Drone detection and tracking using multiple deep learning models from pictures, videos, and real-time

Abstract—**Drones are flying objects that may be remotely controlled or programmed to do a wide range of tasks, including aerial photography, videography, mapping, surveys, crop and animal monitoring, search and rescue missions, package delivery, and military operations. Misuse, however, can pose a significant danger to people's safety, privacy, and security, including eavesdropping, flying close to prohibited locations, interfering with public events, and delivering illicit products. Real-time drone detection and tracking are necessary to solve this issue, as anti-drone system development is progressing quickly. The four most often used drone detection methods are radar, acoustic, visual, and radio-frequency signal-based. Our goal is to develop a drone detection and tracking model that maximizes accuracy while minimizing loss. In order to identify drones, in this study we utilized VGG16, VGG19, ResNet50, InceptionV3, Xception, ResNet101v2, and MobileNetV2. We evaluated the accuracy and loss of the algorithms using mean squared error (mse). Results were better than earlier efforts in terms of accuracy and loss. Each of the 7 object detection models had its accuracy, loss, and IoU values tracked, and we displayed a comparison while saving the model that performed the best. The Xception model was the only 1 of the 7 that simultaneously provided the required output with the best accuracy and the lowest loss. The Xception model's accuracy was 99.185% with a loss of 3.8355, which outperformed all previously existing approaches. We chose the Xception model for drone detection and tracking because of this.**

Index Terms—**Drone detection, UAV, Object detection, Xception, OpenCV, Computer Vision, Drone Tracking, Deep learning, Drone Identification**

I. Introduction

Due to the ongoing advancement of technology, several UAV production companies like Walkera, PowerVision, and Skydio are producing different types of drones. An unmanned aerial vehicles (UAVs), sometimes known as a drone, is a flying object that can be remotely controlled or has been programmed to carry out a variety of functions. Drones can take photos, videos, and data from the air since they are fitted with cameras, sensors, and other technology. Aerial photography and videography, mapping, surveys, animal and crop surveillance, rescue operations, goods transportation, and military activities are just a few of the uses they may be put to. Drones are available in a range of sizes, from tiny quadcopters that can be flown indoors to huge unmanned aircraft that are employed in the military. In spite of drones' rising popularity, misuse can pose a serious risk to people's safety, privacy, and security. For example, spying on people without their permission or consent, flying near restricted areas like airports, military bases, and administrative buildings, interfering with public events like concerts, parades, and sporting events, and delivering illegal goods like narcotics or weapons. Drones have recently made news for infiltrating high-security places and flying over prohibited regions. Gatwick Airport in the UK was closed for 36 hours in December 2018 after a drone was seen flying close by, delaying hundreds of travelers' travel plans [1]. In January 2021, a man was detained for flying a drone in a prohibited area close to the White House [2]. A guy was detained in March 2021 for using a drone to carry narcotics to a South Carolina jail [3]. In April 2022, gun dealers smuggled 11 handguns from the USA into Canada using a big drone [4]. The occurrences using drones that were described demonstrate the necessity of watching over drone flying. In order to ensure security, several drone manufacturing firms have established no-fly zones, which forbid drones from flying within a 25-kilometer radius of particular critical zones, such as airports, jails, power plants, etc. and more vital infrastructure [5]. However, no-fly zones have a very limited influence, and not all drones are equipped with these built-in safety measures. Therefore, the development of anti-drone systems is rapidly advancing, and the issue of real-time drone identification is increasingly crucial in order to overcome this problem. Four different types of drone detection systems are commonly applied: radar, acoustic, visual, and radio-frequency signal-based [6]. Our objective is to create a drone identification algorithm that is more accurate while causing the least amount of loss, and we will identify the drone by using real-time video, video footage, and photographs. According to research, acoustic sensors, radio frequency, radar, and camera sensors backed by computer vision algorithms are the primary technologies that may be utilized for drone detection.

A. Detection of Drones Using Acoustics:

Detection of drones using acoustics is a technique that makes use of microphones to find drone sounds. By identifying and interpreting the drones' distinctive sound, it is possible to estimate the drones' position, altitude, and movement direction. This technique offers a number of benefits, including the capacity to identify drones at night or in low-light situations, as well as drones that are invisible to the unaided eye. There are various drawbacks to acoustic-based drone detection systems. Weather factors, including wind, rain, and snow, can have an influence on acoustic sensors, affecting the accuracy and range of the detecting system. Outside noise, such as that from industry or traffic, can change the acoustic profile of a drone, making it more elusive to detect. The detection capability of acoustic sensors is not limited to drones alone; they are also capable of detecting sounds produced by small animals, birds, and flying insects. There is hence a chance for false-positive detections. Due to the restricted acoustic sensor range, drones must be located near the detection system in order to be picked up. Since acoustic sensors are omnidirectional, they can pick up noises coming from any direction. Finding the drone's location and its flight route may be challenging as a result. Acoustic sensors are capable of detecting the existence of drones, but they are unable to offer details on the size, shape, or kind of the drones, which may restrict their utility in identifying and locating drones.

B. Detection of Drones Using Radar:

A popular method for locating and following drones in the air is radar drone detection. Radars are tools that identify and detect items nearby by using electromagnetic radiation. They send out a signal that bounces off nearby objects and returns to the radar, enabling it to determine the object's size, speed, and direction. In order to find drones, radar systems are made to pick up small, low-flying objects with radar cross sections that are often much smaller than those of aircraft. The radar cross sections, which varies based on the object's size, shape, and substance, measures the capacity of an item to reflect radar signals. Radar can find drones at great distances and in any kind of weather. Radar equipment may also be used to establish the position, height, speed, and direction of a drone that has been identified, which can help authorities assess the danger level and take the necessary action. . Radar systems may also be used with other security tools like cameras and alarms to offer a more complete security solution.The detection range of a drone can be influenced by factors including geography, weather, size, and speed. Due to their narrow radar cross sections and propensity for false positives, miniature drones are particularly challenging to detect using radar. Once a drone is discovered, it might be difficult to establish its precise location, especially if it is moving constantly.

