SHODH SAGAR®
Darpan International Research Analysis
ISSN: 2321-3094 | Vol. 12 | Issue 3 | Jul-Sep 2024 | Peer Reviewed & Refereed

DETECTION ALGORITHM FOR DETECTING DRONES/UAVs

Armaan Parrek[a,1,*]
[a]*Electronics and Communication Engineering , IIT Roorkee, Uttarakhand, India*
arman.pareek@gmail.com

Vanshika Singh[1,2],
[a]*Electronics and Communication Engineering , IIT Roorkee, Uttarakhand, India*

Akshat Jain[1,3]
[a]*Electronics and Communication Engineering , IIT Roorkee, Uttarakhand, India*
akshat2335@gmail.com

DOI: http://doi.org/10.36676/dira.v12.i3.93

Check for updates

Accepted : 10/09/2024 Published 13/09/2024 * Corresponding author

Abstract
Unmanned Aerial Vehicles (UAVs), also popularly known as drones, have had an exponential evolution in recent times. This has resulted in better and affordable artifacts with applications in numerous fields. However, drones have also been used in terrorist acts, privacy violations and involuntary accidents in high risk zones. To address this problem, for our final year project we are working on studying and implementing various techniques and algorithms to automatically detect, identify and track small drones. We did a literature survey on the current deployed methodologies. Many state of the art techniques in recent times include Radio-Frequency, Audio-based and Radio-Frequency based methods. We mainly focused on video surveillance methods supported by computer vision algorithms. We used YOLOv5 architecture and implemented background subtraction methods within it. We modified the network to incorporate these methods. We further tested our model with the test dataset and compared the results with the benchmark models. We compared our results with the state of the art models based on visual data. We deployed our model to identify and locate drones and birds using a live camera in real time. We also tested a pruned version of our model to further improve the result. Further, we examined all the possible improvements and modifications that could be applied to our existing model to enhance the evaluation metrics.
Keywords: Unmanned Aerial Vehicles, drones, Radio-Frequency, YOLOv5

1. Introduction

Small and remotely controlled unmanned aerial vehicles (UAVs), also called drones, are of great benefit to society. They have grown in popularity as a result of rapid technological advancements in both their hardware and software, including the addition of cameras and audio recording technology, as well as the support of autonomous flying and human tracking capabilities. Drones are used in a variety of everyday tasks, including vegetation monitoring, delivery, rescue missions, and security. Despite these advantages, there has been a fast surge in the usage of drones for bad purposes such as invading privacy, security, and obstructing safety standards. Drone attacks on airports and drug smuggling usingdrones have both occurred. Drones are used to spy on and record films and audio snippets of people intheir homes, raising similar privacy issues. Drone Detection is a class of problems to tackle the issue ofmisuse of drones. It focuses on Detecting, Localizing, Tracking and taking the necessary measures to control drones. At the moment, several surveillance and detection technologies are being investigated, each with its own set of tradeoffs in terms of complexity, range, and capability. Radar, radio frequency (RF), acoustic sensors, and video surveillance with computer vision algorithms are the basic models that can be employed for drone detection and classification activities.

- Radar: Radar is a classic sensor that allows reliable detection of flying objects at long distances and near-unaffected performance in adverse lighting and weather conditions. Despite this, they frequently fail to detect small commercial UAVs with non-ballistic trajectory velocities. Radars areunable to discriminate between birds and drones due to a lack of precision in distinguishing the two. They are a costly solution due to their complicated installation and high cost.
- Visual Data: To detect drones, this method employs video or image recognition techniques. Although these methods have shown to be efficient in ideal settings, their performance is highly influenced by external elements such as weather, dust, fog, or rain, as well as other flying objects that may resemble drones, such as birds. Aside from their sensitivity, occlusion is another key challenge.
- Radio-Frequency: One of the most common anti-drone systems on the market is an RF-based UAV detection system, which detects and classifies drones based on their RF signatures. However, not all drones use RF transmission, therefore this method isn't effective for detecting unmanned aerial vehicles (UAVs) that aren't connected to the internet.
- Acoustic Sensors: Acoustic detection systems use a network of auditory sensors or microphones to recognise distinct acoustic patterns of UAV rotors, even in low-light situations. The maximum operational range of these systems, however, is less than 300 meters. Furthermore, the sensitivity of these systems to ambient noise, particularly in urban or industrial locations, as well as windy circumstances, has an impact on detection performance.

