11th International Defense and Homeland Security Simulation Workshop
18th International Multidisciplinary Modeling & Simulation Multiconference

ISSN 2724-0363 ISBN 978-88-85741-63-8 © 2021 The Authors.
doi: 10.46354/i3m.2021.dhss.009

Drone detection with YOLOv5

Jiri Kralicek[1,*]

[1]University of Defence, Kounicova 65, Brno, 662 10, Czech Republic

[*]Corresponding author. Email address: kralicek.jirka@gmail.com

Abstract

We propose a fast and accurate method for visual drone detection based on YOLOv5 architecture providing state-of-the-art performance. The proposed method aims to drone detection in combat and real-world environments for military use based on visual detection in the visible and infrared spectrum. The method provides precision/recall of 99.1/98.5% and 99.0/95.3% for RGB and infrared videos from the AntiUAV dataset.

Keywords: Drone detection; object detection; computer vision

1. Introduction

With the massive proliferation and popularity of drones, this technology has become affordable and interesting for masses. The technology that previously only the military could afford is now used by various entities in many use cases such as monitoring, guarding, distribution, etc. However, this technology has also been adopted for combat use by terrorists, e.g. for bomb attacks. Such an attack does not have to be undertaken by only one drone, but also by a whole swarm, making the defence more difficult. Therefore, fast autonomous drone detection in combat and real-world environments becomes a very important ability, which in combination with other elements, allows to eliminate risks, save military equipment and protect personnel.

In this paper, we utilize fast state-of-the-art object detector YOLOv5 for drone detection in visible and infrared spectrum. Because we assume restricted power-consumption in real deployment, we present results for both server and embedded system on the AntiUAV dataset providing the best similarity to the real-world and combat deployment from all publicly available datasets. The paper also emphasizes one of the main current issues in visual drone detection, which makes a very difficult comparison of different methods.

2. State of the art

In this section, state-of-the-art drone detection methods are described. The section focuses primarily on drone recognition in the vision domain.

Drone detection is on interest in numerous domains leading to various detection approaches, i.e. acoustic (Liu et al., 2017, 2018; Al-Emadi et al., 2019; Anwar et al., 2019), radar (Shin et al., 2017; Ochodnický et al., 2017; Nuss et al., 2017; Jian et al., 2018; Jarabo-Amores et al., 2018; de Quevedo et al., 2018), radio-frequency (Solomitckii et al., 2018; Rydén et al., 2019) and visual.

Visual detection is generally based on a deep learning approach utilizing regression-based (Girshick, 2015) or segmentation-based approach (He et al., 2017). For example, Aker and Kalkan (2017) utilized YOLOv2 (Redmon and Farhadi, 2017) for bounding box prediction and proposed a custom artificial dataset. The dataset consists of publicly available videos of coastal areas with randomly placed drones.

Lee et al. (2018) proposed a two-level approach to detect and identify drone. In the first step, Haar Feature-based Cascade Classifier is applied to obtain 2D area on the frame, where the drone is located. In the second step, the image is passed to the CNN-based classifier to identify the type of drone. The model was trained and

