Proceedings of the 7th International Conference on Trends in Electronics and Informatics (ICOEI 2023)
IEEE Xplore Part Number: CFP23J32-ART; ISBN: 979-8-3503-9728-4

# Drone Detection and Classification using Computer Vision

Ruchita Valaboju
*Dept. of CSE*
*Gokaraju Rangaraju Institute of Engineering and Technology*
Hyderabad, India
ruchita.valaboju@gmail.com

Vaishnavi
*Dept. of CSE*
*Gokaraju Rangaraju Institute of Engineering and Technology*
Hyderabad, India
vaishnavirao66@gmail.com

Chitiprolu Harshitha
*Dept. of CSE*
*Gokaraju Rangaraju Institute of Engineering and Technology*
Hyderabad, India
hchitiprolu@gmail.com

Alekhya Reddy Kallam
*Dept. of CSE*
*Gokaraju Rangaraju Institute of Engineering and Technology*
Hyderabad, India
reddyalekhya230@gmail.com

B. Sankara Babu
*Dept. of CSE*
*Gokaraju Rangaraju Institute of Engineering and Technology*
Hyderabad, India
bsankarababu81@gmail.com

2023 7th International Conference on Trends in Electronics and Informatics (ICOEI) | 979-8-3503-9728-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICOEI56765.2023.10125737

***Abstract*—Drones, also known as Unmanned Aerial Vehicles (UAVs) are based on the principle of rotor torque pushing the air down which results in the upward lift of the drone. UAVs are used in tasks such as rescue operations and item delivery in remote areas, surveillance, agriculture, wildlife conservation, outer space and photography. Due to their low cost and high efficiency, it is used by diverse groups for both better and worse causes. The cases of malicious uses of military drones and spy drones employed are on a rise. The malicious activities deployed by the military drones may include air strikes at enemy military bases, army troops and in some cases end up causing the death of civilians in proximity. Drones that are used for espionage can retrieve valuable information regarding the different strategies of the military, can track the location of the army personnel and spy upon unsuspecting civilians. These drones are eliminated after being detected by the persons involved but sometimes in order to reduce risks, harmless delivery drones are also discarded. In order to aid against the malicious drone activity while making sure that unnecessary panic over the delivery drones and material-loss is not caused, a real-time computer vision system is proposed that can identify the drone in the given region of interest, give its relative location and classify the drone. The Convolutional Neural Network (CNN) architecture, You Only Look Once Algorithm (YOLOV5), is used to classify the drone into one of the categories: Army, Surveillance and Delivery drones.**

***Keywords—Drone Detection, Computer Vision, You Only Look Once, Army Drone, Surveillance Drone, Delivery Drone, Mean Average Precision***

## I. INTRODUCTION

Drones have been primarily used for rescue operations by making use of their surveillance and sensing features. They have been helpful in not only delivering necessary items during emergencies like natural disasters but also sense the origin of the disaster and take the required measures to mitigate the damage to property and population. And then came the commercialization of drones in photography that could capture the stunning and rare views of sceneries, small-item delivery, thorough fertilization and pesticide application in farming and geological inspection operations, to name a few[1][2]. This is owing to the low costs associated with the procurement and the maintenance of drones, their time-saving capabilities and their proven efficiency in the said operations.

The defense operations using drones have been implemented for a wide range of applications ranging from surveillance of the country borders, item-delivery to combat against the enemies and missile-firing[2]. They have been inexpensive in warfare activities because they can eliminate the risk to the soldiers' lives who initially used to operate the military aircrafts before the military UAVs came along. On the other hand, unregulated drone attacks such as air strikes and targeted slaughter have grown at an alarming rate thanks to the ever-developing drone technology[3]. This is leading to compromised soldier and civilian safety. Surveillance drones, once that have been used for rescue operations and observing key details of the region of interest for educational purposes, are now being used to spy upon both unsuspecting civilians and military grounds and personnel by criminals to extract valuable information, causing data and privacy breaches[2][4]. On the other hand, delivery drones that are completely harmless are being eliminated in order to prevent any malicious activity from occurring. This causes heavy resource losses and unnecessary panic to the citizens or the personnel involved[5]. So, there is a need to take measures to identify the malicious activity of drones while also making sure that there are no false alarms caused due to the usage of delivery drones.