C. Detection of Drones Using Radio Frequency:

Another strategy for drone detection and tracking has been developed utilizing radio frequency (RF) technology. This technology picks up the electromagnetic signals generated by drones or their controllers. Similar to radar, RF detection has a finite detection range and is susceptible to the effects of the environment, including weather and topography. Additionally, if there are obstructions or interference in the surroundings, RF detection's efficacy may also be diminished. Other electronic equipment can interfere with RF signals, making it difficult to distinguish between drone signals and other signals in the vicinity.

D. Visualized Drone Detection:

Visualized drone identification uses thermal or optical cameras to visually identify and track drones. To locate and track drones, this technique uses image analysis from the cameras' collected photos, videos, and webcam. Visual drone identification may be highly accurate when combined with machine learning algorithms that can assess the images in real-time to identify drones. This technology is a flexible choice for drone detection since it can be employed in a range of settings and lighting situations. Due to its ability to precisely identify drones based on their visual features, visualized drone identification can also have a lower false-positive rate than other detection techniques. Our approach uses drone images, videos, webcams, and different computer vision algorithms to detect drones.

II. Related Works

Manja Wu et al. [7] created a real-time drone detector using deep learning. To hasten the training of the learned pictures, they created a semi-automatic dataset using a KCF tracker. By modifying the resolution structure of the input photos and tweaking the anchor box's size parameters, the YOLOv2 deep learning model was created. An anti-drone dataset tagged with a KCF tracker and a drone dataset from the University of Southern California were used to train the network. The suggested detector by the authors provided good results in real-time detection at a reasonable system cost. Based on footage taken using stationary cameras, Wang et al. [8] suggested a quick, easy, and effective detection method for unmanned aerial vehicles. Moving items were recognized using the temporal median background removal technique, and global Fourier descriptors and local HOG features were extracted from pictures of moving objects. The SVM classifier conducted classification and recognition using the combined Fourier descriptor (FD) and HOG features. The authors experimentally demonstrated that the suggested FD and HOG algorithms could execute the task of categorizing birds and drones with better accuracy than the GFD algorithm. The total accuracy of the proposed recognition technique was 98%.To identify unmanned aerial vehicles, Peng et al. [9] created a sizable training set of 60,480 generated photos. In the manually annotated UAV test set, the faster R-CNN network performed with an average accuracy of 80.69%, compared to 43.03 percent in the pre-trained COCO (Common Objects in Context) 2014 dataset and 43.36% in the PASCAL VOC (Visual Object Classes) 2012 dataset. This network was tuned to Detectron, as advised by Facebook AI research. In comparison to previous techniques, the faster R-CNN detection network's average accuracy was considerably greater when trained on rendered pictures.Using YOLOv4, Singha et al. [10] developed an automated drone detection system. The model was tested using mean average precision (mAP), frames per second (FPS), precision, recall, and F1-score on drone and bird datasets. Results outperformed other research of a similar kind, with an F1-score of 0.79, an accuracy of 0.95, a recall of 0.68, and a mAP of 74.36%. On the DJI Phantom III and DJI Mavic Pro, video detection reached a frame rate of 20.5 and 19.0, respectively. Behera et al. [11] suggested a model with an accuracy of 98.33% for drone detection and 97.5% for drone classification. The findings were attained utilizing a dataset of 10,000 photos that included and did not include drones. Convolutional neural network (CNN) architecture was employed for pre-processing, feature extraction, and classification as part of the authors' deep learning-based methodology. Hamatapa et al. [12] suggested two techniques for employing motion detection and image processing to find and follow an unmanned aerial vehicle at a distance of 350 feet during the day. They reduced memory use, converted RGB images to grayscale, altered noise levels, and removed noise using OpenCV. Four different drone models—the Phantom 4 Pro, AgrasMG-1s, Pocket DroneJY019, and MavicPro—as well as birds and balloons were used to evaluate the system. In order to identify drones from birds in video and estimate their location, C. Aker et al. [13] modified and improved the single-stage YOLOv2 [14] method. The researchers combined genuine drone and bird photographs with video frames from coastal regions to produce a synthetic dataset, from which they removed the backgrounds from the photographs. Precision-recall (PR) curves were used to evaluate the suggested network, with precision and recall levels both reaching 0.9 at the same time. Before using the Faster-RCNN detector, Magoulianitis et al. [15] preprocessed the images using the Deep CNN with Skip Connection and Network-in-Network (DCSCN) super-resolution approach [16] . As a consequence, the detector's recall performance improved, and it could now detect very faraway drones. Recall and accuracy scores for the task were 0.59 and 0.79, respectively.

For the purpose of detecting drones in this work, we used the algorithms VGG16 [17], VGG19 [18], ResNet50 [19], InceptionV3 [21], Xception [22], ResNet101v2 [20], and MobileNetV2 [23]. We gathered photos of drones from Roboflow to train these neural network designs. To assess the algorithms' accuracy and loss, we utilized mean squared error (mse) as our assessment metric. In comparison to earlier efforts, our algorithms' accuracy was greater and our loss was lower.

