applied sciences

MDPI

Article

CNN Based Image Classification of Malicious UAVs

Jason Brown [1,*], Zahra Gharineiat [2] and Nawin Raj [3]

[1] School of Engineering, University of Southern Queensland, Springfield Central, QLD 4300, Australia
[2] School of Surveying and Built Environment, University of Southern Queensland, Springfield Central, QLD 4300, Australia
[3] School of Mathematics, Physics and Computing, University of Southern Queensland, Springfield Central, QLD 4300, Australia
* Correspondence: jason.brown2@usq.edu.au; Tel.: +61-7-3470-4026

Featured Application: An application of this research is UAV identification (in terms of make and model) based upon a malicious UAV report that includes a photo of the suspicious UAV.

Abstract: Unmanned Aerial Vehicles (UAVs) or drones have found a wide range of useful applications in society over the past few years, but there has also been a growth in the use of UAVs for malicious purposes. One way to manage this issue is to allow reporting of malicious UAVs (e.g., through a smartphone application) with the report including a photo of the UAV. It would be useful to able to automatically identify the type of UAV within the image in terms of the manufacturer and specific product identification using a trained image classification model. In this paper, we discuss the collection of images for three popular UAVs at different elevations and different distances from the observer, and using different camera zoom levels. We then train 4 image classification models based upon Convolutional Neural Networks (CNNs) using this UAV image dataset and the concept of transfer learning from the well-known ImageNet database. The trained models can classify the type of UAV contained in unseen test images with up to approximately 81% accuracy (for the Resnet-18 model), even though 2 of the UAVs represented in the UAV image dataset are visually similar, and the fact that the UAV image dataset contains images of UAVs that are a significant distance from the observer. This provides a motivation to expand the study in the future to include more UAV types and other usage scenarios (e.g., UAVs carrying loads).

Keywords: UAV; drone; image classification; Convolutional Neural Networks

Citation: Brown, J.; Gharineiat, Z.; Raj, N. CNN Based Image Classification of Malicious UAVs. *Appl. Sci.* **2023**, *13*, 240. https://doi.org/10.3390/app13010240

Academic Editor: Wei Huang

Received: 18 November 2022
Revised: 21 December 2022
Accepted: 21 December 2022
Published: 24 December 2022

1. Introduction

The prevalence of Unmanned Aerial Vehicles (UAVs) or drones used for such applications as delivery of goods, remote sensing, surveying, inspection, and recreation has been increasing over the past decade [1]. Like most technologies, UAVs can be misused. The motivation for such malicious use may be to cause annoyance, invade privacy, cause physical harm or even to shutdown airspace with its associated economic impact [2,3].

Once a malicious UAV is detected, there are various countermeasures that can be considered, including capturing or destroying the UAV, jamming its wireless link so it cannot be controlled or report data back, and identifying and fining the owner [4]. Depending upon the exact scenario, the detection may depend upon a person reporting the malicious UAV to authorities, perhaps with a photo of the UAV included as part of the report. It would be very useful to be able to automatically predict the manufacturer and specific product identification of the malicious UAV from the photo using a trained image classification model, even if the UAV is relatively far away from the person taking the photo.

The research problem this paper addresses is whether it is possible to train a deep learning image classification model to accurately classify images of UAVs in flight in

terms of the manufacturer and specific product identification. This is a difficult image classification problem because the UAVs are in flight and so may be quite distant from the camera, thereby appearing relatively small in an image. In addition, different types of UAVs appear visually similar, so it is not straightforward to distinguish between them.

In this paper, we discuss the acquisition of an image dataset of three popular UAVs in flight:

- the DJI Mavic 2 Enterprise.
- the DJI Mavic Air.
- the DJI Phantom 4.

The first two of these are visually similar, so much so that a person would be challenged to distinguish between them. Images are taken at various UAV elevations and distances using different zoom levels on the camera.

