Real-Time Drone Detection Using Deep Learning

Advanced Systems Laboratory Defence Research & Development Organisation

K. Sricharan[1(✉)] and M. Venkat[2]

[1] BML Munjal University, Kapriwas, Haryana, India
renish1419@gmail.com
[2] DRDO, ASL, Hyderabad, Telangana, India

Abstract. Drone detection refers to the process of identifying the presence of unmanned aerial vehicles (UAVs) or drones within a specific airspace. This technology has become increasingly important in recent years due to the growing popularity and use of drones for both civilian and military purposes. With the increasing usage of drones, there is a growing concern over the potential risks they pose, such as privacy invasion, malicious activities, and collisions with other aircraft. It is a critical security measure to prevent unauthorized drone activities like espionage, smuggling, and terrorism. Drone detection technology employs a variety of methods, including radar, acoustic sensors, and video cameras. These systems are integrated with software algorithms to accurately detect and track drones in real-time. This paper primarily focuses on real-time drone detection using deep learning methods to detect real-time UAVs. For the anti-drone system, we are using the YOLOv5 algorithm. Our experiment has shown that the YOLOv5 model produces better accuracy and maintains high detection speed.

Keywords: Deep Learning · Object detection · Drone · YOLO (You Only Look Once) · SSD (Single Shot Detector)

1 Introduction

Drones play an important role in the development of remote sensing and intelligent surveillance. Recently, many advancements were made in drone technology. Drones are becoming increasingly popular, and they are being used for a variety of purposes, including video capture, product delivery, monitoring, search and rescue, and so on. UAV's may be used in a hostile way, due to which drones have been a threat to general security and confidentiality [1]. Few instances of drone adversity are, in Canada when the flight was landing it ran into UAV, however there was only small damage and landed safely. There have been multiple incidents, like a drone crash in the White House [2] and a horrific near collision of an airliner and a drone near LAX airport [3]. Anti-drone-based object detection still faces challenges in actual applications.

B. Raj et al. (Eds.): ICETE 2023, AER 223, pp. 905–918, 2023.
https://doi.org/10.2991/978-94-6463-252-1_91

Drone detection using deep learning is a recent development in the field of unmanned aerial vehicle (UAV) surveillance. Deep learning is a type of machine learning that uses artificial neural networks to model and analyze large amounts of data. Additionally, deep learning algorithms can be used to detect drones in real-time and can be integrated with other surveillance systems to provide a comprehensive view of the airspace. As the use of drones continues to grow, the development of deep learning-based drone detection systems will play a significant role in ensuring the safety and security of people, property, and critical infrastructure.

The development of drone detection systems based on YOLOv5 is a remarkable step forward in the field of UAV surveillance. The requirement for reliable and effective drone detection systems will become more crucial as the usage of drones increases. YOLOv5 offers a promising answer to this challenge, enabling real-time, accurate, and dependable drone detection. Deep learning for drone identification often entails the development of a convolutional neural network (CNN), a form of deep learning algorithm intended exclusively for picture processing. CNN is trained on a large collection of drone photos, which allows it to understand the distinguishing characteristics of drones and separate them from other objects in the sky. The deep learning algorithm then uses this information to recognise drones in photographs or noises, even if they are partially covered or taken from various perspectives.

2 Related Work

Drone detection can be done using different methods, such as radar, acoustic, deep learning, etc. Improved deep learning models like Fast RCNN (Region-Based Convolutional Neural Network) can also be used for performing detection experiments [4]. There are a few stages where the experiment uses feature maps as input and output as object proposals. Then these proposals are considered by Fast RCNN in later stages, and thus object detection is accomplished. In [5], the authors explored the advantage of using SSD for object detection. This particular model has shown results with 99% accuracy and a considerable confidence level. Although this model showed better accuracy, the detection speed is still comparatively less when compared to other models. Hence, more research should be conducted on object detection models.

There are a few experiments that use cameras, specifically PTZ (pan tilt zoom) cameras, to detect objects in real time. After performing comparative analysis based on speed and accuracy using various models, it was found that the Faster RCNN Inception ResNet showed better results in terms of the F1 score [6]. In terms of detection speed, MobileNet and SSD are the fastest. When considering speed and accuracy tradeoffs, YOLOv2 is the best among the selected models.