III. Methodology

In this study, we concentrated on the identification and tracking of drones using real-time video, still photos, and image data. The complete procedure is broken down into a number of subsections, which are demonstrated below:

A. *Dataset Preperation:*

When using deep learning techniques to solve a problem, data collection is the initial step. Data is required to identify a problem's solution since deep learning techniques cannot function without it. Deep learning methods have an overfitting condition when the dataset is too short, which prevents them from learning properly. Our objective is to detect and locate drones from photos, videos, and webcam footage. We first need an image dataset with drones in it. We can gather information from a variety of sources. We obtained image data from Roboflow for our investigation.Fig. 1 displays a few of the dataset's images:

Fig. 1. Samples of the dataset's images

We must label the dataset after it has been collected. We are labeling the dataset with Roboflow annotator tools and standardizing label data files in XML annotation format. The following set of labeled images includes some:

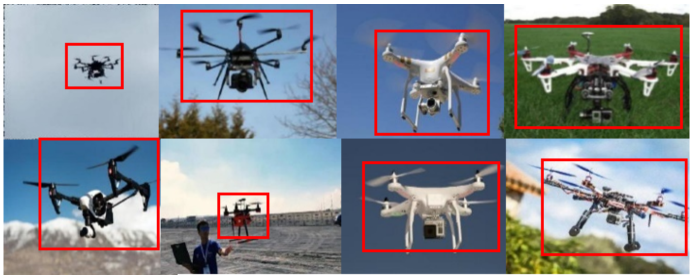


Fig. 2. Samples of the labeled dataset's images

The collected images include a wide range of sizes, shapes, and backgrounds. We are utilizing an 8167 annotator XML file and 8167 colored drone images. The amount of data is adequate to guarantee diversity and comprehensiveness in training and testing.

B. *Image Preprocessing:*

We must preprocess the obtained images before training since they have a wide range of forms, sizes, and backgrounds. The images must be in a format that the algorithms we're utilizing can process. The images in the Roboflow dataset are 640 x 640 in size. But 224 x 224 is the most suitable size for image data for our models. Therefore, the images must be downsized from 640 x 640 to 224 x 224. Then, we used the imread function to process the resized images. MATLAB's image processing toolbox contains a function called imread that reads an image file into a matrix. More specifically, it reads and saves in a matrix the pixel values from an image file. After converting the images to arrays of pixel values, we must scale the annotator values to correspond to the 224 x 224 image size. Our data is now prepared for training.

C. *Models for Drone Detection and Tracking:*

We chose seven object detection models and trained them with our preprocessed data in order to detect and track drones in images, videos, and real-time footage. We used Mean Squared Error to monitor the accuracy and loss of the

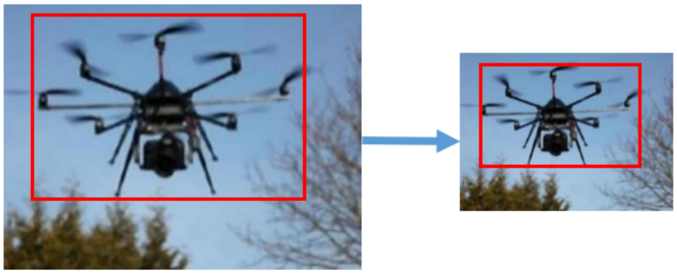


Fig. 3. A sample of a preprocessed image from 640 x 640 to 224 x 224

object detection models throughout training. We complete our drone detection and tracking attempt using the model with the highest accuracy and the least amount of loss. Here is a demonstration of each of the seven object detection models: VGG16 [17], VGG19 [18], ResNet50 [19], InceptionV3 [21], Xception [22], ResNet101v2 [20], and MobileNetV2 [23].

1) VGG16: The Visual Geometry Group at the University of Oxford created the deep convolutional neural network (CNN) model VGG16 [17] for image identification. It has 16 layers, including 3 fully connected layers and 13 convolutional layers. By using non-linear activation functions and convolving filters, the model develops the ability to extract features from pictures during training. A max-pooling layer is next applied to the output of each convolutional layer. A probability distribution for each of the 1000 classes in the ImageNet dataset is produced by the last layer. The architecture of VGG16 [17] has become a common place to start when creating more complicated CNN models for image classification, object identification, and other computer vision applications. VGG16 [17] has achieved state-of-the-art performance on a variety of image recognition tasks.

2) VGG19: VGG19 [18] is a deep CNN model that consists of 19 layers, 16 of which are convolutional layers and 3 of which are fully connected layers. It has similar performance to VGG16 [17] and was trained on the ImageNet dataset; however, it contains more parameters. It has been extensively utilized for image identification applications, serving as an inspiration for several other deep CNN models, and has considerably advanced computer vision.

3) ResNet50: ResNet50 [19] is a deep convolutional neural network architecture that Microsoft Research unveiled in 2015. To avoid disappearing gradients during training, it contains 50 layers and is made up of residual blocks. State-of-the-art performance has been attained on a variety of computer vision tasks, including semantic segmentation, object identification, and picture classification. It is typically utilized as a transfer learning model that has already been trained.

4) InceptionV3: Google unveiled InceptionV3 [21] in 2015, a convolutional neural network architecture. It makes use of ”inception modules” to record characteristics at various sizes, both local and global. During training, it additionally uses batch normalization and an additional classifier. On a variety of computer vision tasks, such as picture classification, object recognition, and semantic segmentation, it has attained state-of-the-art performance. It has drawn a lot of interest as a pre-trained transfer learning model and serves as the basis for several well-known object detection systems.

5) Xception: François Chollet proposed the deep convolutional neural network architecture called Xception [22] in 2016. It uses depth-wise separable convolutions to increase performance and efficiency. It has been extensively utilized for a variety of computer vision applications, including posture estimation, segmentation, and object recognition. Additionally, it has the benefit of lowering computing costs and parameter counts while retaining high accuracy, making it particularly advantageous for embedded and mobile devices with constrained computational capabilities.

6) ResNet101V2: A deep convolutional neural network architecture called ResNet101V2 [20] was released in 2016 as an upgrade to the original ResNet architecture. It is a variation of the ResNet101V2 [20] design, a more complex version of ResNet50 [19], and it includes 101 layers. The use of skip connections, which aid in addressing the issue of disappearing gradients, is the fundamental innovation of ResNet101V2 [20]. It has received widespread application for several computer vision tasks and has attained state-of-the-art performance on picture categorization benchmarks. Overall, the development of computer vision research and applications has been greatly aided by ResNet101V2 [20] robust and effective deep learning architecture.

