A survey on deep learning-based object detection algorithms for drones

Abdelmalek Bouguettaya, Ahmed Kechida and Amine Mohammed Taberkit

Research Center in Industrial Technologies (CRTI)
Po. Box 64, Cheraga 16014
Algiers, Algeria
bouguettaya.abdelmalek@gmail.com, a.kechida@crti.dz, taberkit.amine@yahoo.fr

***Abstract* – Unmanned Aerial Vehicles (UAVs) are being used in a very large number of applications. Building an intelligent UAV is a very exciting and challenging topic. Recently, deep learning and computer vision are highly used for the purpose of realizing a fully-autonomous drone that does not need human intervention. Computer vision is a field focused on enabling drones to interpret and understand the content of an image or a video using Convolutional Neural Networks (CNNs). This paper focuses on reviewing recent deep learning-based object detection algorithms used for UAVs. We will discuss the most important research papers and techniques which helped improve the object detection state-of-the-art for drones. Finally, we will conclude this reviewing with a description of the main challenges for the application of deep learning for drone-based solutions.**

Keywords - Computer Vision, UAV, Deep Learning, Object Detection, Convolutional Neural Network.

1. Introduction

In recent years, there has been a growing interest in the use of drones for a wide range of applications like aerial delivery systems [1], surveillance [2], traffic monitoring [3, 4], search and rescue operation [5]. Using artificial intelligence and computer vision methods can make drone more efficient for various operations.

The purpose of computer vision is allowing drones to analyze, process and understand the content of digital images acquired through cameras and sensors so that they can decide how to act by itself. One of the most important questions in the computer vision field is: What objects are seen in the image and where they are located? This is called object detection, which is a core task for each of the mentioned drone applications. Object detection consists of different subtasks like face detection [6], pedestrian detection [7], car detection [8]. Computer vision is not only used to identify objects. However, it is also used to build up a physical model of the world.

In extreme conditions and harsh environment, drones perform better than humans, and are relatively not expensive compared to other technologies. We can use drones for object detection in many applications, where it must be able to detect the object of interest from a cluttered background such as: detecting weeds in agricultural crops, searching for survivors in floods or open sea.

Object detection with drones present more challenges due to its top view angles, real time problem, and the requirement of high computational resources for video processing. Most of the applied deep learning operations are made on the ground using very powerful workstations or servers. This is because deep neural networks perform convolutions: very expensive operations in calculation and in memory. The classification of images in on-board embedded systems is, therefore, a major challenge due to material constraints.

In the last few years, deep learning has shown an accuracy on many applications. Thanks to Moore's law [9], we are able to integrate high-performance processors in a single small chip. That gives us the ability to implement very efficient deep learning algorithms in these chips in order to build a fully autonomous drone that could navigate without the intervention of humans. Also, we have many efficient software tools at our disposal (Keras, Tensorflow, Theano). These hardware and software give us the ability to rapidly construct deep learning architectures in a fraction of the time, while it took us just very long time before.

Unlike traditional methods, where we needed to apply a hand engineered algorithm, where a hand-defined set of rules and algorithms are applied to extract features from an image [12, 39]. Convolutional neural network is an end to end model that gives us the possibility to skip the feature extraction step, which is automatically learned from the training process [39]. Several studies have been made to tackle object detection problem using CNN.

Due to the top-down view angles and the far target, object detection for drones has more difficulties and challenges, which make it a very exciting subject. These last few years, deep learning field has delivered impressive results that, sometimes, outperform human's ability. There are many efficient deep learning-based objects detection algorithms that we can adapt and use them for drones.

In this paper, we will present an overview of the most common and recent CNN-based approaches, which were used for object detection and localization.

2. Deep learning overview.

Deep learning is an exciting branch of machine learning that uses huge amount of data to teach machines how to do things only humans were capable of before. Solving the problem of perception, recognizing what's in an image what people are saying when they're talking on their phone, helping drones explore its environment and interact with it, are some of the most exciting and challenging topics.