Different approaches can be used to detect the drone or any object [7]. The operation was performed using the MobileNet V2 CNN model with masked images. Here, the backgrounds of objects are subtracted. This problem was solved by dividing it into two different stages. One is detection, and the other is classification. During this experiment, it was noticed that the detection speed is mainly dependent on the rate of background change. So, they fed more bounding boxes into the classifier, which increased the detection speed.

Incorporating machine learning techniques [8] with visible, thermal, and audio sensors exhibited promising results for real-time drone detection. The dataset contains different audio recordings of helicopters, drones, and background noises. To increase the focus of the moving object, a fish-eye lens camera was used. There is similarity between the F1 scores of visible video detectors and infrared detectors, in their experiment, an audio classifier achieved a high F1 score. Overcoming these drawbacks, the selected YOLO models were considered for future work. Thus, using YOLOv3 will provide more efficient results in detecting small objects such as birds and drones.

To tackle the difficulty of detecting small objects, a single shot detector YOLO model is proposed. In order to deal with the unavailability of specific data, an algorithm was created to extract the dataset and combine it with background-subtracted images. This approach works well for predicting the bounding boxes correctly. There are a few defects in the model due to birds being closer to the drone, so it encloses the bounding box containing both the bird and the drone [9].

Developing a reliable drone monitoring system has its own challenges [10]. One of them is making use of the limited drone images available to train the dataset. Augmentation techniques were used in this paper to manage this issue in an effective manner. To track a small object in a clustered environment, the authors used residue information from the images and then trained CNN (Convolutional neural network) based tracking model. MDNet (Multi-Domain Convolutional Neural Networks) object tracker is also used to reduce the complexity when testing the model in an online environment. An integrated system that performs detection and tracking was implemented, and this model outperformed the individual models that were used for tracking and detection separately. This is due to its ability to re-initialize the tracking bounding box when it loses the object.

3 YOLO (You Only Look Once)

3.1 Model Architecture

YOLOv5, or "You Only Look Once" is a real-time object detection algorithm that uses deep convolutional neural networks (CNNs) to detect and classify objects in images and video streams. The algorithm is designed to be fast, accurate, and versatile, making it an ideal choice for drone detection. This is the fifth version of Yolo, and it uses the Pytorch framework. There are various models of one-stage detectors other than YOLO, namely refineNet, SSD, and MobileNet. YOLO is better in a few cases, as we discussed in the previous section. Fast R-CNN has an inefficiency problem because the area generation module is performed in a specific module independent of CNN.

Figure 1 shows the architecture of yolov5. The basic principle of YOLOv5 is to divide an input image into a grid of cells and then use CNNs to predict the presence and location of objects within each cell. The algorithm uses anchor boxes and anchor points to improve its accuracy in detecting objects, especially in complex and cluttered environments. Anchor boxes are pre-defined boxes of different shapes and sizes that are used to approximate the location and size of objects in an image. Anchor points are used to refine the locations of objects by allowing the algorithm to adjust the position and size of anchor boxes to match the objects in an image.

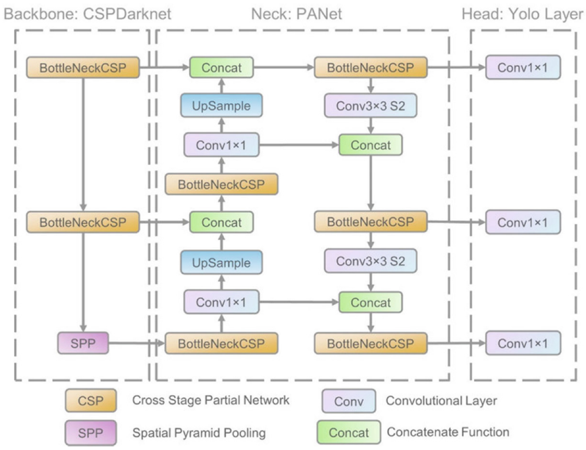

Fig. 1. Architecture of YOLOv5 [11]

To use YOLOv5 for drone detection, the algorithm must be trained on a large dataset of images that contain drones and other objects. During training, the model learns to identify the unique features of drones and to distinguish them from other objects in the scene. After training, the model can be used to analyse video streams from cameras or other sensors to detect drones in real-time. Yolov5 resembles Yolov4 with very few changes, such as the fact that Yolov4 uses the Darknet framework whereas Yolov5 is based on the PyTorch framework. And YOLOv4 uses a.cfg file for configuration, whereas YOLOv5 uses a.yaml file for configuration. The YOLOv5 uses a single neural network to process the entire picture.