## II. LITERATRUE STUDY

The monitoring of drones in the region of interest is generally done using Closed-Circuit Television (CCTV) surveillance by the personnel involved. The drones can be detected by the characteristic sound that they make, using sound detection sensors which are useful in environments or conditions where the visibility is less[6][7]. Another way of detection is by processing the radio frequencies emitted by the drone to communicate with its controller or by using radar which emits electromagnetic waves that allows to process the echoes caused by the reflection of the waves by the drone[8][9]. The computer vision approaches that exist for drone detection include the usage of Convolutional Neural Networks(CNNs) for detection of the drone alone or both detection and localization of the drone. Localization refers to the process of determining the coordinates of the most accurate bounding box that surrounds the object in the image. Faster Regional Convolutional Neural Network (Faster R-CNN) algorithm combined with various predefined CNN architectures like ResNet, have been deployed by [10] for the detection of drones. Faster R-CNNs perform predictions region-wise and are hence faster. The authors have used tuned the models to achieve improved mean Average Precision(mAP), which is an important performance measure of image detection models. Detection of moving drones in a static background in combination with background subtraction method, using an established CNN architecture MobileNetv2 network, has been deployed by [11]. MobileNetv2 uses low computational power and can work on low-power devices like mobiles. Using the Haar Cascade algorithm to detect the drone and then using another custom CNN to classify the model of the drone is done by [12]. Haar cascade is beneficial in detecting the objects regardless of their size and works on the principle of training with positive (contains object) and negative images(does not contain object). Another approach is using a one-pass detection algorithm which can both detect and classify the drone using the same algorithm instead of two different algorithms, which is the case with YOLO. Detection and Classification of drones based on the number of rotors of the drone is experimented by [13], using YOLOV3. Often birds can be misidentified as drones and vice versa and hence can affect the detector's performance. This has been addressed by [14], where the authors have trained YOLOV4 for two objects drones and birds, hence solving the problem mentioned.

Although the radar, radio frequency and acoustic approaches are neck-to-neck with the computer vision approaches, the computer vision approaches involve working with free and open-source databases. They also deal working with optical sensors which are cost-effective, easy to install and work with. They work well in different weather conditions as compared to radar or radio frequency signals which can be degraded during the transmission and interrupt the process of drone detection. Current computer vision approaches are using different algorithms for different parts of the detection process, hence making their training computationally expensive and time-consuming. These approaches can also slow down the detection speed because of the usage of heavy architectures to achieve high accuracy at the cost of being unable to make real-time detections.

## III. PROPOSED METHOD

A computer vision based safety system is proposed which works on the principle of You Only Look Once Object Detection(YOLO)[15]. The system is capable of detection, localization and classification of the drone object when given an image or a video. After detection and localization of the drones, they are classified into three categories: Army, Delivery and Surveillance drones. It performs the three different steps: detection, localization and classification at the same time and using the same algorithm as opposed to other procedures where detection and classification are done by different algorithms. It uses lower computational power for both training and prediction and has faster detection speed. For example, algorithm A is used for detection and algorithm B is used for classification purpose, creating an algorithmic pipeline. A and B require different types of datasets, can take much longer to train and the pipeline might not work effectively if the dataset is not substantially large. Another reason for choosing YOLO is that it can process the image faster than the other established CNNs because it does not perform repeated bounding box predictions for the same regions of an image, can learn the general representation of the objects quickly and can provide real-time detection of objects[15]. The YOLOV5[16] model is chosen for the functioning of the system. The V5 version is chosen because it is a faster and lighter version than the previous YOLO models since they are based on heavy architectures like Darknet.

### *A. Data Collection and Preprocessing*

The three classes Army, Delivery and Surveillance drones are defined below as per the scope of the proposed work:

i. Army Drone: It involves images of drones that are used by the military mainly for firing missiles. The US military drones that are included in the dataset are MQ-1C Gray Eagle, RQ-4 Global Hawk, MQ-4C Triton and MQ-9 Reaper. These are the most-used drones and since they share similar shape characteristics, they can serve as a general representation of an army drone.
ii. Delivery Drone: The deliveries done by a drone may involve picking up an item from the seller at a predefined spot and then delivering it to a buyer at another predefined point, by following a well-established route between the seller-point and the buyer-point. The item to be delivered might be attached to a hook that is lowered by the drone while it is hovering at a height. After the item is attached, the winch that lowered the hook, will tug at the package/item to reach up to the drone. Such delivery drones' representation is included in the dataset by collecting images that had a rectangular item/box, suspended from the hook of the drone.
iii. Surveillance Drone: For this experiment, images with Quadcopter drones (drones with four rotors) equipped with a camera have been collected so that they can represent a surveillance drone.