7) MobileNetV2: Google researchers unveiled a deep convolutional neural network architecture called MobileNetV2 [23] in 2018. It is a modification of the first MobileNet architecture made for portable and embedded devices with constrained processing capabilities. Inverted residual blocks are a unique block structure that requires fewer parameters and calculations while maintaining good accuracy. It has attained cutting-edge performance on benchmarks for image classification and is frequently utilized as the framework architecture for many computer vision tasks. In conclusion, MobileNetV2 [23] is a strong and effective deep learning architecture that is especially beneficial for mobile and embedded devices with constrained computational resources as well as for applications requiring real-time performance.

From the models that were above demonstrated, the Xception model outperformed others with the highest accuracy and the lowest loss. Therefore, we will use this Xception model to achieve our drone detection goal.

D. Drone Detection and Tracking Process:

In this study, drone detection was done using drone images, videos, and real-time footage. The methods utilized for drone detection differ based on the type of data being used. Below is an overview of the steps for each type of data:

1) Drone Detection from Images: Since our model was trained on 224 x 224-size images, preprocessing the image to that size is necessary before we can detect drones from it. In order for the model to forecast the location of the drone, if there is one, we must now load the desired model and provide the image to the model. In some machine learning frameworks, such as Keras and TensorFlow, the predict function, a method for making predictions on new data using a trained model, produces the prediction. The rectangle function, which is a component of the OpenCV library, a well-known computer vision library that offers tools and functions for a variety of image and video processing tasks, is now required to distinguish the drone from the entire image. To draw a rectangle on an image, use the rectangle function. Last but not least, we require the show function from the popular Python data visualization toolkit matplotlib in order to see the detection and tracking. The matplotlib figures and graphs are all displayed using the show function. The general procedure is displayed below:

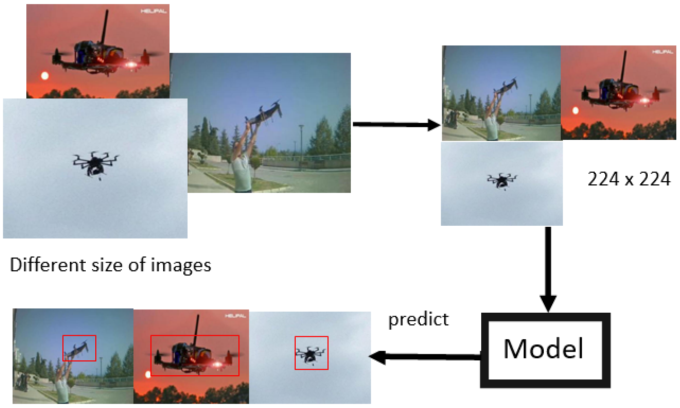


Fig. 4. Drone detection process from image

2) *Drone Detection from Videos:* We must scale the video to 224 x 224 in order to recognize drones from video data because the model we want to use was trained on 224 x 224 images. The video must now be split into numerous picture frames. After dividing the photo frames, we must load the model and give it the appropriate photo frames.In some machine learning frameworks, such as Keras and TensorFlow, the predict function, a method for making predictions on new data using a trained model, produces the prediction. Using the rectangle function, the model will then create a rectangle around the drone and forecast its location based on those photo frames. Following the prediction, the photo frames will be converted to videos and then visualized with the help of the show function. The general procedure is displayed below:

3) *Drone Detection from real-time footages:* We need to first open the webcam using the video capture feature in order to identify the drone from real-time footage. To capture video from a camera, we need to use OpenCV's VideoCapture function. The VideoCapture function will record video using our computer's default camera when argument 1 is passed to it. Our desired model was trained on 224 x 224-pixel images, so we must scale the video to that size. The video must now be split into numerous picture frames. After dividing the photo frames, we must load the model and give it the appropriate photo frames. In some machine learning frameworks, such as Keras and TensorFlow, the predict function, a method for making predictions on new data using a trained model, produces the prediction. Using the rectangle function, the model will then create a rectangle around the drone and forecast its location based on those photo frames. Following the prediction, a loop will be used to reconstruct the photo frames into a video. We must utilize the imshow function inside a loop that collects frames from a video stream or camera in order to see real-time footage in OpenCV.

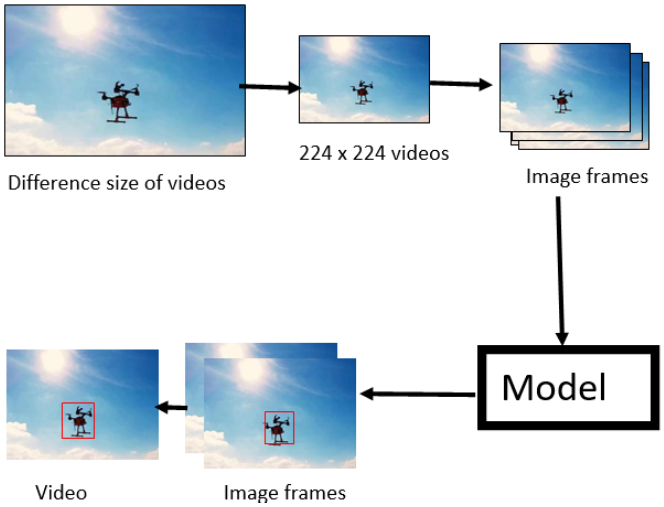


Fig. 5. Drone detection process from videos

IV. Experimental Results

Our objective is to track and identify drones using images, videos, and real-time footage. Initially, we require a collection of images that includes drones. For this reason, we obtained 8167 pictures from Roboflow. Our dataset was split into a training set and a testing set with a ratio of 0.9 and 0.1, respectively. With a ratio of 0.7 and 0.2, the training set was further divided into the training and validation sets. Here is a breakdown of the splits:

