Visual Object Detection For Autonomous UAV Cinematography

Fotini Patrona*, Paraskevi Nousi, Ioannis Mademlis, Anastasios Tefas, and Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece

Abstract

The popularization of commercial, battery-powered, camera-equipped, Vertical Take-off and Landing (VTOL) Unmanned Aerial Vehicles (UAVs) during the past decade, has significantly affected aerial video capturing operations in varying domains. While UAVs have become affordable, agile and flexible, providing access to otherwise inaccessible spots, though, their limited resources burden computational cinematography techniques on operating with high accuracy and real-time speed on such devices. State-of-the-art object detectors and feature extractors are, thus, studied in this work, in an attempt to find a trade-off between performance and speed that will allow UAV exploitation for intelligent cinematography purposes. Experimental evaluation is performed on three newly introduced, publicly available datasets of rowing boats, cyclists and parkour athletes, while evidence is provided that even limited-resource autonomous UAVs can indeed be used for cinematography applications.

1 Introduction

The use of camera-equipped Unmanned Aerial Vehicles (UAVs) for covering public sport events, such as bicycle or boat races, parkour shows and football games, as well as for media production, surveillance, search and rescue operations, etc., is becoming increasingly popular, since UAVs are capable of shooting spectacular videos that would otherwise be very difficult and costly to obtain. Visual analysis tasks may, thus, be of assistance in UAV-based intelligent cinematography [5, 12, 14, 16], e.g., for detecting and tracking a desired target, or even in flight safety related tasks [21], such as obstacle detection and avoidance. Technological progress has led to the production of numerous commercially available UAVs with similar cognitive autonomy and perceptual capabilities, but the limited computational hardware, the possibly high camera-to-target distance and the fact that both the UAV/camera and the target(s) are moving, constitute achieving both high accuracy and real-time performance rather challenging [9, 10, 11].

The most promising state-of-the-art approach towards achieving real-time performance on the restricted computational hardware on-board a UAV is to use one-stage deep neural detectors, structured around the concept of "anchors". Such detectors, e.g., Single-Shot Detector (SSD) [7] and You Only Look Once (YOLOv2) [17], are based on the notion of a convolutional Region Proposal Network (RPN), simultaneously regressing the pixel coordinates of multiple visible object Regions-of-Interest (ROIs) (in the form of spatial offsets from the predefined anchors) and assigning class labels to them.

SSD [7] is a single-stage multi-object detector, meaning that a single feed-forward image pass suffices for the extraction of multiple ROIs with coordinate and class information, without internal ROI pooling. In its original form, the detector relied on the VGG16 [18] architecture for feature extraction, with the addition of a number of layers upon it, so as to extract better defined boxes. Two versions were proposed, one requiring an input image size of 300×300 pixels and one requiring 500×500 pixels, with the latter producing better results in terms of detection precision while being significantly slower than the former. In [3], SSD was used as a meta-architecture for single-stage object detection and was compared to region-based detectors. The experimental evaluation conducted in this work, proved that when combined with MobileNets [2] and Inception v2 [4] feature extractors, SSD is fastest than any region-based detector at

[*] Corresponding Author: foteinpp@csd.auth.gr

https://doi.org/10.7557/18.5099

the cost of detection precision, though.

Similar in nature to SSD, YOLO [17] is a widely used object detector, whose popularity may be attributed to its simplicity, stemming from its ability to detect multiple objects with a single forward image pass, in combination with its speed, which surpasses that of SSD. YOLO relies on a custom architecture for feature extraction and is pretrained on the ImageNet [1] publicly available dataset. Its fully-convolutional architecture [7, 8] allows the network to be trained and deployed at any resolution, although odd multiples of 32 (the network's total subsampling factor) are preferred in order for the final heatmap produced to effectively divide the image into equally sized overlapping regions. Each such region is responsible for detecting any object whose center lies within it, by fitting precomputed anchor boxes to the ground-truth ROIs. Thus, input size affects not only the size of the produced heatmaps, and consequently the speed of the classifier, but also the maximum number of boxes that can be detected.