We then train various deep learning image classification models based upon Convolutional Neural Networks (CNNs) using the labelled image dataset. In this paper, we are most interested in Resnet-18 because it is relatively lightweight and has such a good reputation in a large variety of image classification tasks. We compare the performance of Resnet-18 against other popular and high performing models; these are AlexNet, VGG-16, and MobileNet v2 [5]. Rather than start the model training from the beginning, we employ the established technique of transfer learning, in which we start with a pre-trained model for a different image dataset and optimize the parameters for the UAV image dataset [6]. This reduces the time required to train the model and makes robust training possible with a smaller dataset.

Research into detecting and possibly identifying UAVs, both non-malicious and malicious, using machine learning techniques has been undertaken by several projects over the past decade as the prevalence of industrial and consumer UAVs has increased [7]. The raw data on which these machine learning techniques are trained can be based on audio, images, video, and radar signatures. In this section, we concentrate on the research related to image classification of malicious UAVs since this is the subject matter of the present paper.

In [8], the authors used a vision transformer (ViT) framework to model a dataset comprising 776 images of aeroplanes, helicopters, birds, non-malicious UAVs and malicious UAVs. The distinction between non-malicious and malicious UAVs was made primarily based upon whether the UAV was carrying a payload, which was assumed to be harmful and/or illegal. The model achieved an impressive accuracy of 98.3%. Our paper addresses a different problem, whereby a person reports a UAV as malicious based upon its location or behaviour rather than its visual characteristics or whether it is carrying a load, and the problem is to try to identify the type of UAV from the image provided in the report. However, our paper does not address UAVs carrying loads, and a future direction of our research will be to complement the image dataset with images of UAVs carrying loads.

In [9], the authors trained a You Only Look Once (YOLO) model to detect and track UAVs in video streams. They used the DJI Mavic Pro and DJI Phantom III for validation purposes and achieved a mean average precision (mAP) of 74.36%, which was superior to previous studies. However, they did not distinguish whether the UAV in the video stream was a DJI Mavic Pro or DJI Phantom III, identifying it only as a generic UAV. The objective of our research is different, to be able to distinguish between different UAV types (e.g., by manufacturer and specific model) based upon an image provided.

The study in [10] employed a dataset of 506 images and 217 audio samples to train and test a deep-learning model for the detection of UAVs based upon combined visual and audio characteristics. The best accuracy obtained was 98.5%. However, the aim was not to differentiate between different types of UAVs as in our research; rather, it was to distinguish between UAVs as a general class of object and other objects such as airplanes, birds, kites, and balloons. The combined video/audio approach was also adopted in [11],

but despite using different types of UAV such as a DJI Phantom 4 and DJI Mavic in the training and testing of the model, they did not attempt to distinguish the exact UAV type in the vicinity of the observer; instead, the objective was simply to detect that a UAV of some type was present.

The research presented in this paper is different from that reviewed above and novel in that it attempts to classify the manufacturer and specific product identification of a UAV in an image, with the UAV being at various elevations and distances from the observer, and with the observer using various zoom levels when taking the photo. The task is challenging both because we are attempting to distinguish between UAVs which may look similar, especially from a distance, and because the image may be taken at different distance and viewing angles with respect to the target UAV.

Before we captured our own dataset, we investigated whether any other public domain-specific datasets [12,13] existed in this area. In terms of other image and/or video datasets that could potentially be employed as part of this study, most public datasets are of images and/or videos taken by drones/UAVs rather than images and/or videos of drones/UAVs. One notable exception is a dataset of 1359 UAV images on Kaggle [14]. However, this dataset cannot be meaningfully used in our study for a variety of reasons (1) many of the images are not of UAVs in flight, rather they are images of people holding a UAV or the UAV on the ground, (2) there is no information on the manufacturer or model of the UAV in the metadata, (3) the dataset is not balanced in terms of an equal number of images of each UAV type, and (4) for those images representing UAVs in flight, there is no information on elevation or horizontal distance from the camera.