Deep Learning has emerged as a central tool to solve perception problems in the last decade. It's the state of the art on everything having to do with computer vision and speech recognition. Increasingly, people are finding that deep learning is a much better tool to solve problems, like understanding natural language, detecting objects in a scene, understanding documents.

Today, computer vision is one of the most treated problems using deep learning and convolutional neural networks. Object detection owned significant consideration and appreciated as one of the most promising applications in the field of computer vision.

Drones are one of the applications that can employ automatic object detection, which has been extensively studied in the literature. They are widely in use for military, police, and civilian applications [10].

Traditional object detection techniques utilize handed engineering features extraction, like Haar-like Cascade classifiers to detect vehicles [11]. Most of the latest object detection techniques rely on deep CNN. In [12], authors utilize CNN as a classifier to detect vehicles in grayscale images. However, these approaches are inefficient for drone platforms because they need to process a very large number of sliding windows. N. Audebert and al. [13], present a deep learning framework to segment, detect and classify vehicles from aerial RGB images. The proposed network is unsuitable for a lightweight and low power drones. Because of encoding and decoding the input image into segments and runs off-line on a GPU. The authors of [14] introduce a new vehicle detection approach based on a combination of CNN and Binary Normed Gradients (BING). This approach achieved a significant speed and accuracy against traditional approaches, like the LBP + SVM [15] and HOG + SVM [16] approaches.

3. Object detection methods.

Object detection methods are mainly categorized into two classes. The first one is the region proposals-based framework, which is based on combining region proposals with CNNs. The second method regards object detection as a regression or classification problem.

The region proposals-based methods mainly include R-CNN [17], Fast R-CNN [18], Faster R-CNN [19], Mask R-CNN [20] are some region-based proposals algorithms. For the regression/classification methods the different YOLO versions [21, 22, 23] and SSD [24] are some algorithms that give efficient results.

3.1. Region Proposal methods.

There are a large number of proposed methods in order to detect object in natural images or real-time videos using CNN [17, 18, 25, 26, 27]. The region proposals algorithms are two stages methods, leverage a proposal network to find objects and then use a second network to fine-tune these proposals and output a final prediction.

3.1.1. R-CNN.

The Region-based Convolutional Neural Networks (R-CNN) has been proposed in 2014 [17] for multi-object detection purpose. In the first stage, R-CNN split the input image in many regions, with different scales and positions, using Selective Search algorithm [28]. Each of these regions only contain one object. The second stage consists of feature extraction from these proposals using deep CNN, like VGGnet [29] and AlexNet [30]. In this stage, each of these region proposals are wrapped [30, 31] or cropped [17, 32] in fixed size and fed them to the CNN in order to extract a 4096-dimentional feature vector. Finally, these features will be fed into an SVM (Support Vector Machine) classifier to judge what kind of objects they are.

R-CNN has achieved impressive object detection accuracy, over traditional methods. However, it still has some notable disadvantages. It performs a CNN forward pass for each region proposal (without sharing computation), where the features are extracted from different region proposals and stored on the disk. It will take a long time to train the network as we would have to classify around 2000 region candidates per image. It takes around 2.5 days to train a very deep CNN, such as VGG16, for the VOC07 dataset using GPU [18]. The R-CNN is very slow, where detection time takes 47s per image. In addition, the extracted features need a huge storage memory, hundreds of Gigabytes. Also, the wrapping/ cropping preprocessing can cause unwanted geometric distortion or information loss [33], which reduces efficiency.

These limits make R-CNN not suitable for real-time object detection, especially when we use it for drone, which has limited hardware, both in memory and computation power, in the case of on-board embedded systems.