The above mentioned classes' images are collected from the web. 300 training images per each class are collected, in total 900 training images. Similarly, 30 test images are collected per each class, in total 90 test images. The preprocessing of the images is done with the help of LabelImg software[17]. It allows us to draw the bounding boxes and enter the corresponding labels. It gives the coordinates of the box drawn as an output text file that get saved in the same directory of the corresponding image. The output is in the form of [class, x, y, x+w, y+h], where:

i. Class: An integer denoting the class of the object where
   a. 0 represents Army Drone
   b. 1 represents Delivery Drone
   c. 2 represents Surveillance Drone

ii. x and y denote the top left most coordinates of the bounding box.

iii. x+w and y+h denote the bottom right most coordinates of the bounding box, where w and h are the width and the height of the box, respectively.

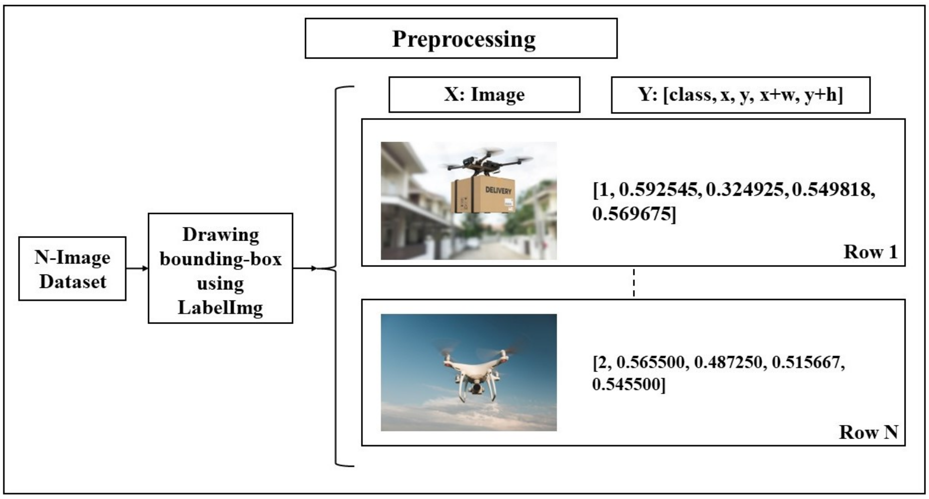


Fig. 1. Image Preprocessing and Dataset Preparation (Drone Image Sources: [20] and [21])

The image preprocessing and the final dataset (X,Y) are represented in Fig. 1. The data corresponding to multiple drones that are present in a single image are stored in the same output text file where each line of (class, x, y, x+w, y+h) corresponds to a single drone's information. This concludes the preprocessing of the images.

*B. Training, Prediction and Metrics*

YOLO is a one-pass detection algorithm, i.e., scans through the image only once, making it ideal for real-time detection. It divides the image into n*n grid cells and searches for the center of the object in each grid cell and when found, it calculates the coordinates of the object's bounding box. When more than one bounding box is predicted for a grid cell, it tries to see if they belong to different objects or the same object. To deal with this problem, the model uses Intersection over Union(IoU) of the boxes. IoU is defined as the intersection of the ground truth bounding box area and predicted bounding box area of a given object divided by the union of boxes' areas. When overlapping bounding boxes exist, the IoU is calculated pairwise and if it is greater than a threshold, the boxes belong to the same object and the box with lower confidence is discarded. This process is called Non-Maximum Suppression. Confidence is defined as the product of the probability of an object belonging to a class and the predicted box's IoU with the ground truth box. This way, it searches for the objects in each grid cell and then collectively outputs the found object/objects information of all the grid cells. The working of YOLO is shown in Fig. 2.