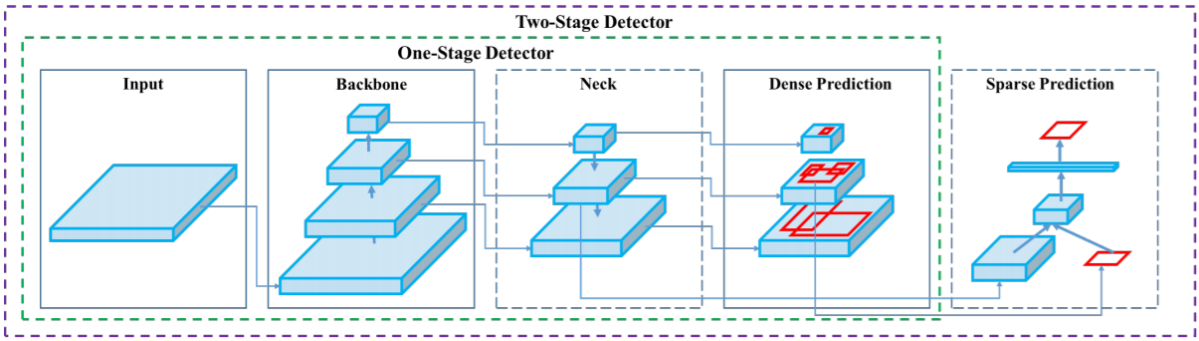

Figure 1. General object detection architecture overview. Utilized YOLOv5 model is one-stage regression-based detector. Courtesy of Bochkovskiy et al. (2020).

tested on the custom dataset obtained from the internet and manually cropped. Drones are very dominant, focused and clear in the dataset images.

Carrio et al. (2018) utilized YOLOv2 to predict boxes containing a drone and a confidence value for each bounding box based on a depth map. In the next step, the 2D point in each bounding box is chosen as actually belonging to the drone. The chosen point is then re-projected to 3D to get the actual drone relative position. The paper proposed a dataset of 6k synthetic depth maps of drones based on Unreal Engine.

Jin et al. (2019) proposed drone detection and pose estimation based on relation graph networks. Inspired by Mask R-CNN, candidate object bounding boxes are proposed by Region Proposal Network (RPN). Class, bounding box offset and keypoints are predicted in parallel by separate convolutional layers. A custom dataset was proposed.

Nalamati et al. (2019) experimented with three architectures for drone detection in long-range surveillance videos. The combination of Faster R-CNN and ResNet-101 provided the best performance on the drone-vs-bird dataset (Coluccia et al., 2017).

Svanstrom et al. (2020) employed multimodal detection, utilizing YOLOv2 for RGB and infrared videos detection. The authors also proposed a novel video dataset containing 650 annotated infrared and visible videos of drones, birds, aeroplanes and helicopters.

3. Method

In this section, we describe the used model for drone detection, a dataset for training and evaluation, training details, experiments and implementation, and evaluation metrics.

3.1. Model

With regard to the inference time (speed) required in real deployment, we utilized YOLOv5[1] model (Jocher et al., 2021) for drone detection. The model provides state-of-the-art performance while maintaining fast inference. The model is a one-stage regression-based detector, see Fig. 1, meaning that objects are predicted by bounding boxes that take the form of quadrangles. YOLOv5 is a natural extension of YOLOv3 (Redmon and Farhadi, 2018).

The YOLOv5 model utilizes Leaky ReLU and Sigmoid Linear Unit (SiLU) functions as non-linearities, and consists of the three parts: backbone, neck, head.

Inspired by Cross Stage Partial Networks (Wang et al., 2020), YOLOv5 utilizes CSP Bottleneck as a backbone. The CSP models are based on the DenseNet (Tan et al., 2020) architecture, see Fig. 2, addressing vanishing gradient, and mitigating duplicate gradient problem. The backbone provides multi-level feature extraction for further processing.

The neck is formed of a series of layers to mix and combine image features. These features are passed to the head for the final prediction. The model adopted the neck architecture from YOLOv4 (Bochkovskiy et al., 2020), where the authors conducted several experiments with different architectures, see Fig. 3, and the PANet architecture has been chosen as the most suitable.

Based on the features, the head generates final output vectors with class probabilities, objectness scores, and bounding boxes. For bounding box prediction, anchor boxes (Redmon and Farhadi, 2017) are used. Predicted boxes are described by the central point, width and height.