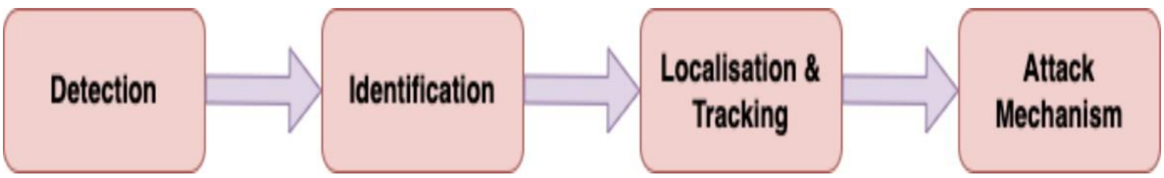


Figure 1: Anti Drone System

The above figure 1 shows the working flow of an anti-drone system. The existence of a drone within a limited area is identified in the first stage. The system then determines if the drone is authorized or illegal by assessing its features, such as the kind or model of the drone. The system should then be able to track and locate the drone. In the end, the system obstructs the drone's objective by employingseveral traditional mechanisms such as shooting drones with guns, nets, or spoofing and jamming tactics. We will be working on the Detection and Tracking aspect of our Bachelor's Degree project usingvisual data from a static camera.

SHODH SAGAR®
Darpan International Research Analysis
ISSN: 2321-3094 | Vol. 12 | Issue 3 | Jul-Sep 2024 | Peer Reviewed & Refereed

2. Moving Object Detection

This section will present the core concepts of image processing and deep learning utilized in the proposed solution. In section 2.1, the YOLO family of models are discussed. Background subtraction is a useful technique for object detection problems which we have discussed in section 2.2. Section 2.3 introduces Pruning which is a technique to improve the weights of any deep learning model. In section 2.4, we discuss the Drone vs Bird challenge. Followed by section 2.5 in which we discuss Performance Evaluation criteria for the proposed solution. Moving Object Detection is an important concept of Image Processing and Computer Vision. Automated Video surveillance has been used in many sectors these days. The basic Framework of Video surveillance consists of Environment Modeling, Motion Segmentation, Object Classification and Object Tracking. In Environment Modeling we basically have to recover and update the Background from the dynamic frames of a video taking into consideration the factors like sunlight, shadows, moving branches etc. In motion segmentation, one aims to find the regions which are related to the moving object. Most of the Motion segmentation methods these days use the temporal or spatial content of the image sequence or frames, the video is broken into. Some famous methods are Background subtraction, Temporal differencing and optical flow. Then we basically classify the object on the basis of shape, motion, feature, color or texture. After this we need to track the object from one frame to the next frame. Given below is a flow chart of the basic flow of moving object detection:

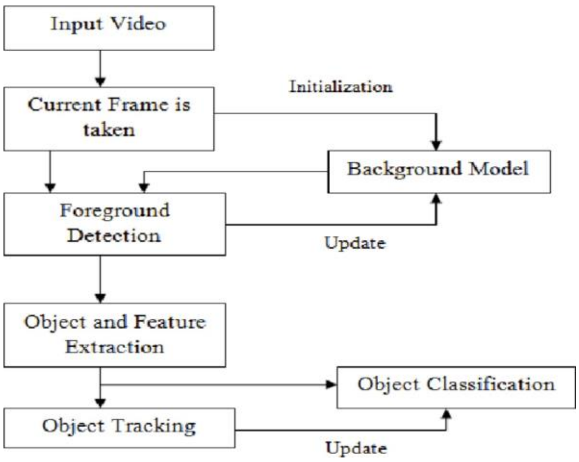
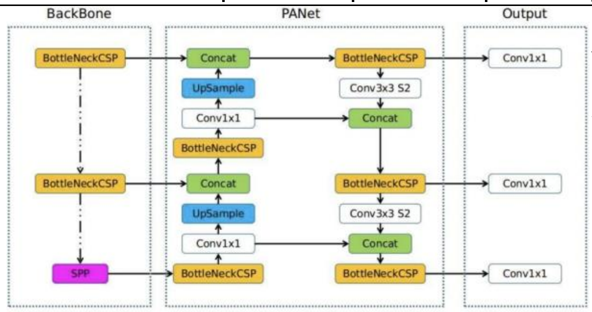


