UWS Academic Portal

Altitude-adaptive and cost-effective object recognition in an integrated smartphone and UAV system

Martinez-Alpiste, Ignacio; Golcarenenji, Gelayol; Wang, Qi; Alcaraz Calero, Jose M.

Published in:
2020 European Conference on Networks and Communications (EuCNC)

DOI:
[10.1109/EuCNC48522.2020.9200951](https://doi.org/10.1109/EuCNC48522.2020.9200951)

Published: 21/09/2020

Document Version
Peer reviewed version

[Link to publication on the UWS Academic Portal](#)

Citation for published version (APA):
Martinez-Alpiste, I., Golcarenenji, G., Wang, Q., & Alcaraz Calero, J. M. (2020). Altitude-adaptive and cost-effective object recognition in an integrated smartphone and UAV system. In *2020 European Conference on Networks and Communications (EuCNC)* (pp. 316-320). (IEEE Conference Proceedings). IEEE.
<https://doi.org/10.1109/EuCNC48522.2020.9200951>

General rights

Copyright and moral rights for the publications made accessible in the UWS Academic Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact pure@uws.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Martinez–Alpiste, I., Golcarenenji, G., Wang, Q., & Alcaraz Calero, J. M. (2020). Altitude–adaptive and cost–effective object recognition in an integrated smartphone and UAV system. In *2020 European Conference on Networks and Communications (EuCNC)* (pp. 316–320). (IEEE Conference Proceedings). IEEE. <https://doi.org/10.1109/EuCNC48522.2020.9200951>

“© © 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Altitude-Adaptive and Cost-Effective Object Recognition in an Integrated Smartphone and UAV System

Ignacio Martinez-Alpiste, Gelayol Golcarenenji, Qi Wang, Jose M. Alcaraz Calero
University of the West of Scotland, UK

Abstract—Human Search and Rescue (SAR) tasks are mission-critical and take place in the wild, and thus solutions require timely and accurate human detection on a highly portable platform. This paper proposes a novel lightweight and practical SAR system that meets those demanding requirements by running optimised machine learning in a smartphone, interoperable with Unmanned Aerial Vehicles (UAV) that provides live video feed. In particular, the proposed approach significantly extends a standard machine learning algorithm to achieve adaptive object recognition in response to changing altitudes to accelerate the speed of finding missing people and eliminate redundant computing. Our approach achieved 91.02% of accuracy and real-time speed on a smartphone that hosts the machine learning platform and the new algorithm. This proposed system is highly portable, cost-effective, fast with high accuracy suitable for UAV applications.

Index Terms—UAV, Machine learning, Deep learning, SAR missions, YOLOv3

I. INTRODUCTION

Over the last few years, Search and Rescue (SAR) teams have been employing various techniques to increase the chances of success in missions to save the lives of humans. Helicopters, dogs, volunteers, sensors and Unmanned Aerial Vehicles (UAV) are typical capabilities put in place to find missing people [1]. Furthermore, the continuous improvements of machine learning techniques and related hardware such as Graphics Processing Units (GPU) have lead to the ability of applying object detection algorithms on constrained device environments such as tablets or smartphones.

Since SAR teams cannot rely on Internet connections due to missions taking place in hazardous environments, their system should be totally portable using constrained device environments. Nevertheless, constrained device environments have limited computational resources and performing object detection algorithms causes reduction in the speed of detection. Moreover, accuracy affects speed. Mostly, when the accuracy is increased, the complexity will also rise leading to speed

This work was in part funded by the Innovation Centre for Sensor and Imaging Systems (CENSIS) and Thales UK, and in collaboration with Police Scotland, under the grant “Smart Unmanned Aerial System for Real-Time Object Detection” (Ref. CAF440). The authors would like to thank all the partners in this project for their support.

Ignacio Martinez-Alpiste(ignacio.alpiste@uws.ac.uk - Corresponding Author), School of Engineering and Computing, University of the West of Scotland, UK. E-mail: Ignacio.Alpiste@uws.ac.uk

reduction. This usually happens in Convolutional Neural Network (CNN) based algorithms, which are also explored as the object detector in this study.