3.1.2. Fast R-CNN.

The next advancement in region-based CNN comes with the fast R-CNN, which is proposed in 2015 [18] to improve some of the drawbacks of R-CNN. Instead of processing each region of interest individually through a CNN, this architecture runs the entire input image through a CNN just once. The image goes through a series of convolutional and pooling layers and at the end of these layers we got a stack of feature maps. We still need to identify regions of interest, but instead of cropping the original image we project these proposals into the smaller feature map layer. Each region corresponds to larger region in the original image. A fixed-length feature vector is extracted, from the feature map, using region of interest pooling layer (RoI layer). RoI pooling layer convert the extracted features, in a valid RoI, into small feature maps of fixed size [34]. Then, we can grab the selected regions in the feature map and feed them one by one into a series of fully connected layers that generate a class foreach of these different regions.

As shown in table 1, this network is faster than its predecessor (R-CNN) with a higher accuracy [18, 35], but it still kind of slow when faced with the test image, which has to generate the region proposals and still looking to regions that do not contain any object. These limits make Fast R-CNN also not suitable for real-time drone-based object detection.

3.1.3. Faster R-CNN.

To speedup the time it takes to run a test image through a network and detect all the objects in it we want to decrease the time it takes to form region proposals. For this, the Faster R-CNN [19] comes to fix this problem.

Faster R-CNN learns to come-up with its own region proposals. It takes an input image, runs it through a CNN up until a certain convolutional layer just like Fast R-CNN. However, this time, uses the produced feature map as input into a separate Region Proposal Network (RPN). It predicts its own regions from the features produced inside the network. If an area in the feature map is rich and detected edges or other features it is identified as a region of interest. Then, this part of the network does a quick binary classification. For each RoI check whether or not that region contains an object. If it does then the region will continue on and go through the classification steps. However, if it does not contain an object, then the proposal is discarded. Both region proposal generation and object detection tasks are all done by the same CNN [33] (Fig. 1), in which the RPN module shares the same convolutional features with the fast R-CNN detection network; hence, it enables nearly cost-free region proposal generation [35]. With such design, object detection is much faster.

In the last step, the proposals are passed to a fully connected layer, which has a softmax layer and a linear regression layer at its top, to classify and output the bounding boxes for objects.

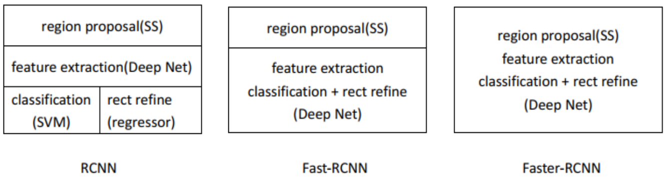

Figure 1: The improvement from R-CNN to fast R-CNN to faster R-CNN [33].

By eliminating the analysis of non-object regions, this model is the fastest of all the region-based CNNs that we have presented (Tab. 1).

Table 1: Comparison between training and testing time for R-CNN, Fast R-CNN and Faster R-CNN.

	R-CNN	Fast R-CNN	Faster R-CNN
Training time (h)	84	9.5	4.5
Train speedup	1x	8.8x	18.6x
Test time per image (s)	47	0.32	0.2
Test speedup	1x	147x	235x
mAP	66	66.9	69.9

For all of the Region Proposal methods, including Faster R-CNN, we have aimed to improve the speed of the object detection models by reducing the time it takes to generate and decide on region proposals. In the next sections we will present two methods that does not rely on region proposals to work and provide a high-speed object detection.

3.2. Regression/Classification Based Framework.

In the previous sections we presented a number of region-based methods for detecting multiple objects in a scene. Architectures like faster R-CNN are accurate, but they are still not fast enough to run in real-time. In this section, we will present one stage algorithms, which predict object classes and locations in a single step.

Drones must be able to identify whether one or more objects of interest exist in a scene with high-speed. In general, single-stage detectors are significantly faster than two stages detectors, and give us a real-time performance. YOLO and SSD are the most used algorithms for real-time object detection, where many works are based on these two architectures. They are prerequisite for any autonomous drones to fly around safely. They need to be able to recognize all the objects around them. On top of that, this detection has to happen in near real-time. It is not just an interesting computer science problem, but it is now a safety requirement.