3.2. Dataset

We used AntiUAV (Jiang et al., 2021) dataset for Unmanned Aerial Vehicle (UAV) tracking for training

1 https://github.com/ultralytics/yolov5

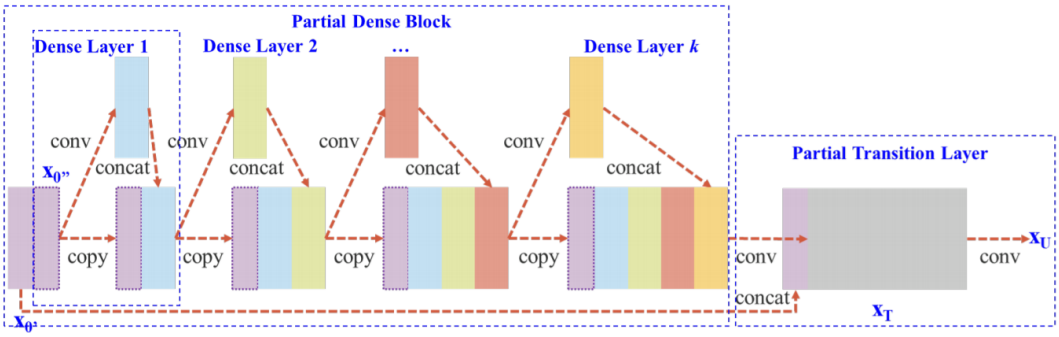

Figure 2. Architecture overview of Cross Stage Partial DenseNet used as the backbone of the YOLOv5 model. Courtesy of Tan et al. (2020).

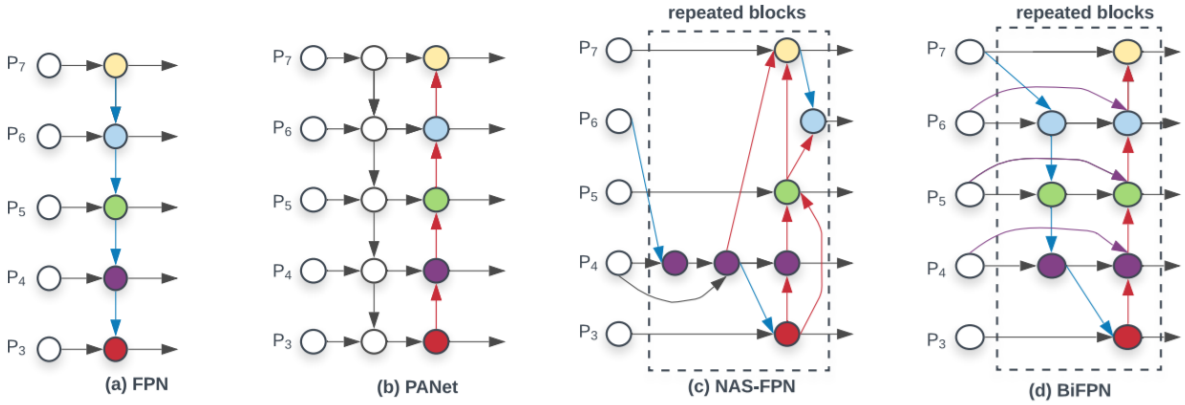

Figure 3. Example of different architectures evaluated in the YOLOv4 as a neck. YOLOv5 utilizes PANet as the neck. Courtesy of Tan et al. (2020).

and testing, due to the closest similarity to the real deployment from all publicly available datasets. However, due to its primary use for tracking evaluation, only the *test-dev* part of the dataset is provided with the annotation. Thus, we used only the subset, which is in the paper referred to as a dataset. We performed random sampling on the dataset with 70/30% split to obtain training/testing subsets.

Dataset consists of 100 high-quality video sequences in visible (RGB) an infrared (IR) spectrum, spanning multiple occurrences of multi-scale UAVs - three sizes (tiny, middle, big). Annotation consists of a ground-truth bounding box and information on whether the object exists in the frame.

3.3. Training

The training was conducted in 20 epochs for each variant, stochastic gradient descent (SGD) was used as an optimizer. Data augmentation was employed in the training.