Detection of everyday objects by the YOLO algorithm is done using the pre-trained weights of the Common Objects in Context (COCO)[18]. The COCO list consists of 91 everyday objects like person, bicycle, car, etc. of which the drone object is not a part. Since there are no pre-trained weights for the drone object, YOLO Custom Object Detection method is adapted which involves training one's own weights. Custom Object Detection is also used when there is a need for one's own training weights for the objects that already have pre-trained COCO weights. This will require obtaining a dataset corresponding to the object and then drawing the bounding boxes around the objects and labeling them.

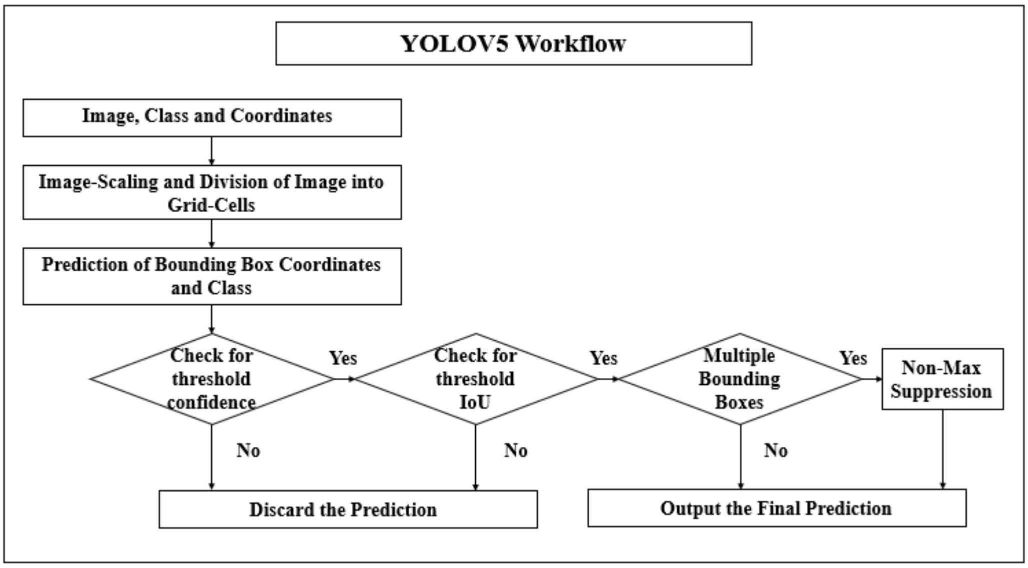


Fig. 2. YOLOV5 Working Process

The image is given as the input and an array consisting of the bounding box coordinates and the class of the object are given as the output for the training purpose. The cloned directory of YOLOV5 is used to access the YOLOV5s(YOLOV5 Small) architecture and the initial weights and the biases of the architecture. Train.py file of the cloned repository is used to run the train command. It includes parameters like:

i. The architecture YOLOV5s, the initial weights YOLOV5s.pt and training and validation images folder path.
ii. The image resolution that is used to resize the image for training(which is 640 for the case of YOLOV5s).
iii. The batch size (batch_size=2).
iv. The number of epochs (epochs=100).

For each training epoch, the model's performance metrics including Box loss, Object loss, Class loss, Precision, Recall and Mean Average Precision(mAP) are obtained. The model's best epoch's weights or the last epoch's weights can be chosen for performing predictions on new images or videos. The Weights and Biases (WandB)[19] module associated with the train command allows us to store the weights in the current experiment's directory. The WandB website that is accessed through an API call from train.py, helps us to log weights, retrieve and visualize the metrics of the corresponding experiment; following successful registration and login into the website. The process of training and weights generation is shown in Fig. 3.

The algorithm is trained on N=750 images and N=900 images to generate two different models. Let the two models be Model_750 and Model_900. Model_750 and Model_900 are validated with 75 images and 90 images, respectively. The comparison of the metrics of the models is shown in Table I. Every time the train command is run, a separate directory is created for that experiment both locally and on WandB website where a comprehensive understanding can be gained from metric variation graphs, confusion matrix and example predictions. The train command makes an Application Interface Programming Interface (API) call to the Weights and Biases website which takes the weights, epoch-wise metrics and losses as inputs and it outputs different visualizations for these metrics. One such visualization, Fig. 4, consists of the class-wise and overall precision versus recall graph, out of which the Average Precision(AP) of each class can be derived.