YOLOv5 has 3 components: Backbone, Neck and Head. These components work together to analyze an input image or video stream and generate object detections in real-time. Figure 2 shows the yolo, which is the single stage detector and its architecture.

Backbone

This layer is responsible for receiving the input image or video stream. The image or video is then processed by the backbone network, which is responsible for extracting high-level features from the image. The backbone network is typically a convolutional neural network (CNN) that is trained on a large dataset of images and is designed to identify important features in an image. And this helps to reduce the spatial resolution

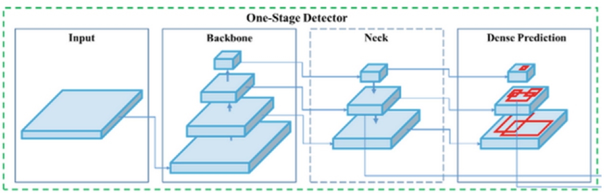

Fig. 2. Single stage detector architecture

of the image and increase its feature resolution. It uses a cross stage partial network strategy (CSP) to extract features and pyramids.

Neck

The neck network is responsible for refining the features extracted by the backbone network and preparing them for the prediction layer. It uses the Path Aggregation Network (PANet) to extract feature pyramids (same image at different scales to detect objects). It is used to improve information flow and to help in the proper localization of pixels in the task of mask prediction. Here, Spatial Pyramid Pooling (SPP) is used to improve the speed of the network.

Head

This layer is also known as the prediction layer and is also responsible for making the final object detections. This layer uses anchor boxes and anchor points to predict the location, size, and category of objects in an image and render the final output. Anchor boxes are pre-defined boxes of different shapes and sizes that are used to approximate the location and size of objects in an image. Anchor points are used to refine the locations of objects by allowing the algorithm to adjust the position and size of anchor boxes to better match the objects in an image.

4 Dataset

4.1 Data Collection

Data is crucial to all models which require training, Insufficient amount of data may cause underfitting of the data. So, we extracted 6,820 images from various sources which were unlabeled. The images were collected from various sources such as Google, roboflow, instagram, etc. And drones images we collected are from open source material. There are various images of drones present in the dataset and the dimensions of these drones differ from each other. All drones are multi-rotored and the images are of different sizes.

4.2 Data Labeling

We used a labeling image tool to label the images, i.e.; LabelImg. Firstly after downloading this software we open an image which has a drone, so we create a Rect box by keeping it in yolo format and it will be saved in a text file. The format of LabelImg is, "**<object class-ID> <x_center> <y_center> <width> <height>**". The main use of this label image tool is to assign a bounding box and use it for training. We divided the dataset into train and test data.

4.3 Training of the Data

We used the yolov5 model for the training and the reasons are explained further in the paper. We trained the enhanced model which was said by the original authors of yolov5 such as learning rate = 0.05 weight_decay = 0.0001, momentum = 0.97 with batch size of 8 and epochs of 100. The whole dataset is divided into training and validation folders with the proportion of 80 and 20 respectively.

5 Experimental Setup

The whole experiment we conducted was on a Linux PC in google Colab using its GPU (Tesla). The system was running on Debian OS. The model yolov5 had trained for about 782 iterations. If the value of IOU is greater than 0.5 then prediction is considered as true positive.

5.1 Working of Yolov5

In our case, the input is the images of drones. The model divides the drone image into a size-by-size grid. If the centre of a drone falls into a grid cell, then that grid cell detects the possible outcome, which in our case is drones. Each grid predicts the bounding boxes and confidence scores. These confidence scores tell us about the object (drone) present in the box and the accuracy of the model that predicted the box. The Yolo algorithm predicts multiple boxes per grid cell. At the time of training YOLO takes only one bounding box predictor for an object. It assigns one predictor to predict an object based on the highest IOU with the ground truth. The IOU concept is discussed in the evaluation metrics section.