As an objective function, total loss compounding of three losses is used:

$$L_{total} = L_{reg} + L_{obj} + L_{cls} \tag{1}$$

where L_{reg}, L_{obj}, L_{cls} denotes regression loss, objectness loss and classification loss, respectively.

Regression loss utilizes Generalized Intersection over Union (GIoU) (Rezatofighi et al., 2019) to describe localization error. GIoU is described as follows:

$$GIoU = \frac{|A \cap B|}{|A \cup B|} - \frac{\left|\frac{C}{A \cup B}\right|}{|C|} \tag{2}$$

Objectness is a measure of the probability that an object exists in a proposed region of interest. Classification loss describes the classification error of the given classes. Both losses utilize Binary Cross-Entropy with Logits Loss described as follows:

$$\ell(x, y) = L = \{\ell_1, \ldots, \ell_N\}^\top,$$
$$\ell_n = -w_n[y_n \cdot log\sigma(x_n) + (1 - y_n) \cdot log(1 - \sigma(x_n))] \tag{3}$$

where N, x, y, w denotes batch size, predicted output, ground-truth label and weight, respectively.

3.4. Experiments and implementation

It is customary to use out-of-shelf GPU servers for the evaluation of object detection models, however, we cannot expect the use of high-end or professional GPUs in the real deployment, due to their

Table 1. Results of YOLOv5 models on AntiUAV dataset in visible spectrum (RGB), and infrared (IR), the threshold was set to 0.5. AP_{50}, AP_{50-95} denote average precision for threshold 0.5 and set of threshold 0.5-0.95 with step 0.05, respectively.

	IR				RGB			
Model	Precision	Recall	AP_{50}	AP_{50-95}	Precision	Recall	AP_{50}	AP_{50-95}
small	0.991	0.985	0.990	0.569	0.990	0.953	0.983	0.632
medium	0.989	0.982	0.989	0.565	0.989	0.927	0.978	0.621
large	0.988	0.979	0.986	0.573	0.991	0.927	0.981	0.629
extra large	0.990	0.983	0.990	0.568	0.992	0.930	0.985	0.633

Table 2. Inference time results of YOLOv5 models on AntiUAV dataset, proportionally resized to resolution 640 pixels for the longer edge. Results shown for graphic card Nvidia 2080Ti (GPU), processor AMD EPYC 7351P (CPU) and Jetson AGX Xavier (ES). *Speed* represents inference time cost in milliseconds, *FPS* Frames per Second.

Model	GPU_{speed}	GPU_{FPS}	CPU_{speed}	CPU_{FPS}	ES_{speed}	ES_{FPS}
small	12.0	83.3	143.8	7.0	23.3	42.9
medium	15.5	64.5	290.6	3.4	33.3	30.0
large	19.0	52.6	511.8	1.9	50.9	19.6
extra large	23.3	42.9	869.4	1.2	84.9	11.8

immense power consumption. Thus, we conducted several experiments on two platforms - server and embedded system. We utilized four different sizes of the YOLOv5 model – small, medium, large and extra-large – and performed experiments to obtain qualitative results a time costs. Subsection 3.5 further describes metrics used for evaluation.

For possible reproduction of our results, we provide implementation details for individual platforms and used hardware. The model was implemented in PyTorch 1.7, Ubuntu 18.04 LTS and CUDA version 10.2 was used for both server and embedded system. Server utilizes AMD EPYC 7351P (CPU), NVIDIA 2080Ti (GPU) and 32GB RAM. As an embedded system, we used Jetson AGX Xavier with JetPack 4.5.

For performance and power consumption comparison between different components, we indicate the consumption of individual components: CPU – 170W, GPU – 250W, ES – 30W.

3.5. Metrics

For qualitative evaluation, Average Precision (AP), precision and recall were used. The Average Precision metric describes the area under the precision-recall curve. For AP, we used threshold 0.5 and set of thresholds from 0.5 to 0.95 with step 0.05, denoted as AP_{50} and AP_{50-95}, respectively. Precision and recall are defined as:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

where TP, FP, FN denotes True Positives, False Positive, False Negatives, respectively.