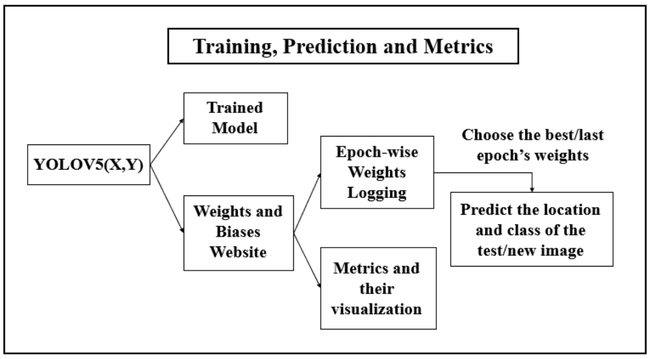


Fig. 3. Training and Prediction

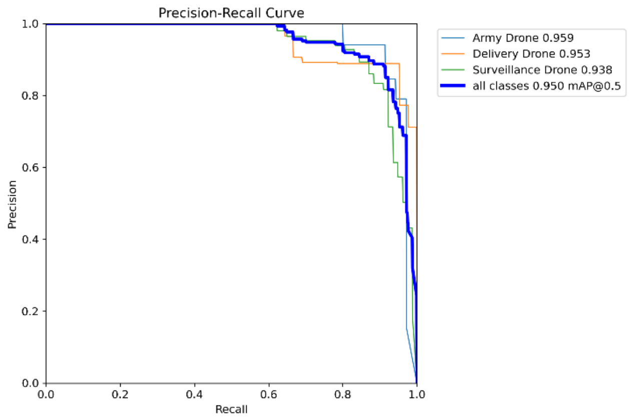


Fig. 4. Precision versus Recall Graph(Graph generated using [19])

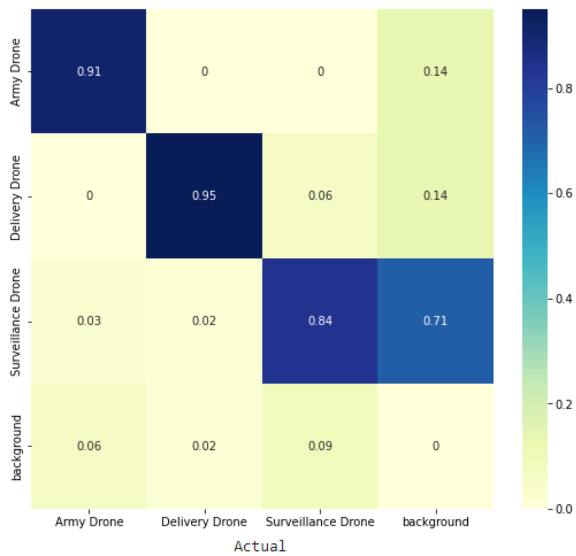


Fig. 5. Confusion Matrix of Model_900 (Matrix generated using [19])

An interesting metric, mean Average Precision(mAP), derived from the confusion matrix using IoU, is considered important while assessing the performance of an object localization model. Fig. 5 depicts the confusion matrix of Model_900. The confusion matrix elements: True positive, True Negative, False Positive and False Negative are calculated for bounding problems by comparing the predicted box's IoU with the ground truth and threshold IoU. For example, if predicted box's IoU with the ground truth is greater than the threshold IoU, the prediction is considered as a True Positive. mAP50 is the mean of APs of all the classes for IoU=0.5 and mAP50-95 is the mean of APs of all the classes over the IoU interval [0.50,0.95]. The class-wise and overall precision, recall, mAP50 and mAP50-95 of the Model_900 are shown in Table II.

## IV. RESULTS

From Table I, it can be seen that the highest accuracy of the proposed system, 68%, is achieved by Model_900. An increase in the dataset from 750 to 900 images led to an increase of 5% in mAP50-95 and 4% increase in accuracy, indicating that the model has improved well with a minimum increase in the dataset size. The confusion matrix of the Model_900, shown in Fig. 5, consists of a "background" row and column, these are the percentages of the drones that are falsely predicted as background information (False Negative) and background information falsely predicted as a drone (False Positive), respectively. It can be seen from Fig. 5 that the background information falsely being predicted as Surveillance Drone is high. This problem can be handled by increasing the dataset size or increasing the threshold confidence. When the training and validation loss curves' values decrease to a point of stability and there is a small but definite difference between the two points of stabilities, then the model is said to be well fit to the dataset. From Fig. 6, it can be understood that is the case with the Model_900's box- loss and object-loss curves but not quite the same with the class-loss curve which seems a bit noisy, indicating a slight underfitting in terms of the Model_900 generalizing with the classes. Fig. 6 also consists of the epoch-wise precision, recall and mean average precision. The comparison of the proposed system's (Model_900) metrics with existing methods' metrics can be seen from Table III. It can be seen that the proposed system is performing substantially well across a good number of metrics.