3.2.1. SSD.

SSD [18] is a single-shot detector, which can do everything in one single shot, that's why it's called the single shot detection algorithm. It aims to combine the speed of the first generation of YOLO with the accuracy of Faster R-CNN detectors. It extends the YOLO CNN architecture by adding six more auxiliary convolutional layers after the VGG16, as shown in figure 2, among these layers, five of them will be added for object detection.

SSD predicts a fixed number of bounding boxes and scores for the presence of object class instances in each box, followed by a Non-Maximum Suppression (NMS) step to produce the final detection. In order to handle objects with different sizes, the network combines predictions from multiple feature maps with different resolutions.

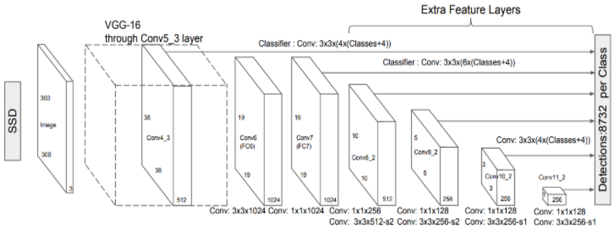

Figure 2: Architecture of Single Shot Multi-Box detector

SSD300 runs at 59 FPS, which makes it faster than two stages methods and YOLO. With such a speed, SSD could be executed in real-time. It presents a higher detection quality than Faster R-CNN and YOLO, achieving a mAP of around 74.3% on VOC 2007 [18, 33, 39].

3.2.2. YOLO.

YOLO stands for You Only Look Once [21]. It is a real-time object detection algorithm that avoid spending too much time on generating region proposals. Instead of locating objects perfectly, YOLO prioritizes speed and recognition. It employs a single CNN for the whole task instead of the RPN + classifier design of the Faster R-CNN. It is designed for real-time predictions.

Firstly, YOLO model takes an image as input. Then, it divides this image into an S × S grid. We can assign output vectors to each grid cell. Each cell has an associated vector that tells us: - if an object is in that cell; - the class of that object; and - the predicted bounding box for that object. In this way, the bounding box coordinates do not have to be contained within a grid cell (Fig. 3).

YOLO can process images in real-time at 45 FPS for the enhanced version, and at 155 FPS for the fast version [21], which means we can process streaming video in real-time. Fast YOLO uses less convolutional layers and fewer number of filters, which make it faster, but less accurate. It is fast, but also not suitable for small object detection.

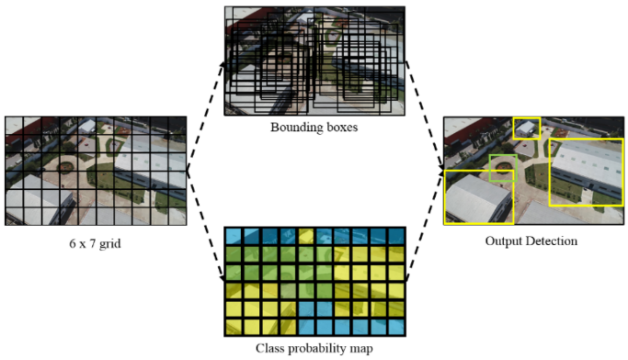

Figure 3: YOLO model.

The Faster R-CNN algorithm is more accurate than YOLO, especially for small objects, although YOLO tends to be faster. The detecting targets, in the field of drones, are always small, on which YOLO's performance is unsatisfactory [39].

Later, R. Joseph has proposed a series of improvements on the original YOLO and has proposed its versions 2 and 3 [22, 23]. These versions improve the detection accuracy while keeping a very high detection speed (Tab. 2) [43].

YOLO and its versions are used for many drone's applications in order to detect objects. In [44], the authors improved a YOLO model to better accommodate detection of drones. A methodology based on YOLOv3 was proposed in [45] to make object detection in UAV imagery both fast and super accurate. The authors in [46] used a lightweight architecture based on YOLOv3 in order to detect drones and small intruder.