Arithmetic mean was employed for inference time measurement:

$$time = \frac{1}{n}\sum_{i=1}^{n} t_i \tag{6}$$

where n denotes number of images/frames in testing dataset and t_i time cost of the i-th image/frame.

4. Results and Discussion

This section provides the results of the proposed approach and emphasizes the main issues of the visual drone detection domain.

From the qualitative results shown in Tab. 1, it is clear that even the smallest YOLOv5 model has the capacity to detect drones of different sizes in the AntiUAV dataset in both IR and RGB. The overall differences between the results of the models are negligible, thus, it can be stated that the models achieve similar results. The RGB results show worse recall for the bigger models, which is most likely caused by worse generalization. Examples of predicted outputs for RGB and IR are shown in Fig. 4 and Fig. 5, respectively.

The results in Tab. 2 show that YOLOv5 models provide sufficient speed for both GPU and ES. The results also show that the gap between GPU and ES performance is acceptable, thus assuming limited power consumption for the deployment, ES provides the best trade-off between speed and consumption. On the other hand, it is common that small and fast models such as YOLO are run on the CPU. However, as expected, the results show that the CPU provides the worst power consumption/performance trade-off, making it undesirable for use in such a restricted environment. It is worthy of mention that the time cost does not represent only the inference time of the models, but also the non-maximum suppression (NMS).

The AntiUAV dataset is publicly available with the best similarity to the real-world deployment,