In UAV scenarios, the UAV could fly at different altitudes, resulting in wider or narrower Field of View (FoV) at high and low altitudes respectively. On the one hand, if the algorithm detects humans at high altitudes (wide FoVs), the algorithm should expose more accuracy due to small size of humans, being in the range of a few pixels. On the other hand, when the UAV is flying at low altitudes, the speed of the object detection algorithms should achieve real-time detection to cover the small FoVs covered in each scene without losing information due to the flying speed of the UAV. As a case of point, a use case where an object detector works at 1 FPS and the same field is traversed at the same speed but at different heights, the UAV will lose detection at low altitudes. Fig. 1 depicts this problem in two use cases regarding the UAV altitudes. Our main contribution is to propose a solution to solve this problem by proposing a real-time, light, and portable object detector suitable for constrained environment and is able to detect humans with high accuracy for SAR missions. The contributions of this paper are summarised as below:

- Design and implementation of a portable, cost-effective human recognition system in constrained device environ-

Fig. 1. Exposition of the problem when performing object detection at different altitudes, where h = altitude; t = time (seconds); x = speed; α = angle of view.

TABLE I
RELATED WORK REGARDING SAR APPROACHES.

Ref	Objective	Algorithm	Exec. Environment	Platform	Accuracy	Speed	Model Size
[2]	Human	YOLOv3	Laptop	NG	97.8	NG	NG
[3]	Human	POInet	NVIDIA Jetson Tegra X2	NG	84%	0.667 FPS	NG
[4]	Human	Customed CNN	PC	NG	88.9%	NG	NG
[5]	Human	Customed CNN	Nvidia Jetson TX2	Tensorflow	77.8% 88.6% 85.5%	NG	1.8 MB 48.5MB 379.7 MB
[6]	Human	Tiny YOLOv3	NVIDIA Jetson X1	Darknet	67%	NG	NG
[7]	Human	YOLOv2 608x320 YOLOv2 416x256	NVIDIA Jetson TX2	NG	86.2% 74%	5 FPS 9 FPS	NG
TP	Human	Customed approach	Android Smartphone	Snapdragon	91.05%	20 FPS	28.4MB

TP = This Paper; NG = Not Given; Red = lack of information

ments (smartphones) for SAR missions without sacrificing the accuracy.

- Realisation of an adaptable automatic optimisation of the human detection algorithm by taking into account the current altitude of the UAV and the FoV.
- Training and validation of the proposed object detection system by employing a real dataset gathered with a UAV and consequent labelling of humans.
- Empirical results by testing different parameters such as speed and accuracy in a real-world use case.

The reminder of this paper is as follows: section II reviews state-of-the-art work related to human detection, with a focus on techniques investigated in this paper for object recognition. Section III illustrates our approach to tackling the problem previously mentioned and achieving an altitude-adaptive and cost-effective UAV-smartphone human detection system. Experiments and results are discussed in section IV where the execution environment and the dataset are evaluated, and our approach is justified. Conclusion is given in section V.

II. RELATED WORK

A. Object Detection

Single-stage and two-stage detectors are two key categories of CNN-based object detection algorithms. Fast R-CNN [8], Faster R-CNN [9], and R-FCN [10] are examples of two-stage detectors in which the region of interest is determined in the first step, followed by the object classification and the regression in the second step. These methods are slow with high accuracy. SSD and YOLO are examples of single-step detectors in which both the extraction of region of interest and the object classification are performed at one go. These methods are faster at lower accuracy.

B. YOLOv3 and Tiny-YOLOv3 with 3 Output Layers

YOLOv3 is a single-stage detector that performs with high accuracy and implements Darknet-53 as its feature extractor. It provides results at three different outputs to detect large, medium and small objects respectively corresponding to each output layer. This detector is suitable for detection of small objects due to using deeper convolutional layers. Nevertheless, it is unsuitable for real-time object detection especially in resource-constrained environments [11] due to

its computational complexity associated, which hampers real-time detection speed. Hence, Tiny-YOLOv3 executes less convolutional layers performing faster compared with YOLOv3. Although Tiny-YOLOv3 is less accurate than YOLOv3, it is specialised for constrained environments and the trade-off between accuracy and complexity is conducted. Tiny-YOLOv3 only has two output layers for large and medium objects; therefore, it has low accuracy for the detection of small objects. To address this drawback, the detection of small objects is solved by adding an extra output layer as the standard YOLOv3, creating a customised Tiny-YOLOv3 for increasing the accuracy of small object detection [12], [13].