TABLE I
Summary of the Dataset

Dataset				
	Training Set		Testing Set	Total
	Training	Validation	Testing	
Ratio	0.7	0.2	0.1	1.0
Number of images	5716	1633	818	8167

We also used 6 YouTube clips of varied lengths for testing purposes.After the dataset had been fully collected, we downsized the picture data from different sizes to 224 x 224. The following is an expression of the picture resizing formula:

$$\frac{a}{b} = \gamma \tag{1}$$

$$a * \gamma = \alpha \tag{2}$$

$$b * \gamma = \beta \tag{3}$$

Here, γ is the scale factor, α is the new width of the image, β is the new height of the image, a and b are the original width and height of the image.
After that, we adjusted the annotator values to fit the 224 x 224 image size using the following formula:

$$\frac{x}{a} * 224 = \lambda \tag{4}$$

Here, λ is the new annotator value, x is the annotator value of the original image, and a is the original size of the image.

A. Training Details:

In order to identify and track drones in photos, videos, and real-time recordings, we employed seven pretrained object detection models of the TensorFlow object detection API. We used the Tesla T4 GPU from Google Collabs to train the models. Then, for running 100 epochs using weights from ImageNet, we established the batch size of the data as 32. To track the models' accuracy and loss, we used the mean squared error. For improved learning, we employed the Adam optimizer with a learning rate of 0.0001. In the output layer, we employed 1 fully connected dense layer with a linear activation function. In the beginning, while the epochs were running, we trained the VGG19 model and kept an eye on its accuracy and loss calculation. The following are some illustrations of training epochs in which accuracy and loss improved:

TABLE II
PERFORMANCE RECORD FOR VGG19 MODEL

Epoch No.	Accuracy	Loss
1	0.63809	2214.1086
9	0.86998	442.9437
37	0.94560	105.1398
91	0.97887	12.9733
98	0.98431	7.9067

As we can see above, as epochs pass, the model's accuracy value increases and its loss value decreases. The final VGG19 model accuracy was 98.431% with a loss of 7.9067. Accurate drone detection and precise drone position prediction require a high value for accuracy and a low value for loss. We trained 6 more object detection models in order to increase accuracy and decrease loss. After completing the VGG19 model's training, we trained the VGG16 model using 100 epochs in an effort to increase accuracy and decrease loss. Even lower than the VGG19 model, our accuracy after training the VGG16 model was 98.10% with a loss of 15.02. Next, we performed 100 training epochs on the MobileNetV2 model. The MobilenetV2 track record of performance is as follows:

The accuracy of the MobileNetV2 model was 98.851%, which is greater than that of the VGG19 and VGG16 models, as demonstrated above. The MobileNetV2 model loss was 9.0970, which is less than the VGG16 model but more than the VGG19 model. We further trained the Resnet50 model to detect drones more precisely for this reason.The Resnet50 track record of performance is as follows:

TABLE III
PERFORMANCE RECORD FOR MOBILENETV2 MODEL

Epoch No.	Accuracy	Loss
1	0.72304	1577.6967
4	0.93363	117.1033
16	0.94871	65.7381
31	0.97361	33.6515
38	0.97431	30.7491
40	0.98036	33.0461
51	0.98280	22.1523
60	0.98473	19.4641
75	0.98773	16.8436
92	0.98851	9.0970

TABLE IV
PERFORMANCE RECORD FOR RESNET50 MODEL

Epoch No.	Accuracy	Loss
2	0.89663	314.7631
3	0.92109	238.1099
12	0.95441	98.2536
18	0.97256	58.9677
32	0.97563	30.5070
35	0.97703	34.8471
50	0.98431	25.5513
54	0.98457	16.3150
68	0.98510	12.6993

As we can see, the accuracy of the Resnet50 model was 98.510%, with a loss of 12.6993 after training. Compared to earlier models, this time the accuracy was lower and the loss was higher. To achieve our aim, we once again trained the Resnet101V2 object detection model. The accuracy we obtained was 98.75%, and the loss was 5.5813, which was the lowest of all the models previously trained. However, the accuracy was lower. We trained the InceptionV3 model because we wanted to detect and track the drone even more precisely.The InceptionV3 track record of performance is as follows:

TABLE V
PERFORMANCE RECORD FOR INCEPTIONV3 MODEL

Epoch No.	Accuracy	Loss
3	0.90935	161.3831
5	0.93139	124.5265
9	0.95467	43.0933
12	0.97177	31.9648
21	0.97510	27.4512
35	0.98220	13.0324
69	0.98482	16.8574
70	0.98799	10.1683
71	0.98948	7.4961

As we can see, the InceptionV3 model outperforms earlier models by having an accuracy of 98.948% and a loss value of 7.4961. However, we wish to minimize the loss as much as we can. So, in order to get the loss value we wanted, we trained our final object detection model, the Xception model. Below is a performance history for the Xception model:

TABLE VI
PERFORMANCE RECORD FOR XCEPTION MODEL

Epoch No.	Accuracy	Loss
1	0.67806	2009.8859
2	0.81974	474.6008
3	0.88015	232.6906
4	0.89514	168.8375
5	0.90813	175.6783
6	0.93161	116.3893
9	0.96370	49.2952
28	0.97850	32.8853
32	0.97861	21.0095
65	0.98561	13.7156
78	0.98851	13.7388
81	0.99185	3.8355

Our goal is to build a model with a high degree of accuracy and low loss. Low loss indicates that the anticipated output is quite close to the actual output, while high accuracy indicates that the model is generally producing accurate predictions. We can observe from the Xception model's performance record table that it has the maximum accuracy of **99.185%** and the lowest amount of loss at **3.8355**. Comparing all the models we have trained up to this point, the Xception model has the best performance. We were able to detect and track the drones more precisely than previously with our model. Consequently, the Xception model outperformed all 7 of our pretrained object detection models.