The yolov5 detection flowchart is shown in Fig. 3.

YOLO uses the NMS (non-maximum suppression) method. NMS is used to improve the accuracy and efficiency of drone detection. Non-maximum suppression (NMS) is a fundamental approach in the YOLO models. NMS is a post-processing procedure

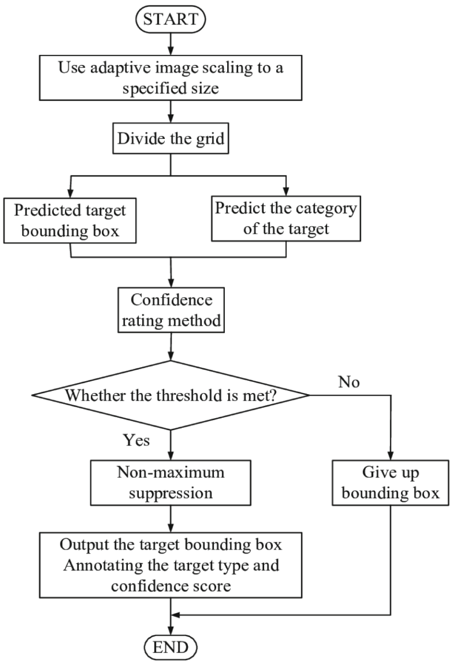

Figure 3. [10] Flowchart of Detection of an object using yolo

used to increase object detection accuracy and efficiency. Multiple bounding boxes are frequently created for a single item in an image during object detection. These bounding boxes may overlap or be in various places, but they all represent the same item. NMS is used to find and delete redundant or inaccurate bounding boxes from images, resulting in a single bounding box for each item in the picture. The Albumentations are automatically applied by the model itself. We Hypertuned the model with following values for albumentations Blur (p = 0.01, blur_limit = (3, 7)), MedianBlur (p = 0.01, blur_limit = (3, 7)), ToGray (p = 0.01), CLAHE (p = 0.01, clip_limit = (1, 4.0), tile_grid_size = (8, 8)).

6 Evaluation Metrics

There are few concepts on YOLOv5 calculation like IOU, ROC curves, F1 scores, Precision and Recall.

6.1 Intersection of Union (IOU)

Intersection Over Union is measured to evaluate the performance of the model (Accuracy), It evaluates the overlap of Ground Truth and Prediction region (Fig. 4).

The higher the IOU, the better the model. The IOU also helps to remove duplicate bounding boxes for the same object. It ranges the values from 0 to 1. If IOU is 0 means there is no overlap and whereas if IOU is 1 then it is perfect overlap. With the help of IOU threshold we can decide whether the prediction is True Positive, True Negative, False Positive and False Negative.

For example, If the prediction is 0.7 and then If we have the threshold of 0.98 then detection becomes False Positive.

1. **True Positive (TP):** The model successfully predicts the positive class.
2. **True Negative (TN):** The model successfully predicts the Negative class.
3. **False Positive (FP):** The model predicts the positive class incorrectly.
4. **False Negative (FN):**The model predicts the negative class incorrectly.

$$\mathbf{IOU} = \frac{\boldsymbol{TP}}{(\boldsymbol{TP} + \boldsymbol{FP} + \boldsymbol{FN})}$$

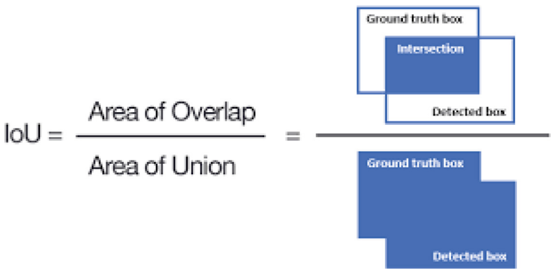

Fig. 4. IOU (Intersection over Union)

6.2 ROC Curve

The adverb of ROC is Receiver Operator Characteristic. The ROC curve is a graphical representation between True positive rate and False positive rate. It generally provides the performance of a classification model at all classification thresholds.