C. Anchors Boxes

Anchor boxes provide information to the algorithms on the likely range of possible sizes of the human. It is a representation of the width and height of the detection units carried out by the neural network. Then, anchors are rescaled to the closest size of the object detected. Anchor boxes are calculated based on the dataset ground-truth by using k-means technique [12]. The number of the anchors boxes chosen per output layer is three; therefore, nine anchor boxes are defined in total.

D. Search and Rescue

An overview of published results related to SAR missions are described in this section. According to literature, there are some publications regarding SAR operations using UAVs. A comparison table I is provided to address the use cases related to this study.

As an illustration, Mobilenet was employed by Ivanovs et al. [14] to detect human on a SAR robot. The detection speed was not determined in the study. Moreover, the detection was carried out in the range of 2 m away, making it unsuitable for farther distance and for smaller object detection. In another study [3], POInet (Mobilenetv2) was used on NVIDIA GTX 1080 with speed of 0.667 FPS which is not fast enough for our use case. Furthermore, people can only be detected up to 30 m of altitude. In [4], Bozic-Stulic et al. applied a custom CNN to detect people for SAR tasks. Although they achieved a very high accuracy, the performance was not real-time. According to table I, in some studies such as [2],

[4], [6], and [7], the speed or model size is not provided, both of which are essential values to make comparisons, especially when addressing constrained environments. In summary, to the best of our knowledge, this is the first time ever to deploy a light and accurate human recognition system on a smartphone by selecting the output size for detection based on the human size and the altitude of the drone, which significantly improves the speed while maintains the accuracy.

III. DESIGN OF THE PROPOSED SYSTEM

Our architecture has been designed to be operational on a deployment scenario where a UAV is flying over a wild environment searching for missing people. The camera attached to the UAV points down at 180 degrees. The UAV continuously streams live video to the smartphone to be analysed by our algorithms for human detection and thus help the pilot operator in this task.

As explained in section II, Tiny-YOLOv3 has three output layers to perform the detection at different object sizes: large, medium and small. In our use case, these sizes are matched to different altitudes: low, medium and high. The current version of Tiny-YOLOv3 processes all the output layers at once, leading to high processing consumption. By splitting Tiny-YOLOv3 into three object detectors (one per output layer), we can reduce the computation resources caused by executing all output layers. Each output layer runs a different number of convolutional layers based on the size of the object to be detected. Convolutional layer processing is a computationally expensive task. The higher the altitude and the smaller the object, and the more convolutional layers are required to detect finer details and thereby the slower the detection will be. Nevertheless, Tiny-YOLOv3 just needs a few convolutional layers at low altitudes, leading to a faster execution of the CNN-based model. Fig. 1 shows the proposed approach. The lower is the altitude of the UAV, the faster the system performs the detection; however, at higher altitudes, the system needs to spend more time in processing the video frames to deal with very tiny object. However, this extra computational overhead is alleviated by the wider FOVs available at that altitudes, which is a perfect symbiosis between the proposed architecture and the use case tackled.

This splitting of Tiny-YOLOv3 for each output layer in three separate CNN models has been the approach carried out in this research. Then, based on the expected pixel size of the objects to be detected, which is also directly linked to the altitude of the UAV, the optimal selection of the output layer is carried out. Based on the selected output layer, the correspondent Tiny-YOLOv3 neural network model is then loaded into memory to be executed by the GPU.

The relation between the altitudes of the UAV and the output layers is the pixel size of the human at those altitudes. The anchor boxes assists to provide information regarding the size of a human in our approach. The first Tiny-YOLOv3 layer predicts the largest boxes, whilst the last Tiny-YOLOv3 layer predicts the smallest ones. Anchor boxes parameters sustain our approach. These indicators represent the recalculations

TABLE II
CORRESPONDENCE VALUES BETWEEN ALTITUDE AND OUTPUT LAYERS.

Altitude	Low		Medium		High	
Output Layer	0		1		2	
Anchor Box	Max	Min	Max	Min	Max	Min
Pixel Size	5328	1344	636	264	216	36
Altitude	$h \leq 6$		$6 < h \leq 15$		$h > 15$	

Pixel size = anchor x anchor; h = altitude (meters)

of human sizes at three different types of altitudes based on how large a human could be depending on the output layer and the altitude of the UAV. This calculation is based on the gathered dataset (section IV-B). Table II shows this correlation between the output layer and altitude according to human size and anchor boxes. The human size and corresponding height has been experimentally calculated while flying the UAV to maximise the efficiency of the presented results.