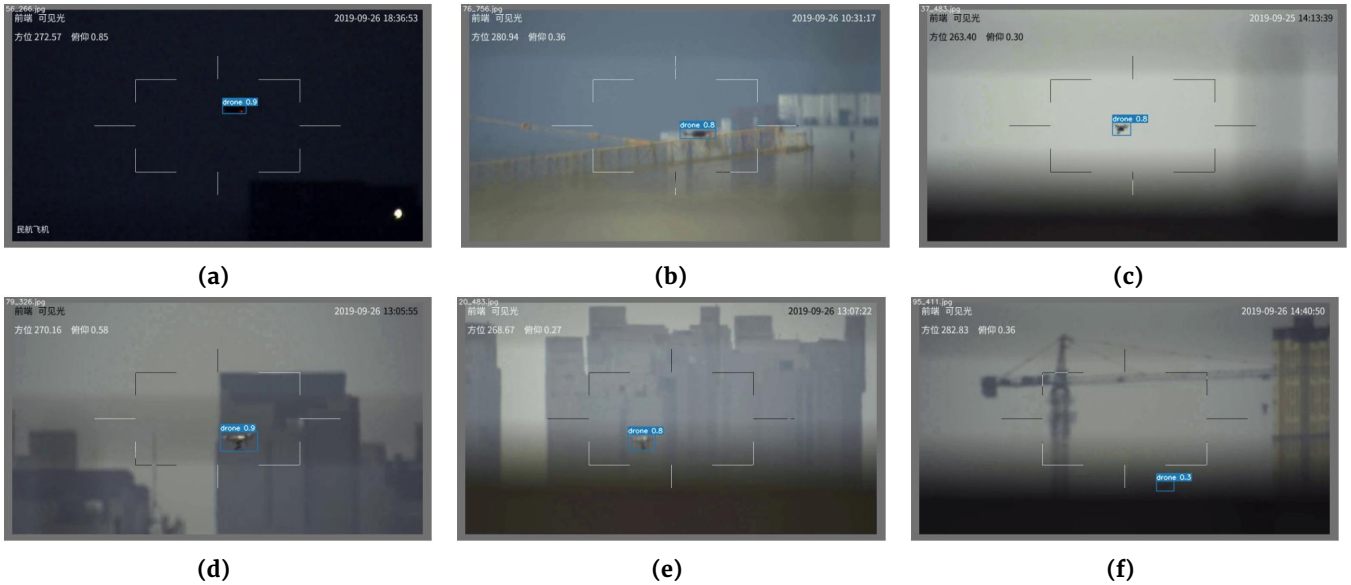

Figure 4. Examples of YOLOv5 (small) results on the RGB videos of the AntiUAV dataset. Predicted drone bounding boxes are indicated by a blue rectangles with probability written above the rectangle. Figures (a–e) show correctly recognized drones, figure (f) shows a false negative example caused by the low probability; the threshold was set up to 0.5 for the experiments.

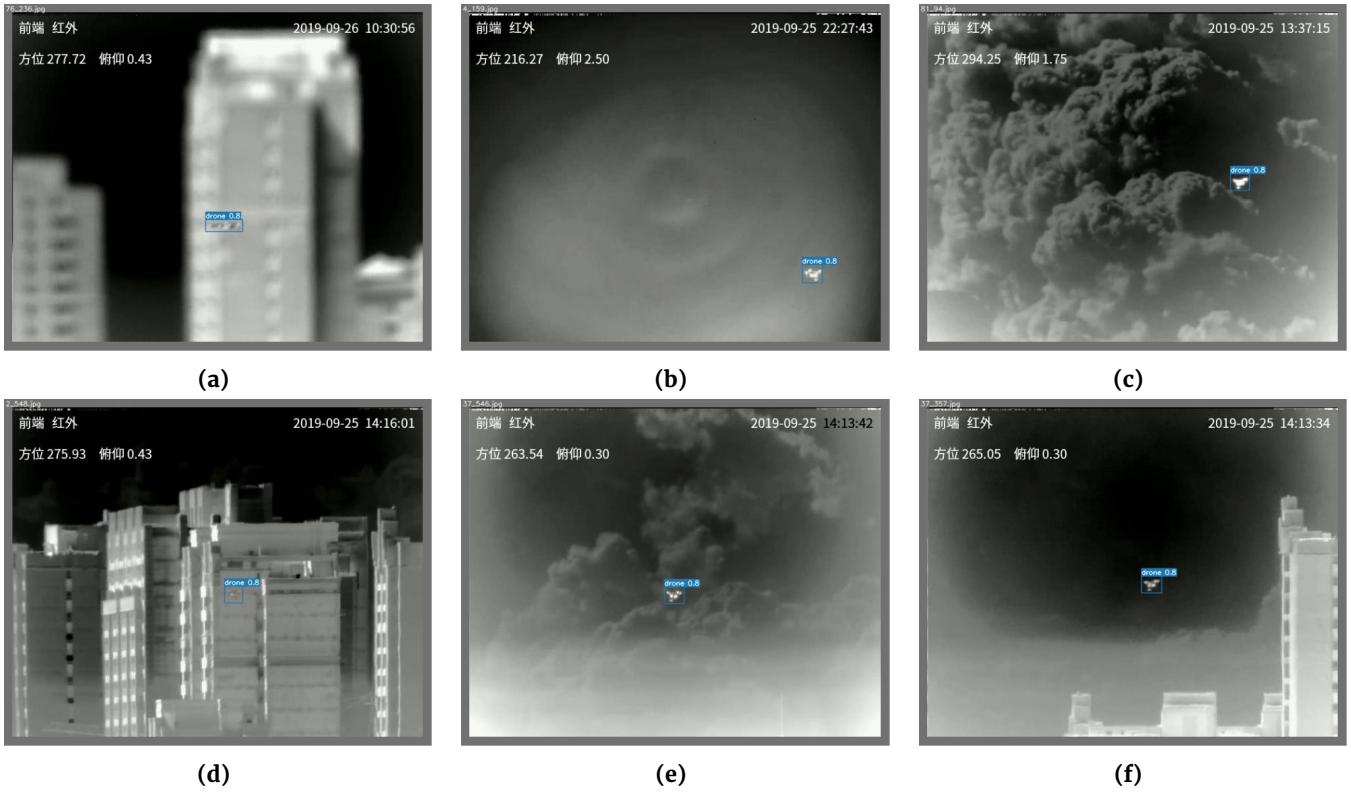

Figure 5. Examples of YOLOv5 (small) results on the IR videos of the AntiUAV dataset. Predicted drone bounding boxes are indicated by a blue rectangles with probability written above the rectangle.