B. Comparison of Findings:

We compared the outcomes after our 7 object detection models' training was successful. We compared accuracy, loss, trainable parameters, IoU, picture frame division time, and other factors in this section. The following figures are required to show comparisons between the 7 models:

We used the following formula to determine the IoU of the models on 32 images:

$$\frac{w}{z} = \sigma \tag{5}$$

Here, w is the area of intersection, z is the area of union and σ is the IoU(Intersection over Union)

TABLE VII
COMPARISON OF IOU BETWEEN 7 MODELS

Model Name	Above 75%	Below 75%
VGG16	17	15
VGG19	15	17
InceptionV3	19	13
Resnet50	15	17
Resnet101V2	22	10
MobileNetV2	3	29
Xception	27	5

C. Testing Details:

1) Testing Image data:: We used images of various backgrounds for testing purposes. If the image contains any drones, a bounding box is displayed around each one. The red bounding box represents the anticipated bounding box values predicted by our models, while the green bounding box displays the actual bounding box values. Following are a few test results:

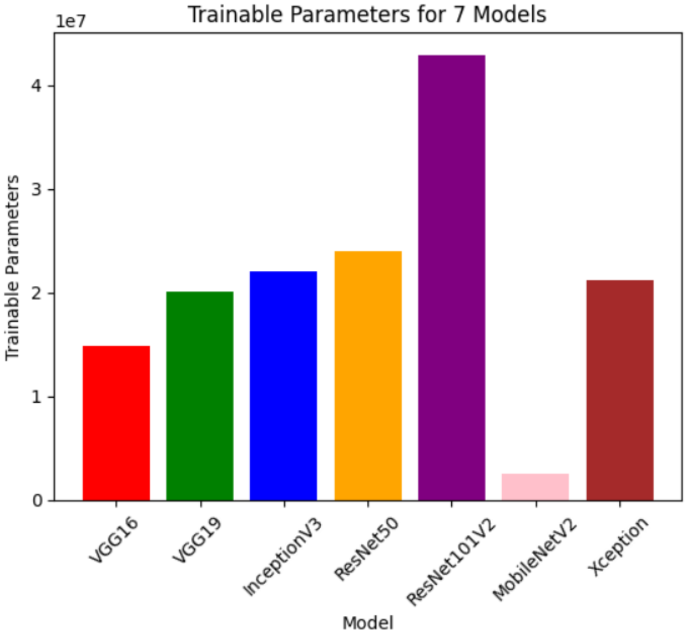


Fig. 6. Comparison of trainable parameters between 7 models

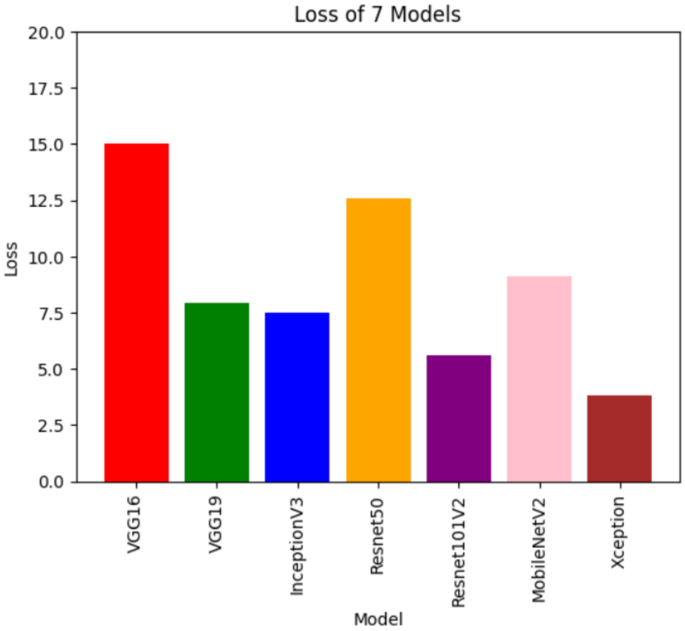


Fig. 7. Comparison of loss between 7 models

D. Testing Video and Real-Time data:

Drone tracking and detection are both done for video data. In a video, the drone is moving continuously, so we must continuously track its location. The models detect and track the drone by breaking the video down into picture frames. We

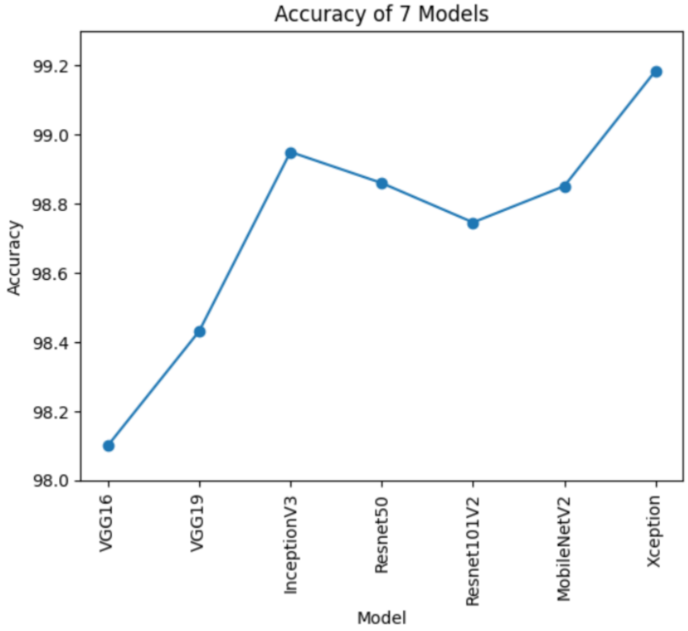
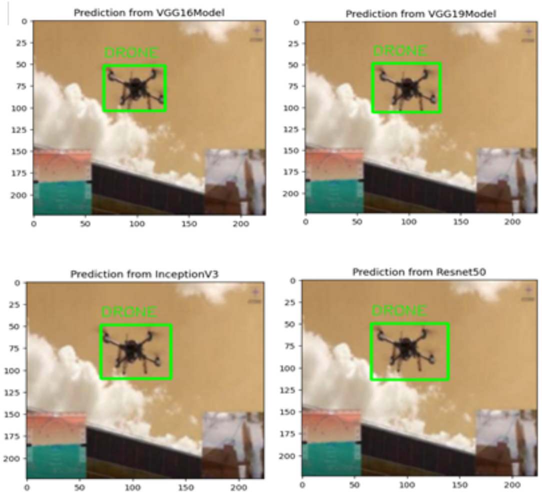