These widely used, heavily studied, lightweight neural architectures, along with the fact that autonomous UAV usage for cinematography purposes tends to become mainstream, inspired this study aiming to identify the circumstances under which these architectures could operate on such resource-limited devices, making them suitable for use in intelligent cinematography applications. To this end, an extensive experimental evaluation is performed, testing the detectors paired with different feature extractors, and recording the accuracy and time performance achieved for several input image resolutions, in search of a trade-off between detection accuracy and speed. Evaluation is performed on three use cases, corresponding to real-life applications suitable for autonomous UAV coverage. The created datasets, which are publicly available and can be downloaded from `http://www.aiia.csd.auth.gr/LAB_PROJECTS/MULTIDRONE/AUTH_MULTIDRONE_Dataset.html`, the adopted protocols and the obtained results are subsequently discussed.

2 Use Cases

In this Section, the experimental protocols adopted are discussed and results on the following three use cases are reported: row boat race, cycling race and parkour. These scenarios were selected based on the large benefits induced on their media coverage process by exploiting autonomous UAVs for filming and broadcasting. All time-dependent measurements were made on an NVIDIA Jetson TX2 computing board, i.e., a common embedded AI hardware platform which is easily deployable on drones. All three datasets were manually collected, annotated and made public, as no publicly available datasets of such data currently exist to the best of our knowledge.

2.1 Lightweight Rowing Boat Detection

The use of lightweight convolutional object detector [13, 15, 19, 20] SSD (with various backbones) was investigated regarding rowing boat detection. First, rowing videos publicly available on YouTube, as well as footage shot by Deutsche Welle from the 2018 rowing regata in Wannsee were amassed, constituting a large-scale dataset of 40786 images, 34191 used for training and 6595 for testing. The dataset was then annotated with ROIs of rowing boats. No need for the explicit creation of a validation set arose, due to the way the employed SSD implementation operates (it automatically withholds a random 10% of the training data for validation purposes).

Table 1: Performance and speed (FPS) of various versions of the SSD detector on TX2 trained on the boats dataset.

Architecture	**Input Size (px)**	**AP(%)**	**FPS**
SSD Inception v2	300 × 300	64.25	9
SSD Inception v2	400 × 400	67.20	8
SSD Mobilenet v1 FPN	320 × 320	59.84	5.8
SSD Mobilenet v1 FPN	480 × 480	71.34	3.8
SSD Mobilenet v1 FPN	640 × 640	76.15	1.2
SSD Mobilenet v2	300 × 300	61.90	7
SSD Mobilenet v2	400 × 400	65.40	6.1
SSDlite Mobilenet v2	300 × 300	56.13	10

Average precision (AP) and speed results on a Jetson TX2 module, obtained using the SSD detector coupled with MobileNet v1 and Inception v2 backbone feature extraction networks, are summarized in Table 1, while rowing boat detection examples are given in Fig. 1.

The Inception v2 extractor seems to be faster than MobileNet v1, while also leading to more accurate detections. Moreover, as expected, decreases in input image resolution result in execution speed-up but deteriorate performance, as the relatively small input sizes used to train the detectors are responsible for most of the false negative sample instances arising. This is due to the fact that the objects to be detected may sometimes be of rather small sizes, e.g., boats far away from the camera, and thus, lowering input image resolution shrinks small target items to tiny, making them indistinguishable, even to the human eye.

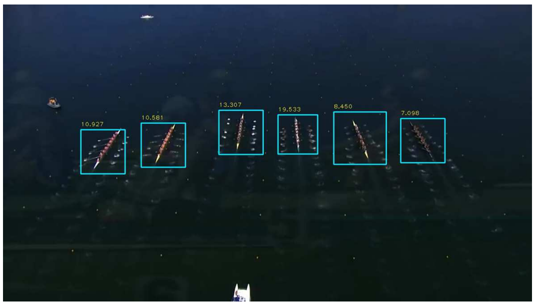

Figure 1: Rowing boats detected by SSD detector.

2.2 Bicycle Detection

SSD detector was evaluated on another single-class problem, that of detecting bicycles in a cycling race. To this end, a dataset consisting of about 12k images was gathered from cycling events and about 77k cyclists were annotated, along with their bicycles. As most of the shots were aerial, the annotated objects (i.e., professional bicycles) are small relative to image size and can be easily confused with other vehicles, such as motorcycles, especially in distant shots, while many partially occluded targets as well as motion blurred instances are also included in the dataset. Finally, it should be noted that, despite the fact that both the cyclist ("person") and "bicycle" classes are popular in datasets such as COCO [6] and ImageNet, on which the detectors have been pretrained, in this scenario, a target is considered to exist only when both "sub-objects" are detected close to each other.