$$\textbf{True positive rate}\text{: TPR} = \frac{TP}{TP + FN}$$

$$\textbf{False positive rate}\text{: FPR} = \frac{FP}{FP + TN}$$

$$\textbf{Accuracy} = \frac{\textbf{\textit{True Positives}} + \textbf{\textit{True Negatives}}}{\textbf{\textit{Total samples}}}$$

6.3 Precision

Basically, it tells about how the model is detected for a particular object. It helps when the cost of false positives are high. If there is a model to detect a fruit like mango, so if it detects mango correctly then it comes under precision.

$$\textbf{Precision} = \frac{\textbf{\textit{True Positives}}}{\textbf{\textit{True Positives}} + \textbf{\textit{False Positives}}}$$

6.4 Recall

It describes how well the model did for actual observations of a particular class, like how the model behaved for the whole class. In the above example if the model detects the whole class correctly as mango then its recall. Recall is better measure than precision

$$\textbf{Recall} = \frac{\textbf{\textit{True Positives}}}{\textbf{\textit{True Positives}} + \textbf{\textit{False Negatives}}}$$

6.5 F1 Score

$$\textbf{F1 Score} = 2 * \frac{\textbf{\textit{Precision}} * \textbf{\textit{Recall}}}{\textbf{\textit{Precision}} + \textbf{\textit{Recall}}}$$

F1 is an overall accuracy measure that combines precision and recall. Good F1 score indicates that you have few false positives and false negatives, indicating that you are accurately recognising serious threats and are not bothered by false alarms. When an F1 score is 1, the method is regarded as perfect, but when it is 0, the model is considered a complete failure.

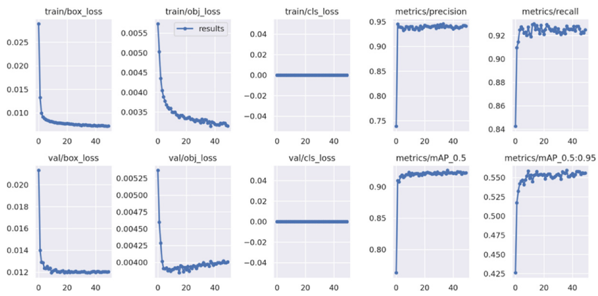

Fig. 5. All the metrics of training

7 Results

To know the performance of the models, there are a few metrics to evaluate the model by, such as mAP, precision, recall, and F1-score. Since the model was split into train and test, the evaluation metrics are calculated using a test dataset. There are plots of the model that show the performance of our model. Figure 5 shows the synopsis for the training. The loss curve is downward, and this indicates that during training the losses were minimized. In the metrics graph, the mAP graph is initially upward, then it continues to increase and has a mAP value of 0.928. Throughout the training, the model improved over the iterations, and then it decreased at the end. Figure 6 depicts the precision recall curve, it is right, which means the model's rate of classification is good, when using the model. The tradeoff between accuracy and recall for various thresholds is depicted by the precision-recall curve. High accuracy is correlated with a low false positive rate, while high recall is correlated with a low false negative rate. A high area under the curve denotes both high recall and high precision. The graph is oriented to the right, indicating that it is effective at categorization (drone). Figure 7 shows the F1 curve depicted in the graph, and the maximum F1 value is 0.93, which corresponds to the confidence value that maximises recall up-to 0.95. Typically, it is preferable to have a higher F1 score and confidence value.

Figure 8 are results of the model which predicted on the test dataset (val), the model predicted every drone with good accuracy.

Figures 9 and 10 are the results we got in Real time drone detection.

8 Comparison of Models

Comparison between object detector models is not an easy task as we have to make trade-offs between speed, accuracy. So, knowing what your application needs more between speed and accuracy is the real question. Though accuracy is the main metric used when deciding whether a model performs better on the dataset, we cannot ignore the importance of speed when it comes to detection. Object detection models such as SSD performs at a much higher speed when compared to R-CNN but it performs worse when detecting small sized objects. When coming to Faster RCNN and YOLO, both use CNN at their core but their frameworks differ from each other. To perform detection, Faster RCNN