however, it is still not sufficient for a deployment. The dataset provides only 100 videos with annotation, thus, it covers only a fraction of the real world. The dataset is formed by a majority of frames with drones, containing only one drone per frame, thus, false positives and false negatives are rare. Another problem is the absence of any confusing objects such as birds, helicopters, aeroplanes etc. These features of the dataset make drone detection easy for the state-of-the-art models, which can be seen in Tab. 1. It is worthy of mention that the AntiUAV dataset was proposed for tracking purposes.

From the description of other methods in the section 2, we can also emphasize that the majority of the methods use a custom dataset which clearly indicates the lack of a standardized benchmark. Absence of such a benchmark leads to a problem with evaluation and comparison between different methods.

To address the issues, a novel standardized publicly available dataset should be proposed to provide a challenging benchmark for further research in the visual drone detection domain.

Even with the missing benchmark for visual drone detection, based on the results presented in the paper, we assume that the proposed method provides state-of-the-art performance. Also, based on the result comparison between YOLOv5 and previous YOLO architectures in object detection domains with the standardized benchmarks (Jocher et al., 2021), we state that the proposed method outperforms all previous YOLO-based methods by a large margin.

5. Conclusion

In the paper, we utilized the YOLOv5 model of different sizes to detect drones in visible (RGB) and infrared (IR) videos. The models were trained and tested on the AntiUAV dataset, showing the capability for drone detection for combat and real-world environments while maintaining fast inference. The proposed method provides state-of-the-art results.

The paper have provided a comparison between out-of-shelf GPU server and embedded system (ES), showing ES (Jetson AGX Xavier) as a suitable system for the power-consumption restricted environment.

The paper also emphasized the need of challenging standardized publicly available benchmark for visual drone detection, for a fair comparison of different methods.

References

Aker, C. and Kalkan, S. (2017). Using deep networks for drone detection. *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6.

Al-Emadi, S., Al-Ali, A., Mohammad, A., and Al-Ali, A. (2019). Audio based drone detection and identification using deep learning. In *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, pages 459–464.

Anwar, M. Z., Kaleem, Z., and Jamalipour, A. (2019). Machine learning inspired sound-based amateur drone detection for public safety applications. *IEEE Transactions on Vehicular Technology*, 68(3):2526–2534.

Bochkovskiy, A., Wang, C.-Y., and Liao, H. (2020). Yolov4: Optimal speed and accuracy of object detection. *ArXiv*, abs/2004.10934.

Carrio, A., Vemprala, S., Ripoll, A., Saripalli, S., and Campoy, P. (2018). Drone detection using depth maps. pages 1034–1037.

Coluccia, A., Ghenescu, M., Piatrik, T., De Cubber, G., Schumann, A., Sommer, L., Klatte, J., Schuchert, T., Beyerer, J., Farhadi, M., Amandi, R., Aker, C., Kalkan, S., Saqib, M., Sharma, N., Khan, S., Makkah, K., and Blumenstein, M. (2017). Drone-vs-bird detection challenge at ieee avss2017. pages 1–6.

de Quevedo, Á. D., Urzaiz, F. I., Menoyo, J. G., and López, A. (2018). Drone detection with x-band ubiquitous radar. *2018 19th International Radar Symposium (IRS)*, pages 1–10.