Figure 2 illustrates the performance of SSD with MobileNet v1 and Inception v2 backbone detectors.

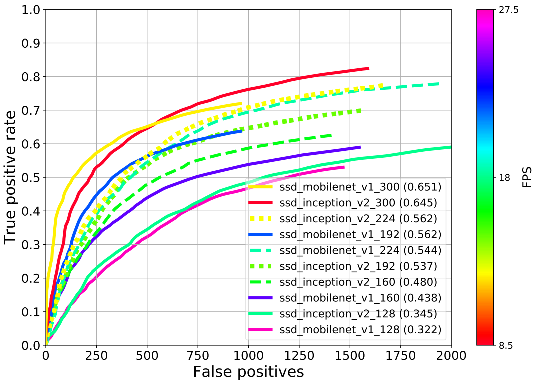

Figure 2: True positive vs false positive rate for the SSD detectors with MobileNet and Inception base models on the Bicycle Detection benchmark.

As expected, for a specific number of false positive detections, e.g., 500, performance rises dramatically as resolution increases — from 32.2% to 65.1% for MobileNet and from 34.5% to 64.5% for Inception. By allowing more false positives, the Inception models achieve higher recall rates, while at around 22 FPS and 56.2% recall rate (at 500 false positives), the MobileNet v1 model at an input resolution of 192 × 192 pixels is identified to offer a great trade-off between speed and accuracy. The same results are also summarized in Table 2 for both backbones and all input sizes. Bicycle detection examples are depicted in Figure 3.

Table 2: Frames per second and precision scores for various input sizes for the SSD with Inception v2 and MobileNet v1 feature extractors on the bicycles dataset.

Input Size (px)	**Extractor**	**FPS**	**AP(%)**
300 × 300	Inception v2	8.5	73.0
	MobileNet v1	12.4	64.5
224 × 224	Inception v2	12.7	57.8
	MobileNet v1	18.4	53.8
192 × 192	Inception v2	14.7	45.7
	MobileNet v1	22.0	48.1
160 × 160	Inception v2	16.4	35.6
	MobileNet v1	24.4	32.9
128 × 128	Inception v2	18.0	27.7
	MobileNet v1	27.5	28.2

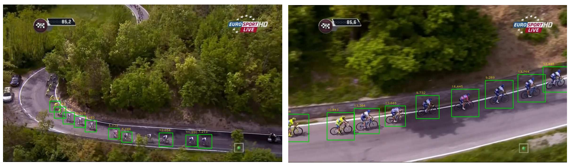

Figure 3: Bicycles detected by SSD detector.

2.3 Parkour Athlete Detection

The single-stage detector YOLOv2, was also trained to detect parkour athletes. More specifically, YOLOv2 pretrained on COCO public object detection dataset was finetuned with parkour data extracted from publicly available Youtube videos. In detail, 8 videos were manually annotated with parkour athlete ROIs, resulting in a 30624-image dataset, 28372 of which were used for training and 2252 for validation purposes. Model testing was performed on footage specifically captured for this purpose at Bothkamp, resulting in 4987 more video frames.

Table 3: One-class YOLOv2 results on parkour dataset.

Input size (px)	mAP(%)	F1	FPS
608 × 608	76.17	0.79	3.8
544 × 544	77.20	0.81	4.1
480 × 480	77.74	0.81	7.2
416 × 416	78.01	0.81	8
352 × 352	78.56	0.78	9
288 × 288	70.99	0.75	10
224 × 224	70.20	0.68	16
192 × 192	61.56	0.63	19

The training protocol adopted was the following. A one-class implementation of YOLOv2 detector, pretrained on COCO dataset, was finetuned in order to detect only parkour athlete instances. To this end, only athlete annotations were used for training, and COCO person class weights were employed for network initialization, aspiring to make the detector capable of detecting athletes performing parkour as "persons". Training sessions for several input image resolutions were conducted, and the obtained results are presented in Table 3, along with the respective processing speeds for algorithm execution on an nVidia Jetson TX2 board. The reported metrics are mean Average Precision (mAP), F1-measure and Frames per Second (FPS), in order of appearance. Parkour athlete detection examples are depicted in Fig. 4.