Fig. 2 illustrates the workflow of the architecture designed. The following list enumerates each step involved in the continuous execution of the detection algorithms being executed on the smartphone:

- 1) The UAV transmits video from the camera to the smartphone. The smartphone processes each video frame by executing Tiny-YOLOv3. At the take-off phase, the output layer selected is layer 0. Nevertheless, future changes in the altitude will change the execution model.
- 2) While the UAV is sending video streams, it also reports the current altitude.
- 3) This altitude parameter is received by the architectural component in charge of performing the “selection” of the output layer that will processes the frames. When the UAV changes its altitude, the selection of the layer is conducted based on a threshold with the values indicated in table II.
- 4) At the same time, the received video frame is decoded, and resized in order to fit the exact input sizes required by Tiny-YOLOv3. The frame decoded has a 1080p resolution and it is rescaled to 416x416 pixels.
- 5) The Switch changes the output layer immediately and

Fig. 2. Design of the proposed system selecting the correspondent output layer according to the altitude.

updates the values of the anchor boxes. From now on, the frames are processed by the new output layer.

- 6) The output of the execution of any layer is exactly the same, providing the coordinates and the confidence values of all the detection. The frame is then displayed on the screen with the people being detected in a highlighted bounding box.

The execution of our system simplifies the execution of Tiny-YOLOv3 on a constrained environment such as a smartphone. In subsection IV-D, the results regarding the speed of the system demonstrates the efficiency of our approach in such constrained environments.

IV. EXPERIMENTS AND RESULTS

A. Execution Environment

The proposed architecture has been prototyped and executed on a “Xiaomi Blackshark” smartphone, with a Qualcomm “Snapdragon 845” processor (2.8GHz 8 cores) and 8GB RAM. The video captured by the UAV (DJI Inspire 1) is transmitted in real-time to the controller using DJI Lightbridge protocol, which is able to transfer the video stream up to 5 Km away. The onboard camera is a DJI Zenmuse X3 which shoots 4K video at 24 FPS. It features a 1/2.3” CMOS sensor with 94-degree FOV lens which works out to about 20mm in 35mm equivalent.

B. Dataset

This use case required a custom dataset suitable to recognise humans with small pixel sizes. To this end, 150 videos were recorded using the UAV in Scotland wilderness and employing a set of volunteers that are part of our research team to be detected. These videos were recorded in different weather conditions (sunny, cloudy and foggy), at different times of the day, and at different locations. The volunteers were with different appearance, postures and clothing colours in the images to increase the chance to include entropy and diversity in the dataset. A total amount of 26,700 images were extracted and manually labelled. The dataset was split as 80% for training and 20% for validation. In addition, the anchor boxes were calculated based on this dataset in order to separate output layers according to altitudes.

C. Accuracy

As stated, the object recognition algorithm extended Tiny-YOLOv3 with extra output layers able to detect at low, medium and high altitudes. Fig. 3 shows the accuracy each thousand iterations over the gathered dataset. After 15,000 iterations, the Tiny-YOLOv3 with three output layers obtained 91.05% of accuracy (mAP). The split of the Tiny-YOLOv3 in three (one per output layer) separate models explained in design section III does not affect the accuracy since each output layer is just able to detect a specific range of pixels. Moreover, to increase the probability of people to be detected when using UAVs in real-world environments, which is very imperative in this use case, the confidence threshold should be lowered even if some false positives appear since the price

Fig. 3. Accuracy (mAP) over 15000 training iterations.

to do not detect someone when it is there is very high. In consequence, the confidence value was assigned to 10%.

The memory requirements of the trained models is also a key factor. First, the size of the model with all output layers is 35 MB. However, for each output layer, three separate models are loaded into memory with an average model size of 28.4 MB, totalling 85.5 MB, but it is noted that only one model is loaded at a time.