TABLE I. COMPARISON OF THE MODEL_750 AND MODEL_900 METRICS

| **Number of Images (N)** | **Metric** | | | | | |
|---|---|---|---|---|---|---|
| | ***Accuracy*** | ***Precision*** | ***Recall*** | ***F1-Score*** | ***mAP50*** | ***mAP50-95*** |
| N=750 | 0.641 | 0.868 | 0.865 | 0.866 | 0.913 | 0.617 |
| N=900 | 0.680 | 0.889 | 0.904 | 0.896 | 0.95 | 0.665 |

TABLE II. MODEL_900'S CLASS-WISE AND OVERALL METRICS

| **Class** | **Metric** | | | | |
|---|---|---|---|---|---|
| | ***Precision*** | ***Recall*** | ***F1-Score*** | ***mAP50*** | ***mAP50-95*** |
| All | 0.889 | 0.904 | 0.896 | 0.95 | 0.665 |
| Army | 0.934 | 0.914 | 0.924 | 0.959 | 0.66 |
| Delivery | 0.837 | 0.952 | 0.891 | 0.953 | 0.741 |
| Surveillance | 0.897 | 0.844 | 0.870 | 0.938 | 0.594 |

TABLE III. COMPARISON OF THE PROPOSED METHOD METRICS (MODEL_900) WITH THE EXISITING METHODS' METRICS

| **Approach** | **Metric** | | | |
|---|---|---|---|---|
| | ***Precision*** | ***Recall*** | ***F1-Score*** | ***mAP50*** |
| [10] | -- | -- | -- | 0.490 |
| [11] | 0.701 | 0.788 | 0.742 | -- |
| [13] | -- | -- | -- | 0.740 |
| [14] | 0.950 | 0.680 | 0.790 | -- |
| Proposed System | 0.889 | 0.904 | 0.896 | 0.950 |