It can be easily noticed that as the input image resolution falls, processing speed increases, while the best mAP and F1 results are obtained at an image resolution of 416 × 416 pixels. This can be attributed to the fact that as image resolution gets greater than 416 × 416 pixels, the increase in True Positive Rate (TP) becomes smaller than the increase in False Positive Rate (FP), thus resulting in lower mAP scores.

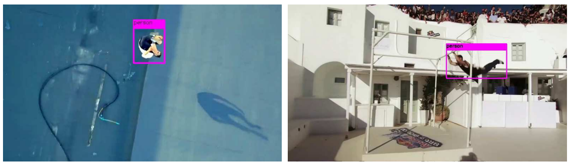

Figure 4: Parkour athletes detected as "persons" by one-class YOLOv2 detector.

3 Conclusions

This paper studied the use of state-of-the-art CNN-based visual object detectors, namely SSD and YOLO, on autonomous UAVs for cinematography applications, under the assumption of limited resources. A trade-off between the obtained accuracy and time required was searched for, and experiments on three newly introduced datasets consisting of rowing, cycling and parkour data, respectively, indicated that for relatively low-resolution input images, rather satisfactory results can be obtained regarding detection accuracy, while also achieving real-time or near real-time execution speed on NVIDIA Jetson TX2 module. This is made feasible with the aid of the fastest feature extraction neural architectures currently available, namely MobileNets and Inception v2. The obtained results can thus be considered to provide evidence that despite their limited resources, UAVs can be employed effectively for computational cinematography and embedded visual analysis tasks.

Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No731667 (MULTIDRONE). This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

References

[1] 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*, pages 248–255. IEEE, 2009.

[2] 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.

[3] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017.

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

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

[6] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2014.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. *European Conference on Computer Vision*, pages 21–37, 2016.

[8] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[9] I. Mademlis, V. Mygdalis, N. Nikolaidis, M. Montagnuolo, F. Negro, A. Messina, and I. Pitas. High-level multiple-UAV cinematography tools for covering outdoor events. *IEEE Transactions on Broadcasting*, 65(3):627–635, 2019.

[10] I. Mademlis, V. Mygdalis, N. Nikolaidis, and I. Pitas. Challenges in autonomous UAV cinematography: An overview. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, Jul 2018.

[11] I. Mademlis, N. Nikolaidis, A. Tefas, I. Pitas, T. Wagner, and A. Messina. Autonomous unmanned aerial vehicles filming in dynamic unstructured outdoor environments. *IEEE Signal Processing Magazine*, 36(1):147–153, 2018.

[12] I. Mademlis, N. Nikolaidis, A. Tefas, I. Pitas, T. Wagner, and A. Messina. Autonomous UAV cinematography: A tutorial and a formalized shot type taxonomy. *ACM Computing Surveys*, 52(5):105, 2019.

[13] P. Nousi, D. Triantafyllidou, A. Tefas, and I. Pitas. Joint lightweight object tracking and detection for unmanned vehicles. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 160–164. IEEE, 2019.

[14] N. Passalis and A. Tefas. Concept detection and face pose estimation using lightweight convolutional neural networks for steering drone video shooting. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 71–75, Aug 2017.

[15] N. Passalis and A. Tefas. Training lightweight deep convolutional neural networks using bag-of-features pooling. *IEEE transactions on neural networks and learning systems*, 30(6):1705–1715, 2018.

[16] F. Patrona, I. Mademlis, A. Tefas, and I. Pitas. Computational UAV cinematography for intelligent shooting based on semantic visual analysis. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4155–4159, Sept 2019.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[19] D. Triantafyllidou, P. Nousi, and A. Tefas. Lightweight two-stream convolutional face detection. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1190–1194. IEEE, 2017.

[20] D. Triantafyllidou, P. Nousi, and A. Tefas. Fast deep convolutional face detection in the wild exploiting hard sample mining. *Big Data Research*, 11:65 – 76, 2018.

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