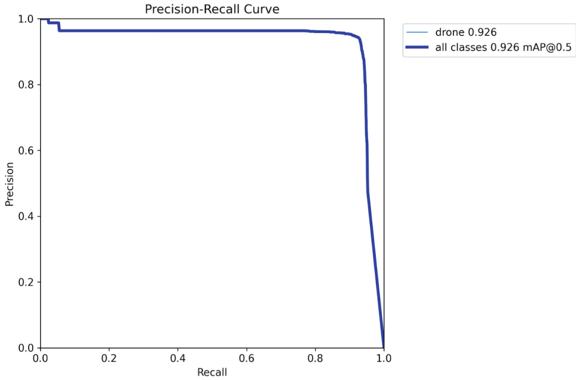

Fig. 6. Precision-Recall Curve

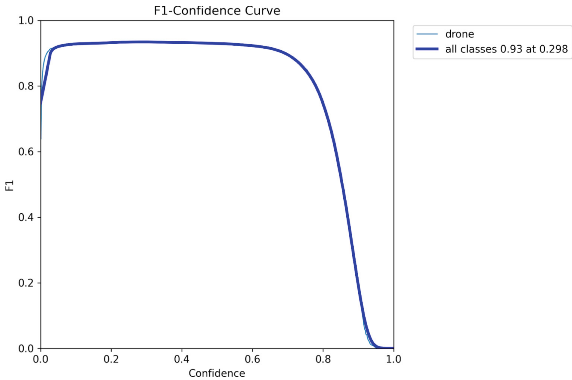

Fig. 7. F1 score

Fig. 8. Results of val batch prediction

uses ROI pooling which is much faster than its predecessors and the other hand YOLO performs detection and classification at the same time which makes it a more efficient

Fig. 9. Real-time prediction of drone in low light

Fig. 10. Real-time prediction of drone during sunlight

model for detecting a drone from an image or video. In Table 1 we compare the results of the models.

Many authors performed experiments on drone detection problems using multiple models as discussed above [12]. Drone detection performed using MobileNetSSD with 1800 images produced an accuracy of 0.788 and recall of 0.524. In [13], authors conducted training with a dataset from the SafeShore project using Faster R-CNN (resnet101) which gave an accuracy of 0.934. In another paper Mask R-CNN with MobileNet Architecture was the preferred model for identification of drones. The training was done on 1000 images of drones and the results obtained were an accuracy of 0.812 and recall of 0.642 [14]. An accuracy of 0.743, recall of 0.68 and F1-score of 0.79 was acquired when the same study was done using YOLOv4 [15] with 2395 images (479 birds, 1996 drones). The comparison is not done to rank the models based on their

Table 1. Comparison of models

S.no	**Model**	**Training Data (Images)**	**Accuracy**
1	**SSD**	**1800**	**78.8%**
2	**Faster R-CNN**	**2664**	**66.9%**
3	**Mask R-CNN**	**1000**	**81.2%**
4	**YOLOv4**	**2395**	**74.36%**
5	**YOLOv5**	**>8000**	**92%**

performance rather it was compared to give us an overview of how different models performed on the same problem. Our model YOLOv5 achieved a desirable accuracy, recall of 0.926, 0.95 respectively.

9 Conclusion

In this study, we implemented a drone detection algorithm using a deep learning model called YOLOv5. Drones can be useful in various scenarios, but they also have cons that are difficult to overcome. For this reason, we made use of deep learning techniques to develop this model. Using our algorithm, we can detect drones with normal cameras as well. We fine-tuned the YOLOv5 model's hyperparameters and a few other parameters for better accuracy. This model also has the ability to capture small sized drones. Our model is capable of detecting drones in real time with a high detection speed. This feature of the model can be useful in many real life scenarios. Even though our selected dataset is large, we showed better results when compared to previous models that used a smaller amount of data. In addition to its real-time performance, YOLOv5 also provides high accuracy in detecting drones. The algorithm uses anchor boxes and anchor points to accurately locate the objects in an image, even in complex and cluttered environments. This allows it to detect drones accurately and reliably, even in challenging conditions such as low light or adverse weather. There were certain limitations to our model that occurred due to insufficient computation power and inconsistencies in the dataset. There were some difficulties during the training of our model which impacted our model performance for instance, there were few corrupted files that we had to remove in order to clean the dataset. Since the data was unlabelled we had to manually label our data but during the process of training it resulted in having multiple classes for a single object. For this reason we had to re verify our data and correct mislabelled text files. As for the future work, we will explore different options to configure our existing model and compare it with other models. By performing an analysis, we can understand where our model falls short and thus improve its accuracy. The model is still in the training process, and in the future we will update the details of the model.