Fig. 8. Comparison of accuracy between 7 models


Prediction from VGG16Model
Prediction from VGG19Model
Prediction from InceptionV3
Prediction from Resnet50
DRONE
0
25
50
75
100
125
150
175
200
0
50
100
150
200

split the video into picture frames by setting the FPS value initially to 10. The amount of time needed to divide a picture frame and a few examples of test results are shown below:

TABLE VIII
IMAGE FRAME TIME

Frame No.	Time
Frame 1	24ms
Frame2	48ms
Frame 3	43ms
Frame 4	35ms
Frame 5	66ms

V. Conclusion and Future Works

In this research, 7 object detection models were trained to detect and track drones. Compared to earlier research of a similar kind, our model performed better. Considering that drone participation in illegal and emergency jobs is common, drone detection is required. But because of their small size, quick speed, and high altitude, drones can be challenging to detect at different elevations. In this paper, we provide multiple neural network object detection models for drone detection and tracking. To evaluate the effectiveness of using deep learning in drone tracking, we trained and tested these models.In further work, we'll focus on training advanced

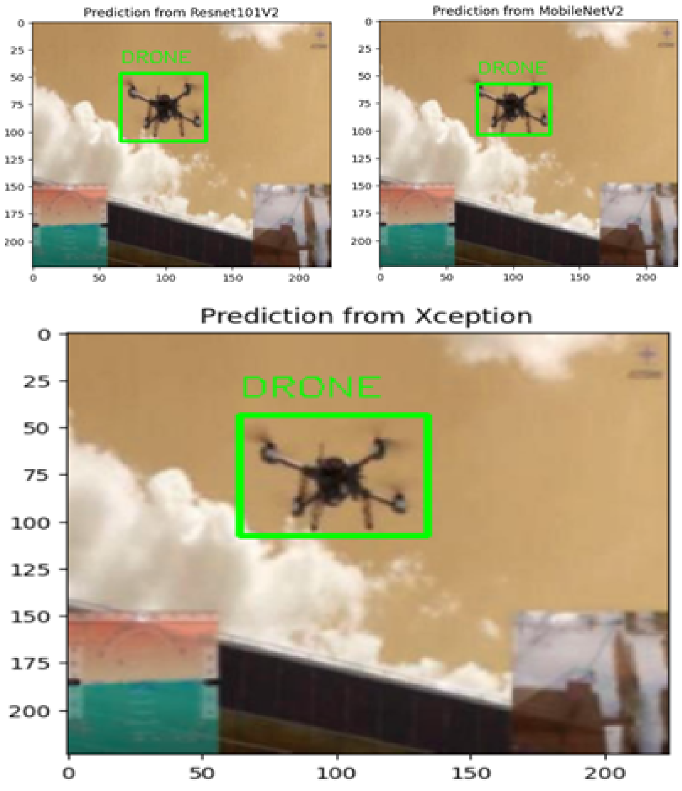


Fig. 9. Visual comparison between 7 models

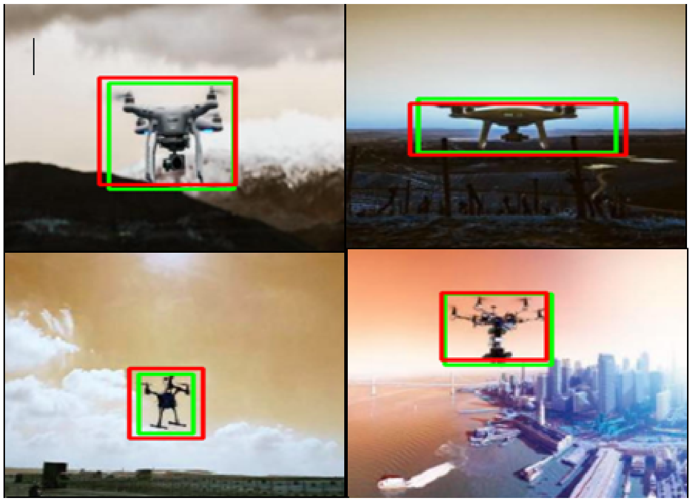


Fig. 10. Test outcome of drone images

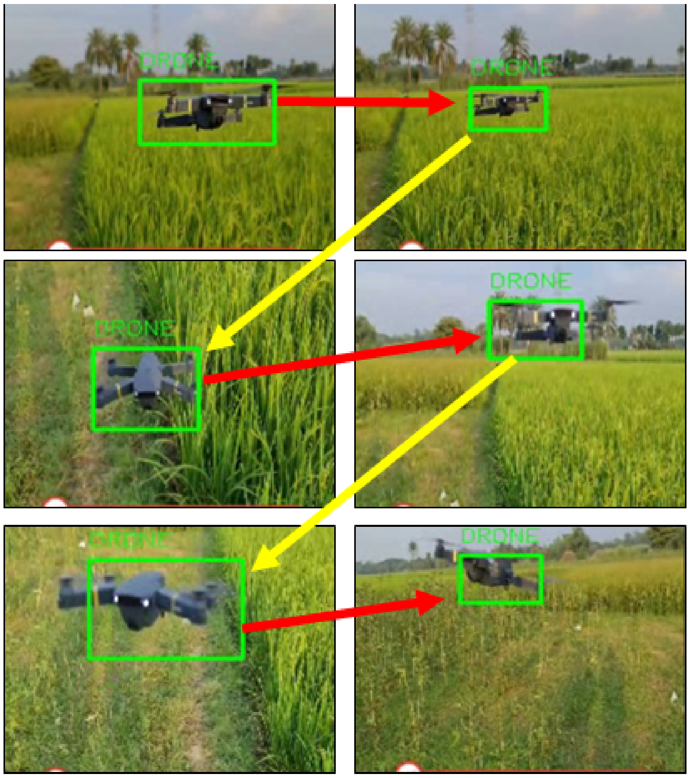


Fig. 11. Drone video where the drone is moving, detects, and tracks

object detection algorithms that can improve efficiency.

Acknowledgment

The project would not have been easy without the incredible help of Sovon Chakraborty and Muhammad Borahn Uddin Talukdar, two of our lecturers. We sincerely acknowledge the amazing assistance and collaboration of our co-authors in completing this study. Their suggestions, expertise, and resources were essential to the success of our research. We would also like to thank everyone for their thoughtful remarks and enlightening inquiries over the course of the research. Finally, we would like to thank everyone for voluntarily contributing their time and effort to this project. The task would not have been possible without them. We shall remember to express our gratitude to our family, particularly to our parents. Through this process, their efforts have kept us enthused and optimistic.

References

[1] BBC News. (2018, December 21). Gatwick drone chaos: Ministers criticised for lack of action. Retrieved from https://www.bbc.com/news/uk-england-sussex-46623754.

[2] BBC News. (2021, January 19). Drone pilot arrested for flying near White House. https://www.bbc.com/news/world-us-canada-55762127.

[3] Associated Press. (2021, March 22). Man used drone to deliver drugs to South Carolina prison, officials say. https://apnews.com/article/sc-state-wire-south-carolina-columbia-drug-crimes-arrests-ee4b4d4aa5e5b5c73d0df03e68c463cf.

[4] Tapper, J. (2022, May 3). Drone carrying bag of guns intercepted at US-Canada border. The Guardian. https://www.theguardian.com/world/2022/may/03/drone-us-canada-border-intercepted-bag-guns.