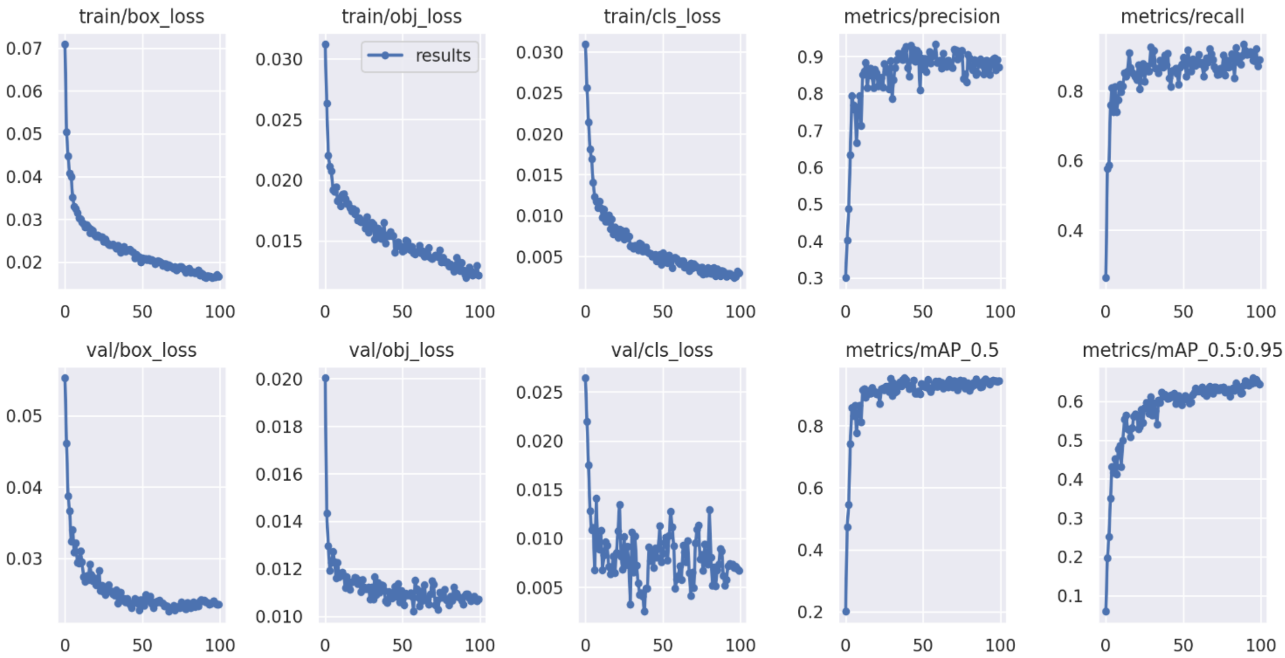


Fig. 6. Metrics' Variance per Epoch Visualisation (Graphs generated using [19])


Fusing layers...
Model summary: 157 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs
Coordinates of bounding box 1 [0.46875, 0.6498829126358032, 0.746874988079071, 0.17564402520656586]
tensor(0.49689)

Number of objects detected : 1
image 1/1 /content/general-atomics-gff009dd7c_640.jpg: 448x640 1 Army Drone, 288.6ms
Speed: 6.1ms pre-process, 288.6ms inference, 1.1ms NMS per image at shape (1, 3, 640, 640)
Results saved to **runs/detect/exp88**

Fig. 7. Prediction of Army Drone (Image source used for prediction: [22])


```
Fusing layers...
Model summary: 157 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs
Coordinates of bounding box 1 [0.7425000071525574, 0.7462499737739563,
0.4839999737739563, 0.2975000143051147 5] tensor(0.85159)
Coordinates of bounding box 2 [0.2709999978542328, 0.3372499942779541,
0.5230000019073486, 0.27250000834465027] tensor(0.89849)
Coordinates of bounding box 3 [0.7425000071525574, 0.33799999952316284,
0.5080000162124634, 0.25699999928474426] tensor(0.89978)
Coordinates of bounding box 4 [0.2667500078678131, 0.7494999766349792,
0.5005000233650208, 0.289000004529953] tensor(0.91040)

Number of objects detected : 4
image 1/1 /content/4drones.jpg: 640x640 4 Surveillance Drones, 418.7ms
Speed: 1.6ms pre-process, 418.7ms inference, 1.6ms NMS per image at shape (1, 3, 640, 640)
Results saved to runs/detect/exp86
```

Fig. 8. Prediction of Multiple Surveillance Drones (Image source used for prediction: [23]

## V. CONCLUSIONS AND FUTURE WORK

The trained model is successful in its purpose as seen from the prediction of new images, shown in Fig. 7 and Fig.8. Fig. 8 shows that the predicted drones are numbered so that the corresponding coordinates of the drones can be identified in an order. The newly predicted images are automatically stored and can be used for re-training in the future for better performance. The difficulty in re-training lies in the preprocessing which involves manual-labeling of large number of images. Storage of the predicted images and videos for future use will require large databases and it will be taxing to maintain these databases and might require constant scaling-up. If one opts to use public cloud services for easy scaling-up of resources and pay-as-you-go facility, security of the data cannot be ensured which is important for an application like this. Future goals of this system include improving the performance of the model by eliminating the risk of mis-identifying birds as surveillance drones and airplanes as army drones and incorporating more varieties of military drones into the army dataset and evolve the model to classify specific kinds of army drones as separate classes.

## REFERENCES

[1] M. Hassanalian, A. Abdelkefi, Classifications, applications, and design challenges of drones: A review, Progress in Aerospace Sciences, Volume 91, 2017, Pages 99-131, ISSN 0376-0421, https://doi.org/10.1016/j.paerosci.2017.04.003.

[2] Jean-Paul Yaacoub, Hassan Noura, Ola Salman, Ali Chehab, Security analysis of drones systems: Attacks, limitations, and recommendations, Internet of Things, Volume 11, 2020, 100218, ISSN 2542-6605, https://doi.org/10.1016/j.iot.2020.100218.

[3] Pledger, T. (2021). The role of drones in future terrorist attacks. *Association of the United States Army*.