Figure 2: Basic Flowchart of Moving Object Detection In video data [5]

2.1. 2.1 YOLO

YOLO was first introduced in 2015 by Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". The motivation was to create a unified model of all phases. The system produces predictive vectors for all objects present in the image after passing through a single network of multiple CNNs. YOLO avoids iterating over different regions of an image instead it computes all the features of the images and makes predictions for all the objects. YOLO applies a grid cell on the image and checks whether the center of the object falls into the grid cell. If yes then that particular grid cell is responsible for identifying the object even if there are multiple instances of the object in different grids. Each grid predicts a bounding box based on the confidence score given by EQUATION where p(object) represents the probability of the object being inside the bounding box and IOUpred truth over the union of the prediction box and ground truth. YOLO removes all the unnecessary bounding



boxes which do not contain objects or bounding boxes which contain objects which are already present in other bounding boxes using Non maximum suppression.

Figure 3: Yolov5 Architecture Model [6]

YOLOv5 is the latest version of YOLO models and it was the first model which was written exclusively for Python. YOLOv5 architecture can be divided into three parts. First is its backbone which is similar toYOLOv4 and contains its main feature extractor CSP. Then there is its neck part containing SPP block,PANet etc. The last part is its head part which is derived straight from YOLOv3.

2.2. 2.2 Background Subtraction

Background Subtraction as the name just suggests are the techniques to remove background from images. There are different methods to achieve this. Some popular methods are using Gaussian Mixture models or Bayesian Segmentation. These probabilistic models are used to identify the color and other background related things in the image to cancel it out.

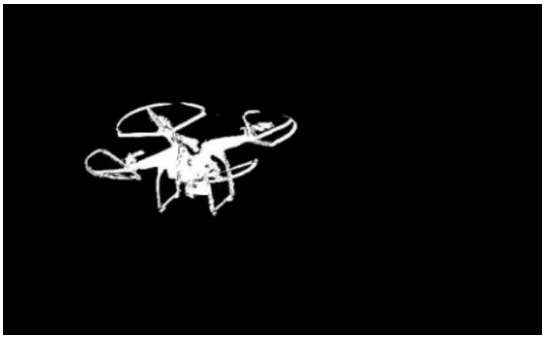


Figure 4: Background Subtracted Image from our dataset

2.3. 2.3 Pruning

Pruning is a data compression technique used to remove unnecessary information which is redun- dant and non critical for classification. Pruning reduces overfitting by simplifying the final classifier. It reduces the size of the decision tree without affecting the predictive accuracy of the classifier. Too large decision trees are vulnerable to overfitting and new instances of data can significantly impact the accuracy of the model.

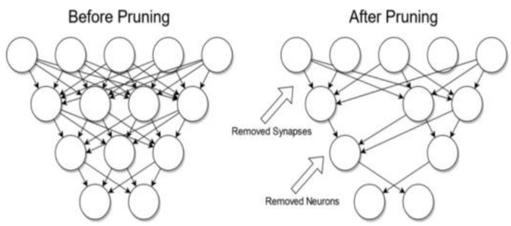


Figure 5: Before and After Pruning Result [7]

2.4. 2.4 Drone Vs Bird Challenge

The “International Workshop on Small-Drone Surveillance, Detection and Counteraction Techniques” (WOSDETC) proposed the Drone Vs Bird Challenge to invite researchers from all around the world to work on the problem of detecting drone and distinguishing it from birds from a far away distance. It requires one to create a system that raises alarm when there is a drone in sight. The Organization provides a dataset which has been regularly updated with each new installment of the challenge. The shorts video include many different types of drones with a diverse background ranging from mountains, sky, buildings etc. The developed algorithms should strive to precisely localize drones and generate bounding boxes as close to the targets as possible. The Averaged Precision metric (AP) will be used to assess the results. We mostly used the dataset from the challenge's 2020 iteration in our project.