4. On-board deep learning-based object detection.

There are two cases for drone-based object detection. The first one is using on-ground work stations with powerful GPUs. However, the second case is to implement the object detection algorithms on the drone itself.

In the case of on-ground object detection, the drone collects data via its camera and transmit them to a desktop computer for analysis. This gives to drones the ability to save power by off-loading compute intensive operations. However, the wireless transmission takes a long time in sending the collected data, which mean an additional cost added to the latency of the system.

Table 2: Comparison of accuracy and speed on PASCAL VOC 2007.

	mAP (%)	FPS	Real-time
Fast R-CNN	70.0	0.5	No
Faster R-CNN (VGG16)	73.2	7	No
Faster R-CNN (ZF)	62.1	18	No
Faster R-CNN (ResNet-101)	76.4	5	No
Fast YOLO	52.7	155	Yes
YOLO	63.4	45	Yes
YOLO (VGG16)	66.4	21	No
YOLOv2	78.6	40	Yes
SSD300	74.3	59	Yes
SSD512	76.8	19	No

The requirement of high computational resources for video processing poses a challenge in mapping deep learning-based algorithms on low-cost and low-power computing platforms. In order to solve these problems, developing deep learning-based object detection algorithms that are suitable for on-board real-time object detection is important. For that purpose, lightweight versions of the deep learning-based object detection algorithms are used on limited hardware.

All the previous methods are based on CNNs with a large number of layers and filters to make them accurate. They are used with high-performance computers on the ground, and they are not suitable for on-board systems. In order to solve these problems, developing CNNs that are suitable for on-board real-time object detection is important to reduce model parameters and accelerate their calculations. For that purpose, lightweight versions of the CNNs architectures, with a reduced number of layers and filter are used on limited hardware.

The lightweight versions improve the power consumption. The battery life and flight time could be increased using lightweight models. An additional 0.5 to 1 W power is required to operate cooling system for each watt of power dissipated in a computing equipment [47]. Moreover, a low-power computing system can reduce thermal problems and cooling requirement, which is very important in the field of autonomous drones.

Several lightweight CNNs are proposed in research papers for use on mobile platforms with limited resources. In [49], the authors proposed MobileNet, which is an efficient CNN architecture for mobile and embedded vision systems. MobileNet is not usually accurate as the bigger CNN architectures. However, MobileNet shines in the resource/ accuracy trade-off. It gives high accuracy only with limited resources. MobileNet v2 [36] is an updated version of MobileNet v1, which makes it more efficient and powerful in terms of accuracy and speed.

SqueezeNet [38] is another lightweight CNN architecture that achieves the accuracy of AlexNet CNN on ImageNet dataset with 50x fewer parameters (Tab. 3). SqueezeNet can be 500x smaller than AlexNet using compression techniques.

Table 3: Comparison of accuracy and reduction in model size between AlexNet and SqueezeNet.

	Reduction in model size	Top 1 ImageNet accuracy	Top 5 ImageNet accuracy
AlexNet	1x	57.2	80.3
SqueezeNet	50x	57.5	80.3

ShuffleNet [50] is a computation-efficient lightweight CNN architecture for mobile devices with limited computing power. It provides better performance than MobileNet on the tasks of ImageNet classification and COCO detection. ShuffleNet achieves 13× actual speedup over AlexNet architecture, on an ARM-based mobile device, while maintaining comparable accuracy.

PelleNet [51] is an efficient architecture for embedded platforms. It achieves better accuracy and 1.8x faster over the two versions of MobileNet, on ImageNet ILSVRC 2012 using Nvidia Jetson TX2. Meanwhile, the PeleeNet model size is smaller than MobileNet by 66%. PeleeNet achieves higher accuracy and speed than MobileNet and MobileNet v2 on Nvidia Jetson TX2 (Tab. 4).

Table 4: Comparison of deferent lightweight CNN architectures performance on Nvidia Jetson TX2.