[5] Coluccia, A., Saqib, M., Sharma, N., Blumenstein, M., Magoulianitis, V., Ataloglou, D., Dimou, A., Zarpalas, D., Daras, P., Craye, C., & others. (2019). Drone-vs-Bird Detection Challenge at IEEE AVSS2019. In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 18-21 September 2019, Taipei, Taiwan.

[6] Taha, B., & Shoufan, A. (2019). Machine learning-based drone detection and classification: State-of-the-art in research. IEEE Access, 7, 138669-138682. https://doi.org/10.1109/ACCESS.2019.2944877

[7] Wu, M., Xie, W., Shi, X., Shao, P., & Shi, Z. (2018). Real-Time Drone Detection Using Deep Learning Approach. In Proceedings of the 2018 3rd International Conference on Machine Learning and Intelligent Communications (MLICOM) (pp. 22-32). Hangzhou, China.

[8] Wang, Z., Qi, L., Tie, Y., Ding, Y., & Bai, Y. (2018, October). Drone detection based on FD-HOG descriptor. In 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC) (pp. 433-4333). IEEE.

[9] Peng, J., Zheng, C., Lv, P., Cui, T., Cheng, Y., & Lingyu, S. (2018). Using images rendered by PBRT to train faster R-CNN for UAV detection.

[10] Singha, S., & Aydin, B. (2021). Automated drone detection using YOLOv4. Drones, 5(3), 95.

[11] Behera, D. K., & Raj, A. B. (2020). Drone Detection and Classification using Deep Learning. In 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1012-1016). Madurai, India: IEEE. doi: 10.1109/ICICCS48265.2020.9121150.

[12] Hamatapa, R., & Vongchumyen, C. (2019). Image processing for drones detection. In Proceedings of the 2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST), 2-5 July 2019, Luang Prabang, Laos.

[13] Aker, C., & Kalkan, S. (2017). Using deep networks for drone detection. In Proceedings of the 2017 IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 29 August-1 September 2017, Lecce, Italy.

[14] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. arXiv preprint arXiv:1612.08242. Retrieved from https://arxiv.org/pdf/1612.08242.pdf

[15] Magoulianitis, V., Ataloglou, D., Dimou, A., Zarpalas, D., & Daras, P. (2019). Does deep super-resolution enhance UAV detection? In Proceedings of the 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-6). Taipei, Taiwan.

[16] Yamanaka, S., Kuwashima, S., & Kurita, T. (2017). Fast and accurate image super resolution by deep CNN with skip connection and network in network. Paper presented at the International Conference on Neural Information Processing (ICONIP). Retrieved from https://arxiv.org/ftp/arxiv/papers/1707/1707.05425.pdf

[17] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[18] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations (ICLR) (pp. 1-14), San Diego, CA, USA, May 7-9, 2015. Retrieved from https://arxiv.org/pdf/1409.1556.pdf

[19] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770-778). doi: 10.1109/CVPR.2016.90.

[20] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In P. H. S. Torr, D. W. Murray, & B. Jegou (Eds.), European Conference on Computer Vision (pp. 630-645). Springer, Cham.

[21] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2818-2826). doi:10.1109/CVPR.2016.308.

[22] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1800-1807). doi:10.1109/CVPR.2017.195.

[23] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4510-4520). doi: 10.1109/CVPR.2018.00474.