D. Speed

These results are obtained according to the time required for each frame to be processed. The inference time is measured from the moment the frame enters the input layers of the proposed CNN until the output is received. Fig. 5 represents the cumulative average of 1,000 frames to be processed by each output layer. In addition, there is a fourth line (shown in black) that represents the real-time speed (24 FPS). The results below the black line indicates the real-time performance. In this use case, our system performs in real-time at layer 0, for detection at low altitudes where faster speed is required. These results show that our system detects 25.3 FPS, 20.8 FPS and 15 FPS at layer 0 (big-sized people at low-altitude range), layer 1 (medium-sized human at medium-altitude range) and layer 2 (tiny-sized human at high-altitude range) respectively. These results have clearly validated the suitability of our approach for the SAR use case.

E. Qualitative Results

Our approach was deployed and tested in a real environment by Police Scotland to assist them in SAR missions. Fig. 4 shows three different screenshots of human detection from a UAV in various locations. Fig. 4a displays a detection performed in the first output layer at 6 m of altitude. In fig. 4b, the detection was performed at the second output layer with the flying height of 14m. Fig. 4c presents a detection at the height of 52m with the third output layer in a tall-grass environment. As apparent in the results, the system successfully detected

(a) 1st output layer detection of a human at 6m of altitude in a forest.

(b) 2nd output layer detection of a human at 14m of altitude in an open area.

(c) 3rd output layer detection of a human at 52m of altitude in a tall-grass.

Fig. 4. Detection based on three output layers at different altitudes and environments.

people in all the situations and no cases of false negatives were observed.

V. CONCLUSIONS

In this paper, a novel smartphone-based real-time altitude-adaptive human detection (object recognition) system has been proposed for SAR missions. The main approach for the machine learning algorithms is based on extending Tiny-YOLOv3 by splitting Tiny-YOLOv3 into three separate models based on the output size, and the proposed system leverages the altitude sensor to increase the speed and eliminates redundant computation while maintaining the accuracy. Our novel approach achieved 91.05% of mAP and real-time detection when

Fig. 5. Cumulative average of the inference time for each output layer over a thousand frames. Black line represents real-time object recognition at 24 FPS.

running on the smartphone. This system expects to improve the efficiency and performance of the SAR missions in wilderness.

REFERENCES

- [1] M. B. Bejiga, A. Zeggada, and F. Melgani, "Convolutional Neural Networks for Near Real-Time Object Detection from UAV imagery in Avalanche Search and Rescue Operations," *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 693–696, 2016.
- [2] A. Quan, "Project Vulture : A Prototype for Using Drones in Search and Rescue Operations," *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 619–624, 2019.
- [3] R. Galdes, A. Gonçalves, T. I. N. Lai, M. Villerabel, W. Deng, A. N. A. Salta, K. Nakayama, Y. Matsuo, and H. Prendinger, "UAV-based Situational Awareness System using Deep Learning," vol. 4, pp. 1–12, 2019.
- [4] D. Božić-Štulić, Ž. Marušić, and S. Gotovac, "Deep Learning Approach in Aerial Imagery for Supporting Land Search and Rescue Missions," *International Journal of Computer Vision*, vol. 127, no. 9, pp. 1256–1278, 2019.
- [5] S. V. Kapur, "Marine search and rescue using light weight neural networks," no. August, 2019.
- [6] K. Tarchanidis, A. Mitropoulos, and A. Gasteratos, "Unsupervised Human Detection with an Embedded Vision System on a Fully Autonomous UAV for Search and Rescue Operations," pp. 1–20, 2019.
- [7] N. Tijtgat, W. Van Ranst, B. Volckaert, T. Goedemé, and F. De Turck, "Embedded real-time object detection for a UAV warning system," *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, vol. 2018-Janua, pp. 2110–2118, 2018.
- [8] R. Girshick, "Fast R-CNN," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 International Conference on Computer Vision, ICCV 2015, pp. 1440–1448, 2015.
- [9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [10] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," 2016. [Online]. Available: <http://arxiv.org/abs/1605.06409>
- [11] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018. [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [12] AlexeyAB, "Darknet." [Online]. Available: <https://github.com/AlexeyAB>
- [13] W. He, Z. Huang, Z. Wei, C. Li, and B. Guo, "Tf-yolo: An improved incremental network for real-time object detection," *Applied Sciences*, vol. 9, no. 16, p. 3225, 2019.
- [14] A. Ivanovs, A. Nikitenko, M. Di Castro, T. Torims, A. Masi, and M. Ferre, "Multisensor low-cost system for real time human detection and remote respiration monitoring," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2019, pp. 254–257.