2.5. *2.5 Performance Evaluation Criterion*

Any classification is evaluated on the basis of number of instances correctly classified and number of instances miss-classified. True positive refers to those instances which are correctly classified. False positives are those instances which were misclassified as positive. True negatives are those instances which are false in both predicted data and true data. Similarly we can say for False negative as well. These four parameters are used to make all the important metrics considered for evaluation.

TP, TN, FP and FN are true positives, true negatives, false positives and false negatives respectively.

- Accuracy It is a ratio of correct prediction and total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- Precision It is a ratio of correctly predicted outcomes over all the positive outcomes.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

- Recall It measures the sensitivity of the model and is a ratio of correctly identified positive to allthe positively identified examples.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

- F1 score It is the harmonic mean of precision and recall. It gives the performance of the model interms of both precision and recall.

$$F1score = 2 \frac{Precision \cdot Recall}{Precision + Recall} \quad (4)$$

3. Proposed Framework

3.1. Model - An altered version of YOLOv5

Our main idea was to use YOLOv5 along with background subtraction to improve YOLO capabilities for tiny objects like drones. Initially, we tried to implement this through two modules. One to identify drones solely using YOLOv5, then passing the result to a classifier identifying bird and drone. If the results are positive with a small probability the image would be then passed on to a classifier that uses background-subtracted images. So, in Module 1 we apply a basic Yolov5 model and if we get a confidence value above a particular threshold (we choose 0.15 here), then we move to a bird-drone classifier to differentiate between the detected objects. If the confidence value fails to meet the threshold we pass to module 2. In module 2, we planned to employ Background Subtraction. We start by extracting Background Subtracted frames, then dilation. In Dilation we add more pixels to the boundaries of objects in an image.This connects the closely spaced pixel and helps to reduce

the regions to be checkedby the classifier. Then we applied morphological filtering followed by deploying MobileNetv2, a small lightweight CNN model. Then we again passed the results to a bird Drone classifier used in Module 1 as well. We tried to implement this initial approach but we couldn't find the desired results. We used an ensemble inherited in the YOLOv5 repository forthis purpose. To our surprise, our ensemble did notimprove the results. On the test dataset, this ensemble was only able to produce 58.9% which even both models individually were delivering. This was because the background subtracted image's features were drastically different from a normal image. So, their correlation was poor. A major disadvantage ofthe YOLOv5 model is that it is not adaptive and so we tried to merge its architecture itself with adaptive background subtraction techniques rather than combining three different models. So we switched to trying to integrate background subtraction capability into existing YOLOv5 architecture.