	MACs (million)	# parameters (million)	Top 1 accuracy
MobileNet V1	569	4.24	70.6
MobileNet V2	300	3.47	71.8
ShuffleNet 2x (g = 3)	524	5.2	73.7
PeleeNet	508	2.8	72.6

Several authors used lightweight architectures as backbone networks for various object detection frameworks. Faster R-CNN was used for many drone-based object detections. In [37], the authors combine a tiny version of Faster R-CNN for people detection with a KCF tracker for single object tracking on a drone. It is a lightweight version used for on-board object detection systems. They achieved real-time performance at 71 FPS for tracking and 1.6 FPS for detection on Nvidia Jetson TX1 and 182 FPS for tracking and 6 FPS for detection on a desktop GTX980. The GTX980 is four time faster than TX1, but consumes 20x more power.

Y. Yang et al. [33] combine Faster R-CNN with the frame difference method in order to detect and track moving objects for a naval unmanned aircraft system. However, all vision-navigation systems in the proposed method are placed on the ground.

A large number of object detection applications for drones are based on SSD algorithm. In [51], the authors combined PeleeNet with SSD for object detection. As shown in Table 5, the object detector based on PeleeNet architecture presents higher accuracy than the other architectures.

Table 5: Comparison of deferent lightweight CNN-based object detectors performance on PASCAL VOC 07 + 12.

	Input dimension	MACs (million)	Data	mAP %
Tiny-YOLO v2	**416*416**	**3490**	**07 + 12**	**57.1**
SSD + MobileNet	**300*300**	**1150**	**07 + 12**	**68**
SSD + PeleeNet	**304*304**	**1210**	**07 + 12**	**70.9**
SSD + MobileNet	**300*300**	**1150**	**07+12+COCO**	**72.7**
SSD + PeleeNet	**304*304**	**1210**	**07+12+COCO**	**76.4**

SSD + PeleeNet of [51] achieves 76.4% mAP on VOC07 and 22.4 mAP on MS COCO dataset, achieving 23.6 FPS on iPhone 8 and 125 FPS on NVIDIA Jetson TX2. It is 13.6x lower computational than YOLOv2 and 11.3 smaller. Also, it provides higher accuracy on COCO dataset.

In [40], the authors proposed SSDLite to detect the potential receiver for a delivery using computer vision techniques. It looks for a specific person in a limited area. In low resolution images, this method achieved 92% of accuracy from a 10 meters high and 5 meters horizontal distance. Also, the authors of [41], proposed another SSD-based model for vehicles detection, called UAV-Net. It operates in real time on a Jetson TX2 platform and achieve a high accuracy.

In [46], the authors used a lightweight version of YOLO algorithm in order to detect possible drones on the wide-angle of the camera. Thanks to its speed and robustness, it becomes a popular choice. Vandersteegen et al. [45] proposed an on-board object detector for drones that detects common objects like humans and vehicles well from both frontal and top-down views. The implemented their algorithm on Nvidia Jetson TX2 and Xavier. In [48], the authors decrease the number of filters, used in Tiny-YOLO, in each layer to reduce the number of operations per input. Hence, they got a smaller network which can lead to a faster detection. In [49], the authors proposed MobileNet-SSD, which comprises depthwise separable convolutions. It achieves a significant detection accuracy on COCO dataset.

5. Conclusion.

Drones are one of the most promising systems that can utilize deep learning. They are becoming an attractive solution for a large number of applications. Object detection is one of them, which in its turn applied for a wide range of applications like rescuing, surveillance, autonomous driving. Under the above deployments, drones are responsible for searching, collecting and sending, in real-time, object information.

We saw that deep learning-based object detection algorithms achieves a high accuracy and speed for real-time issues. Drones need accurate and fast object detection methods that are suitable for its low-power and low-processing. Lightweight CNN architectures give us the possibility to execute these accurate algorithms on a limited hardware resources embedded systems.

