on Spatial Cognition. pp. 54–73. Springer (2012)
16. de Nijs, R., Ramos, S., Roig, G., Boix, X., Van Gool, L., Kühnlenz, K.: On-line semantic perception using uncertainty. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. pp. 4185–4191. IEEE (2012)
17. Pangercic, D., Pitzer, B., Tenorth, M., Beetz, M.: Semantic object maps for robotic housework-representation, acquisition and use. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. pp. 4644–4651. IEEE (2012)
18. Polastro, R., Corrêa, F., Cozman, F., Okamoto, J.: Semantic mapping with a probabilistic description logic. In: Brazilian Symposium on Artificial Intelligence. pp. 62–71. Springer (2010)
19. Pronobis, A., Jensfelt, P.: Large-scale semantic mapping and reasoning with heterogeneous modalities. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on. pp. 3515–3522. IEEE (2012)
20. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA workshop on open source software. vol. 3, p. 5. Kobe, Japan (2009)
21. Remolina, E., Kuipers, B.: Towards a general theory of topological maps. Artificial Intelligence **152**(1), 47–104 (2004)
22. Roth, S.D.: Ray casting for modeling solids. Computer graphics and image processing **18**(2), 109–144 (1982)
23. Shah, S., Dey, D., Lovett, C., Kapoor, A.: Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics (2017), `https://arxiv.org/abs/1705.05065`
24. Shao, J., Kang, K., Change Loy, C., Wang, X.: Deeply learned attributes for crowded scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4657–4666 (2015)
25. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI. vol. 4, p. 12 (2017)
26. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1653–1660 (2014)
27. Tzelepi, M., Tefas, A.: Human crowd detection for drone flight safety using convolutional neural networks. In: Signal Processing Conference (EUSIPCO), 2017 25th European. pp. 743–747. IEEE (2017)
28. Tzelepi, M., Tefas, A.: Deep convolutional learning for content based image retrieval. Neurocomputing **275**, 2467–2478 (2018)
29. Zender, H., Mozos, O.M., Jensfelt, P., Kruijff, G.J., Burgard, W.: Conceptual spatial representations for indoor mobile robots. Robotics and Autonomous Systems **56**(6), 493–502 (2008)