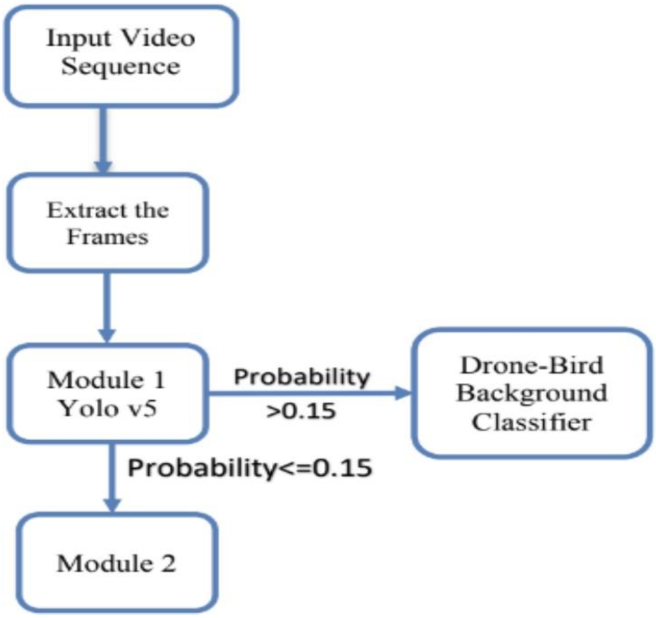


Figure 6: Initially Proposed Model-Module 1

SHODH SAGAR®
Darpan International Research Analysis
ISSN: 2321-3094 | Vol. 12 | Issue 3 | Jul-Sep 2024 | Peer Reviewed & Refereed

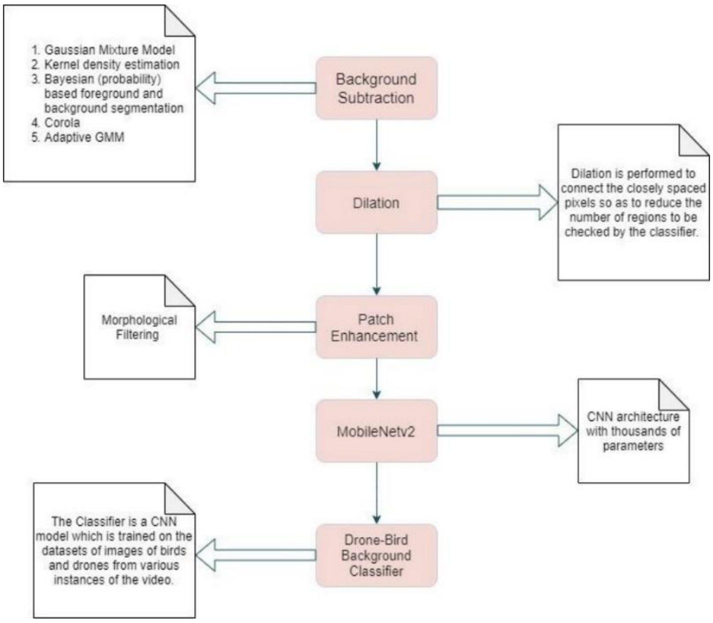


Figure 7: Initially Proposed Model-Module 2

YOLOv5 architecture can be broadly divided into three parts. Each part consists of different types ofCNN with different parameters. The first part is called its backbone, which consists of its main feature extractor, an improved version of CSP Darknet. The second part is its neck layer which divides detection into three parallel streams based on dense or sparse prediction. The final layer, also known as the output layer, consists of a detect node that combines the output of the three parallel detectors. Our main focus is to include a background subtraction method in YOLOv5 architecture to improve the detection of objects which are very similar to the background and often go undetected. So, to incorporate the background subtraction method we first used OpenCV methods and a Convolutional Layer to process the image in the background-subtracted frame. We used a CNN based Classifier to process this imageparallelly with the YOLOv5 architecture and get the final output. We merged it with the detect node, asmentioned earlier. In this way, instead of using an ensemble of two models, we have merged the two processes BG Subtraction and YOLOv5 during the training itself. To add this background subtracted model in the YOLOv5, we first need to understand the working of the YOLOv5 repository. The most important code files for YOLOv5 are present in the model folder of the repository. The file common.py contains all the required classes of different types of neural nets used in its architecture. We added a new class for background subtracting Conv2D to it. We used OpenCV to run the background subtraction after converting the PyTorch tensor to NumPy array, running background subtraction on it, then again converting it back to the PyTorch tensor. The file yolo.py is where all the model generation is present. We have modified the class model present in yolo.py to add our background-subtracted layer. Yolov5 takes input for all the hyperparameters using yaml files which are also processed in yolo.py only. We modified it as well to accommodate new hyperparameters.