Girshick, R. (2015). Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

Jarabo-Amores, M. P., Mata-Moya, D., del Hoyo, P. G., Bárcena-Humanes, J., Rosado-Sanz, J., Rey-Maestre, N., and Rosa-Zurera, M. (2018). Drone detection feasibility with passive radars. *2018 15th European Radar Conference (EuRAD)*, pages 313–316.

Jian, M., Lu, Z., and Chen, V. (2018). Drone detection and tracking based on phase-interferometric doppler radar. *2018 IEEE Radar Conference (RadarConf18)*, pages 1146–1149.

Jiang, N., Wang, K., Peng, X., Yu, X., Wang, Q., Xing, J., Li, G., Guo, G., Zhao, J., and Han, Z. (2021). Anti-uav: A large multi-modal benchmark for uav tracking. *arXiv preprint arXiv:2101.08466*.

Jin, R., Jiang, J., Qi, Y., Lin, D., and Song, T. (2019). Drone detection and pose estimation using relational graph networks. *Sensors (Basel, Switzerland)*, 19.

Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Changyu, L., Laughing, tkianai, yxNONG, Hogan, A., lorenzomammana, AlexWang1900, Chaurasia, A., Diaconu, L., Marc, wanghaoyang0106, ml5ah, Doug, Durgesh, Ingham, F., Frederik, Guilhen, Colmagro, A., Ye, H., Jacobsolawetz, Poznanski, J., Fang, J., Kim, J., Doan, K., and Yu, L. (2021). ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub

integration.

Lee, D., La, W. G., and Kim, H. (2018). Drone detection and identification system using artificial intelligence. *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1131–1133.

Liu, H., Qu, F., Liu, Y., Zhao, W., and Chen, Y. (2018). A drone detection with aircraft classification based on a camera array. *IOP Conference Series: Materials Science and Engineering*, 322:052005.

Liu, H., Wei, Z., Chen, Y., Pan, J., Lin, L., and Ren, Y. (2017). Drone detection based on an audio-assisted camera array. *2017 IEEE Third International Conference on Multimedia Big Data (BigMM)*, pages 402–406.

Nalamati, M., Kapoor, A., Saqib, M., Sharma, N., and Blumenstein, M. (2019). Drone detection in long-range surveillance videos. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6.

Nuss, B., Sit, L., Fennel, M., Mayer, J. R., Mahler, T., and Zwick, T. (2017). Mimo ofdm radar system for drone detection. *2017 18th International Radar Symposium (IRS)*, pages 1–9.

Ochodnický, J., Matoušek, Z., Babjak, M., and Kurty, J. (2017). Drone detection by ku-band battlefield radar. *2017 International Conference on Military Technologies (ICMT)*, pages 613–616.

Redmon, J. and Farhadi, A. (2017). Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *ArXiv*, abs/1804.02767.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., and Savarese, S. (2019). Generalized intersection over union.

Rydén, H., Redhwan, S. B., and Lin, X. (2019). Rogue drone detection: A machine learning approach. *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6.

Shin, D.-H., Jung, D.-H., Kim, D.-C., Ham, J.-W., and Park, S. (2017). A distributed fmcw radar system based on fiber-optic links for small drone detection. *IEEE Transactions on Instrumentation and Measurement*, 66:340–347.

Solomitckii, D., Gapeyenko, M., Semkin, V., Andreev, S., and Koucheryavy, Y. (2018). Technologies for efficient amateur drone detection in 5g millimeter-wave cellular infrastructure. *IEEE Communications Magazine*, 56:43–50.

Svanstrom, F., Englund, C., and Alonso-Fernandez, F. (2020). Real-time drone detection and tracking with visible, thermal and acoustic sensors. *ArXiv*, abs/2007.07396.

Tan, M., Pang, R., and Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10778–10787.

Wang, C.-Y., Liao, H., Yeh, I.-H., Wu, Y.-H., Chen, P.-Y., and Hsieh, J.-W. (2020). Cspnet: A new backbone that can enhance learning capability of cnn. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1571–1580.