As future work, we want to implement different lightweight deep learning algorithms on a Pixhawk 2.1, which is based on Cortex M4F processor.

References

[1] S. Sawadsitang, D. Niyato, P. S. Tan and P. Wang, "Joint Ground and Aerial Package Delivery Services: A Stochastic Optimization Approach," IEEE Transactions on Intelligent Transportation Systems, Volume 20, Issue 6, pp. 2241-2254, 2019.

[2] G. Ding, Q. Wu, L. Zhang, Y. Lin, T. A. Tsiftsis and Y. D. Yao, "An Amateur Drone Surveillance System Based on the Cognitive Internet of Things," IEEE Communications Magazine, Volume 56, Issue 1, pp. 29-35, 2018.

[3] H. Niu, N. G. Prelcic and R. W. Heath, "A UAV-Based Traffic Monitoring System - Invited Paper," IEEE 87th Vehicular Technology Conference (VTC Spring), 2018.

[4] C. Kyrkou, S. Timotheou, P. Kolios, T. Theocharides and C. G. Panayiotou, "Optimized vision-directed deployment of UAVs for rapid traffic monitoring," IEEE International Conference on Consumer Electronics (ICCE), 2018.

[5] J. N. McRae, C. J. Gay, B. M. Nielsen and A. P. Hunt, "Using an Unmanned Aircraft System (Drone) to Conduct a Complex High-Altitude Search and Rescue Operation: A Case Study," Wilderness & Environmental Medicine, Volume 30, Issue 3, pp 287-290, 2019.

[6] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 1, pp. 39–51, 2002.

[7] C. Wojek, P. Dollar, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 4, p. 743, 2012.

[8] X. Zhao, F. Pu, H. Wang, H. Chen and Z. Xu, "Detection, Tracking, and Geolocation of Moving Vehicle from UAV Using Monocular Camera," IEEE Access, vol. 7, pp. 101160 - 101170, 2019.

[9] G. E. Moore, "Cramming More Components onto Integrated Circuits," In proc. of the IEEE, Volume 86, Issue 1, pages 82-85, 1988.

[10] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," International Journal of Computer Vision 57(2), pp. 37–154, 2004.

[11] C. L. Azevedo, J. L. Cardoso, M. Ben-Akiva, J. P. Costeira, and M. Marques, "Automatic vehicle trajectory extraction by aerial remote sensing," Procedia - Social and Behavioral Sciences, vol. 111, pp. 849 – 858, 2014.

[12] X. Chen, S. Xiang, C. L. Liu, and C. H. Pan, "Vehicle detection in satellite images by parallel deep convolutional neural networks," In 2nd IAPR Asian Conference on Pattern Recognition, pp. 181–185, 2013.

[13] N. Audebert, B. Le Saux, and S. Lefvre, "Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images," Remote Sensing, vol. 9, no. 4, 2017.

[14] S. Qu, Y. Wang, G. Meng and C. Pan, "Vehicle Detection in Satellite Images by Incorporating Objectness and Convolutional Neural Network," Journal of Industrial and Intelligent Information Vol. 4, no. 2, 2016.

[15] T. Zhao and R. Nevatia, "Car detection in low resolution aerial images," in Proc. International Conference on Computer Vision, vol. 1, pp. 710-717, 2001.

[16] H. Grabner, T. Nguyen, B. Gruber, and H. Bischof, "On-line boosting-based car detection from aerial images," J. Photogrammetry and Remote Sensing, vol. 63, no. 3, pp. 382-396, 2008.

[17] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[18] R. Girshick, "Fast r-cnn," in ICCV, 2015.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards realtime object detection with region proposal networks," in NIPS, pp. 91–99, 2015.

[20] K. He, G. Gkioxari, P. Dollar, and R. B. Girshick, "Mask r-cnn," in ICCV, 2017.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in CVPR, 2016.

[22] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," arXiv:1612.08242, 2016.

[23] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in ECCV, 2016.

[25] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV, pp. 346–361. 2014.