3.2. Dataset

We prepared the data set for training the YOLOv5 model using the Drone vs Bird 2020 challenge dataset and random internet videos. The drone vs. bird training set consists of 77 different videos with annotations. There are 14 more videos in the challenge test set that do not have annotations. These videos share characteristics with the training set. To use YOLOv5, we first segmented these videos to obtain images, and then labeled them with a labeller that uses OpenCV and Tracking algorithms to make labeling a larger set of images easier. The videos comprise 1384 frames on average and 3 different resolutions, namely 1920 × 1080 @25 fps, 720 × 576 @50 fps, 1280 × 720 @30 fps. We labeled the images on our own as we needed the normalized values and in the same format in which YOLOv5 takes the annotations.

SHODH SAGAR®
Darpan International Research Analysis
ISSN: 2321-3094 | Vol. 12 | Issue 3 | Jul-Sep 2024 | Peer Reviewed & Refereed

Figure 8: Different Drones in the Videos of the Dataset [16]

3.3. Dataset for the initial approach

We prepared three datasets for training the YOLOv5 model from videos of resolution 1920 x 1080:

- This is the simplest data set which was made just to check that YOLOv5 can detect far away small objects too just like it detects normal moving objects like cars at traffic lights etc. We took asingle video and extracted 48 frames from it. We divided the frames into train images of 40 framesand 8 frames of validation images. We manually labeled the images. Then we trained the model using Google Collab. To improve the generalization of the model we should use more imagesfor training and validation, frames from video with different backgrounds including sequences with sky or vegetation as background, different weather types (cloudy, sunny), direct sun glare and variation in camera characteristics. The validation and training set should also contain a different set of images(from different sequences) to avoid overfitting of the model.
-

Figure 9: Training Result using the First Dataset

Figure 10: Test Result using the First Dataset

- To solve the previous problem (wrong detection of the moon as a drone), we will create a data set of contrasting videos and about 300-400 frames. We used only one class that is 'Drone' hereas well. For train images we had 538 frames from different videos and for validation images we had 180 frames. Because the model was only trained to detect one type of flying item (drones), the detector is biased toward detecting drones and may fail to distinguish them from other similar-looking flying objects (especially if they are really small). To eliminate this bias, you might either train for more object classes (acquire and annotate more photos including things like birds, planes, and helicopters) or use several frames to obtain information about their flying patterns.

Figure 11: Test Result using the Second Dataset

- In the third data set we created two classes of 'Drone' and 'Bird' and used about 150-180 frames in the train images folder and 50-80 frames for validation images. We used video of birds to train the model for different bird motions in the sky.

Figure 12: Test Result using the Third Dataset

3.4. Dataset for the altered approach

For this, we created a Train dataset of about 600 images, a validation dataset of about 200 images, and a test dataset consisting of 150 images. We planned to train with a larger number of images to improve the results. In creating this dataset we focused more on making the train and validation data diverse. We also included images of birds from different videos available online with different resolutions.

Figure 13: Frames from the Train and Validation set of Birds

4. Experimental Work

4.1. Final Model Training

Our final model, as mentioned in chapter 4, performed as well as the single-class classifier. We ran 60 epochs on the multi-class dataset. The dataset didn't contain a background-subtracted image as the model first removed the background and passed it to ConV2D for parallel detection. It was also trained on the instances in which the background was superimposing the drone to test the modificationof background subtraction. Also, to distinguish between smaller drones and birds. The data imparity mentioned earlier was evident here, looking at the volatile precision graph. But the loss functions have decreased monotonically, whether it be the overall loss function or general class loss function. This proves that our addition of the background-subtracted model to the YOLOv5 has trained well, and the two models worked cordially. Providing more data with less class imbalance and having an equal number of drones and birds may help solve this volatile behavior of precision. Our absolute precision and recall scores were 0.954 and 0.973 on the training dataset.