[4] Thomasen, K. (2017). Beyond airspace safety: A feminist perspective on drone privacy regulation.

[5] Khan, R, Tausif, S, Javed Malik, A. Consumer acceptance of delivery drones in urban areas. *Int J Consum Stud*. 2019; 43: 87–101. https://doi.org/10.1111/ijcs.12487

[6] B. Taha and A. Shoufan, "Machine Learning-Based Drone Detection and Classification: State-of-the-Art in Research," in IEEE Access, vol. 7, pp. 138669-138682, 2019, doi: 10.1109/ACCESS.2019.2942944.

[7] M. Z. Anwar, Z. Kaleem and A. Jamalipour, "Machine Learning Inspired Sound-Based Amateur Drone Detection for Public Safety Applications," in IEEE Transactions on Vehicular Technology, vol. 68, no. 3, pp. 2526-2534, March 2019, doi: 10.1109/TVT.2019.2893615.

[8] Mohammad F. Al-Sa'd, Abdulla Al-Ali, Amr Mohamed, Tamer Khattab, Aiman Erbad, RF-based drone detection and identification

using deep learning approaches: An initiative towards a large open source drone database, Future Generation Computer Systems, Volume 100, 2019, Pages 86-97, ISSN 0167-739X, https://doi.org/10.1016/j.future.2019.05.007.

[9] Huan Lv *et al* 2021 *J. Phys.: Conf. Ser.* 1738 012044, doi: 10.1088/1742-6596/1738/1/012044.

[10] M. Nalamati, A. Kapoor, M. Saqib, N. Sharma and M. Blumenstein, "Drone Detection in Long-Range Surveillance Videos," 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2019, pp. 1-6, doi: 10.1109/AVSS.2019.8909830.

[11] Seidaliyeva, U., Akhmetov, D., Ilipbayeva, L., & Matson, E. T. (2020). Real-Time and Accurate Drone Detection in a Video with a Static Background. *Sensors*, *20*(14),3856. https://doi.org/10.3390/s20143856

[12] D. Lee, W. Gyu La and H. Kim, "Drone Detection and Identification System using Artificial Intelligence," 2018 International Conference on Information and Communication Technology Convergence (ICTC), 2018, pp. 1131-1133, doi: 10.1109/ICTC.2018.8539442.

[13] D. K. Behera and A. Bazil Raj, "Drone Detection and Classification using Deep Learning," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020, pp. 1012-1016, doi: 10.1109/ICICCS48265.2020.9121150.

[14] Singha, S., & Aydin, B. (2021). Automated Drone Detection Using YOLOv4. Drones, 5(3), 95. https://doi.org/10.3390/drones5030095

[15] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[16] Jocher, G. (2020). YOLOv5 by Ultralytics (Version 7.0) [Computer software]. https://doi.org/10.5281/zenodo.3908559

[17] Tzutalin (2015). LabelImg Free Software: MIT License .

[18] Lin, TY. *et al.* (2014). Microsoft COCO: Common Objects in Context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48

[19] L. Biewald, "Experiment Tracking with Weights and Biases," Weights & Biases. [Online]. Available: http://wandb.com/. [Accessed: 19 January 2023]. Software available from wandb.com

[20] ArthurHidden on Freepik, Published: 24 January 2020, Free Photo: White drone hovering in a bright blue sky, Accessed: 19 January 2023, <https://www.freepik.com/free-photo/white-drone-hovering-bright-blue-sky_6659452.htm>

[21] user6702303 on Freepik, Published: 23 August 2021, Free Photo: Delivery drone concept, Accessed: 19 January 2023, <https://www.freepik.com/free-photo/delivery-drone-concept_17573701.htm#query=drone%20delivery&position=5&from_view=search>

[22] Military_Material on Pixabay, General Atomics Mq-1C Gray Eagle - Free photo on Pixabay, Accessed: 19 January 2023,< https://pixabay.com/photos/general-atomics-mq-1c-gray-eagle-2486766/>

[23] Image on Freepik, Published: 21 December 2017, Free Vector: Drone collection with realistic style, Accessed: 19 January 2023, <https://www.freepik.com/free-vector/drone-collection-with-realistic-style_1503436.htm#page=4&query=drones&position=44&from_view=search&track=sph>