Acknowledgements. We thank Sri. Sanjeev Kumar (Technology Director), Smt. Sowmya Mitra (Scientist) and IV&V team members for their motivation, support and guidance during the development, training and testing of the model.

References

1. Pham, S. Drone Hits Passenger Plane in Canada. CNN. Available online: https://money.cnn.com/2017/10/16/technology/drone-passenger-plane-canada/index.html (accessed on 16 October 2017).
2. Jansen, B. Drone crash at white house reveals security risks. USA Today 26 January 2015. Available online: https://www.usatoday.com/story/news/2015/01/26/drone-crash-secret-service-faa/22352857/.

3. Guardian, T. Gatwick Drone Disruption Cost Airport Just £1.4 m. The Guardian. Available online: https://www.theguardian.com/uk-news/live/2018/dec/21/gatwick-drone-airport-limited-flights-live (accessed on 21 December 2018).
4. R. Girshick, "Fast r-CNN", *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.
5. G. Chandan, A. Jain, H. Jain and Mohana, "Real Time Object Detection and Tracking Using Deep Learning and OpenCV," *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2018, pp. 1305–1308, doi: https://doi.org/10.1109/ICIRCA.2018.8597276.
6. J. Park, D.H. Kim, Y. S. Shin and S. -h. Lee, "A comparison of convolutional object detectors for real-time drone tracking using a PTZ camera," *2017 17th International Conference on Control, Automation and Systems (ICCAS)*, Jeju, Korea (South), 2017, pp. 696–699, doi: https://doi.org/10.23919/ICCAS.2017.82043.
7. Seidaliyeva, U.; Akhmetov, D.; Ilipbayeva, L.; Matson, E.T. Real-Time and Accurate Drone Detection in a Video with a Static Background. *Sensors* 2020, *20*, 3856. doi: https://doi.org/10.3390/s20143856.
8. F. Svanström, C. Englund and F. Alonso-Fernandez, "Real-Time Drone Detection and Tracking with Visible, Thermal and Acoustic Sensors," *2020 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy, 2021, pp. 7265–7272, doi: https://doi.org/10.1109/ICPR48806.2021.9413241.
9. C. Aker and S. Kalkan, "Using deep networks for drone detection," *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Lecce, Italy, 2017, pp. 1–6, doi: https://doi.org/10.1109/AVSS.2017.8078539.
10. Y. Chen, P. Aggarwal, J. Choi and C. -C. J. Kuo, "A deep learning approach to drone monitoring," *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Kuala Lumpur, Malaysia, 2017, pp. 686–691, doi: https://doi.org/10.1109/APSIPA.2017.8282120.
11. A Forest Fire Detection System Based on Ensemble Learning - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/The-network-architecture-of-Yolov5-It-consists-of-three-parts-1-Backbone-CSPDarknet_fig1_349299852 (accessed on 14 February 2023).
12. A. Pansare et al., "Drone Detection using YOLO and SSD A Comparative Study," *2022 International Conference on Signal and Information Processing (IConSIP)*, Pune, India, 2022, pp. 1–6, doi: https://doi.org/10.1109/ICoNSIP49665.2022.10007489.
13. A. J. Garcia, J. Min Lee and D. S. Kim, "Anti-Drone System: A Visual-based Drone Detection using Neural Networks," *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, Korea (South), 2020, pp. 559–561, doi: https://doi.org/10.1109/ICTC49870.2020.9289397.
14. A. S. Mubarak, M. Vubangsi, F. Al-Turjman, Z. S. Ameen, A. S. Mahfudh and S. Alturjman, "Computer Vision Based Drone Detection Using Mask R-CNN," *2022 International Conference on Artificial Intelligence in Everything (AIE)*, Lefkosa, Cyprus, 2022, pp. 540–543, doi: https://doi.org/10.1109/AIE57029.2022.00108.
15. S. Singha and B. Aydin, "Automated Drone Detection Using YOLOv4," *Drones*, vol. 5, no. 3, p. 95, September 2021, doi: https://doi.org/10.3390/drones5030095.