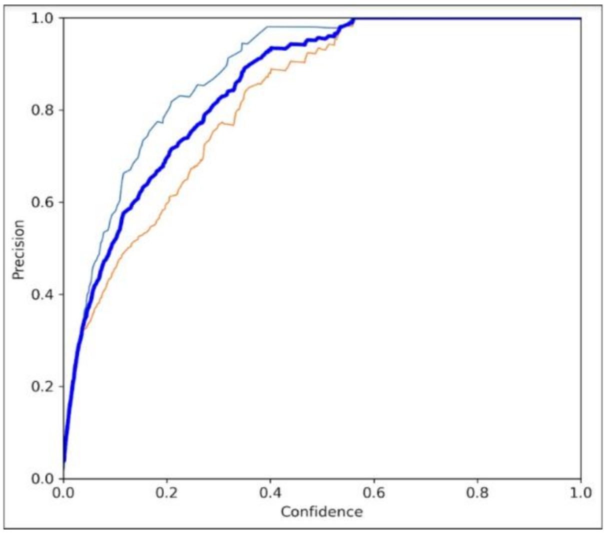


Figure 14: Enter Caption

4.2. Raspberry Pi Deployment and Real-Time Detection

We also created a deployment file that we can run on Raspberry Pi. The file can use a camera provided by the device (Pi camera) to identify drones. We have used an OpenCV based library to run the camera of the device. To use the model, it directly imports it from GitHub using Pytorch's hub class. Then it returns the label and parameters of detection after passing a frame through the model. We have also written a function that uses these labels to generate the camera's bounding box and plot these boxes on the screen using the camera. We can run our model in real-time to identify drones using Raspberry

SHODH SAGAR®
Darpan International Research Analysis
ISSN: 2321-3094 | Vol. 12 | Issue 3 | Jul-Sep 2024 | Peer Reviewed & Refereed

and a monitor.

5. Performance Evaluation

The altered version of YOLOv5 was able to produce decent results on the test dataset also. With a recision of 78.2% our implementation.

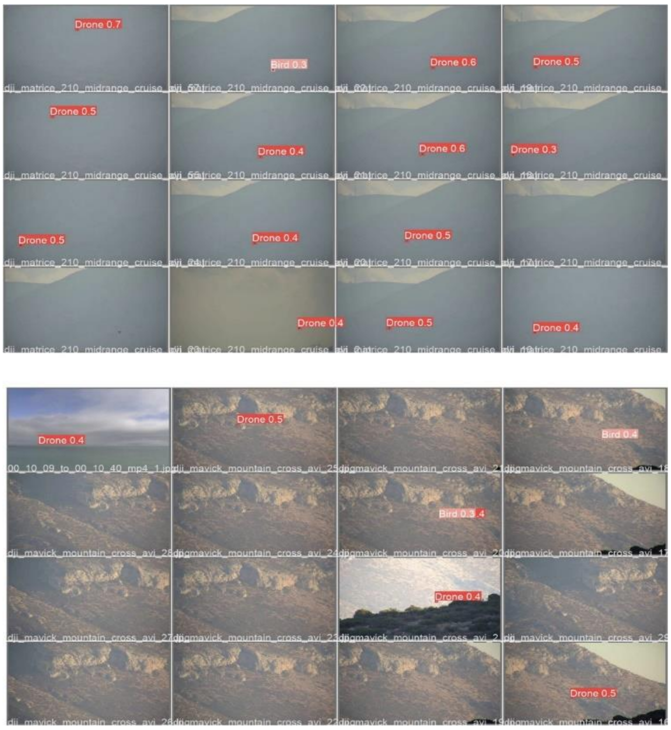


Figure 15: Result on the Test Batch using the Altered Final Model

Some objects whose spatial features are similar to drones are being misidentified as drones. This is because our dataset contains less number of negative instances. Also, our model was always able to identify white-colored drones. It sometimes fails to identify black-colored drones. This is yet again due to most drones in train videos being of white color. These are clear cases of overfitting on training data.We also tried to prune our model to improve its performance. We used Pytorch library's utils class to prune the model. After pruning, we had the updated weights for our model. As we mentioned earlier, the reason to do so was to avoid overfitting. We found a minor improvement in the performance, and precision came to around 80