[26] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In CVPR, pp. 2155–2162, 2014.

[27] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee. Improving object detection with deep convolutional networks via bayesian optimization and structured prediction. In CVPR, pp. 249–258, 2015.

[28] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. IJCV, 2013.

[29] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv:1409.1556, 2014.

[30] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in NIPS, 2012.

[31] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional neural networks,” arXiv:1311.2901, 2013.

[32] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” arXiv:1310.1531, 2013.

[33] Y. Yang, H. Gong, X. Wang and P. Sun, “Aerial Target Tracking Algorithm Based on Faster R-CNN Combined with Frame Differencing,” aerospace, 2017.

[34] S. Khan, H. Rahmani, S. A. A. Shah and M. Bennamoun, “a guide to convolutional neural networks for computer vision,” Publisher: Morgan & Claypool, 2018.

[35] J. Han, D. Zhang, G. Cheng, N. Liu and D. Xu, “Advanced Deep-Learning Techniques for Salient and Category-Specific Object Detection: A Survey,” IEEE Signal Processing Magazine, vol. 35, iss. 1, pp. 84-100, 2018.

[36] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, “Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation,” arXiv:1801.04381, 2018.

[37] S. Han, W. Shen and Z. Liu, “Deep Drone: object detection and tracking for smart drones on embedded system,” Stanford University, 2015.

[38] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size,” arXiv:1602.07360, 2016.

[39] Z.-Q. Zhao, P. Zheng, S.-t. Xu and X. Wu, “Object Detection with Deep Learning: A Review,” arXiv:1807.05511v2, 2019.

[40] D. Safadinho, J. Ramos, R. Ribeiro, V. Filipe, J. Barroso and A. Pereira, “System to Detect and Approach Humans from an Aerial View for the Landing Phase in a UAV Delivery Service,” 10th International Symposium on Ambient Intelligence, pp. 162-170, 2019.

[41] T. Ringwald, L. Sommer, A. Schumann, J. Beyerer and R. Stiefelhagen, “UAV-Net: A Fast Aerial Vehicle Detector for Mobile Platforms,” CVPR, 2019.

[42] Q. Hu, Q. Duan, Y. Mao, X. Zhou and G. Zhou, “DiagonalNet: Confidence Diagonal Lines for the UAV Detection,” In IEEJ Transactions on Electrical and Electronic Engineering, 2019.

[43] F. Fang, M. Tambe, B. Dilkina and A. J. Plumptre, “Automatic Detection of Poachers and Wildlife with UAVs,” In book: Artificial Intelligence and Conservation, pp.77-100, 2019.

[44] M. Wu, W. Xie, X. Shi, P. Shao and Z. Shi, “Real-Time Drone Detection Using Deep Learning Approach,” In book: Machine Learning and Intelligent Communications, pp.22-32, 2018.

[45] M. Vandersteegen, K. V. Beeck and T. Goedemé, “Super accurate low latency object detection on a surveillance UAV,” 16th International Conference on Machine Vision Applications, 2019.

[46] E. Unlu, E. Zenou, N. Riviere and P.-E. Dupouy, “Deep learning-based strategies for the detection and tracking of drones using several cameras,” IPSJ Transactions on Computer Vision and Applications, 2019.

[47] S. Mittal, “Power management techniques for data centers: a survey,” Technical Report, Oak Ridge National Laboratory, USA, 2014.

[48] C. Kyrkou, G. Plastiras, T. Theocharides, S. I. Venieris and C.-S. Bouganis, “DroNet: Efficient Convolutional Neural Network Detector for Real-Time UAV Applications,” arXiv:1807.06789v1, 2018.

[49] 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:1704.04861v1, 2017.

[50] X. Zhang, X. Zhou, M. Lin and J. Sun, “ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices,” arXiv:1707.01083, 2017.

[51] R. J. Wang, X. Li and C. X. Ling, “Pelee: A Real-Time Object Detection System on Mobile Devices,” arXiv:1804.06882v3, 2019.