6. References

1. Sarvepalli, Sarat Kumar. (2015). Deep Learning in Neural Networks: The science behind an Artificial Brain. 10.13140/RG.2.2.22512.71682.
2. F. Chollet, Deep Learning with Python, 1st. USA: Manning Publications Co., 2017, isbn: 1617294438.
3. Google Trends: https://trends.google.com/trends/explore?date=allq=dronedetection April 2020.
4. Rahaman, Muhammad Mahin, Md Ali, Md Hasanuzzaman, (2019). BHCDR: Real-Time Bangla Handwritten Characters and Digits Recognition using Adopted Convolutional Neural Network. 10.13140/RG.2.2.28972.10881.
5. Weiming Hu, Tieniu Tan, Liang Wang, Steve Maybank, A survey on visual surveillance of object motion and behaviors, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications andReviews), 34 (3) (2004) 334-352.
6. Github Repository of YOLOv5: https://github.com/ultralytics/yolov5
7. "Real-Time Drone Detection Using Deep Learning Approach": Third International Conference,
8. MLICOM 2018, Hangzhou, China, July 6-8, 2018, Proceedings
9. "Establishing Drone Detection System by Using Deep Learning and YOLOv3 Formatting", Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Man- agement Singapore, March 7-11, 202.
10. "Automated Drone Detection Using YOLOv4" by Subroto Singha and Burchan Aydin

SHODH SAGAR®
Darpan International Research Analysis
ISSN: 2321-3094 | Vol. 12 | Issue 3 | Jul-Sep 2024 | Peer Reviewed & Refereed

11. Wikipedia Page Pruning : https://en.wikipedia.org/wiki/Decision tree pruning

12. "Real-Time and Accurate Drone Detection in a Video with a Static Background" Ulzhalgas Seidaliyeva, Daryn Akhmetov, Lyazzat Ilipbayeva, Eric T. Matson Sensors (Basel) 2020 Jul; 20(14): 3856.Published online 2020 Jul 10.

13. S. Srigrarom and K. Hoe Chew, "Hybrid motion-based object detection for detecting and tracking of small and fast moving drones," 2020 International Conference on Unmanned Aircraft Systems (ICUAS), 2020, pp. 615-621, doi: 10.1109/ICUAS48674.2020.9213912.

14. "Small Object Detection using Deep Learning"-Aleena Ajaz, Ayesha Salar, Tauseef Jamal, Asif Ullah Khan

15. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks"

16. Alqaysi, Hiba et al. "A Temporal Boosted YOLO-Based Model for Birds Detection around WindFarms." Journal of imaging vol. 7,11 227. 27 Oct. 2021, doi:10.3390/jimaging7110227

17. Coluccia, A.; Fascista, A.; Schumann, A.; Sommer, L.; Dimou, A.; Zarpalas, D.; Méndez, M.; de la Iglesia, D.; González, I.; Mercier, J.-P.; Gagné, G.; Mitra, A.; Rajashekar, S. Drone vs. Bird Detection: Deep Learning Algorithms and Results from a Grand Challenge. Sensors 2021, 21, 2824.https://doi.org/10.3390/s21082824

18. Swamy, H. (2020). Unsupervised machine learning for feedback loop processing in cognitive DevOps settings. Yingyong Jichu yu Gongcheng Kexue Xuebao/Journal of Basic Science and Engineering, 17(1), 168-183. https://www.researchgate.net/publication/382654014

19. Swamy, H. (2024). Smart spending: Harnessing AI to optimize cloud cost management. International Journal of Artificial Intelligence Research and Development (IJAIRD), 2(2), 40-55. https://doi.org/10.5281/zenodo.13132258

20. Swamy, H. (2024). Leveraging AI for enhanced application service monitoring. International Journal of Computer Engineering and Technology (IJCET), 15(4), 85-99. https://doi.org/10.5281/zenodo.13131932

21. Unlu, E., Zenou, E., Riviere, N. et al. Deep learning-based strategies for the detection and tracking of drones using several cameras. IPSJ T Comput Vis Appl 11, 7 (2019). https://doi.org/10.1186/s41074-019-0059-x