The novel contributions of the paper are as follows:

- Development of a methodology to capture a balanced and structured image dataset of different UAVs in flight. The dataset is balanced in terms of the number of images of each UAV and structured in that images are captured at specified elevations, horizontal distances and zoom levels. There is no similar (public) image dataset available.
- Training, testing and cross validation of various deep learning image classification models to be able to distinguish the manufacturer and specific product identification of a UAV in an image.
- An analysis of the ability of the image classification models to distinguish between UAVs which are extremely similar in visual appearance, and to classify UAVs which are distant from the observer. Specifically, the average testing accuracy of the trained Resnet-18 model on the dataset is greater than 80% even though two of the three UAVs, the DJI Mavic 2 Enterprise and the DJI Mavic Air, are very similar in appearance, and even though the UAVs may be at an elevation of 30 m and a horizontal distance of 30 m from the observer.

2. Materials and Methods

2.1. UAV Selection

It was decided to restrict the number of distinct UAVs employed in the current study to three to understand whether existing state-of-the-art deep-learning image classification models could distinguish between them even when the UAVs are quite far from the observer. The UAVs employed are illustrated in Figure 1.

DJI Mavic 2 Enterprise DJI Mavic Air DJI Phantom 4

Figure 1. UAVs Employed in Current Study.

Clearly, the DJI Mavic 2 Enterprise and DJI Mavic Air are visually similar, so we would intuitively expect an image classification model to struggle to distinguish between them. In contrast, the visual appearance of the DJI Phantom 4 is strikingly different in color and shape, so we would intuitively expect an image classification model to be able to easily distinguish between this UAV and the other two.

2.2. Image Capture

The methodology for capturing images of UAVs to form a trial dataset was designed to mirror how people are likely to take photos of malicious UAVs for reporting purposes in the field. Specifically, people are likely to use a smartphone for image capture, possibly with optical and/or digital zoom, and the target UAV may be at a different elevation h and a different horizontal distance from the person taking the photo d (see Figure 2).

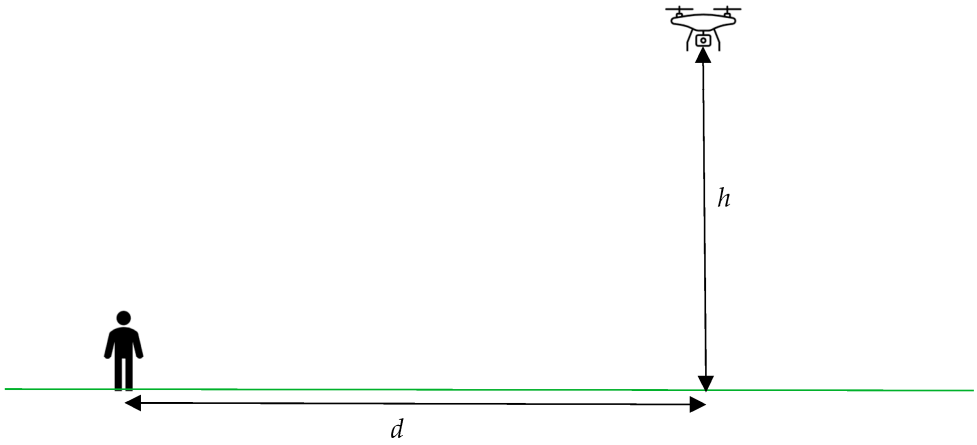

Figure 2. Distance d and Elevation h Parameters Used for Image Capture.

The images were all captured with an iPhone X which supports a 12 MP (3024 × 4032 pixels) autofocus camera with 2× optical zoom and 10× digital zoom. For each UAV position in terms of a distinct pair of h and d values, images were captured with 1×, 2×, 3×, 5× and 10× zoom. It should be stressed that a distinct photo was taken for each zoom level as opposed to a single image being taken and that image subsequently processed with different zoom levels.

Elevation values h of 5 m, 10 m, 15 m, 20 m, 25 m and 30 m were employed. These were measured from the UAV controller display.

Horizontal distances d of 0 m (i.e., observer directly below the UAV), 10 m, 20 m and 30 m were employed. These were measured with a standard measuring wheel.

With an image taken of each UAV for each of 5 camera zoom levels, 6 elevation levels and 4 horizontal distances, the dataset was expected to comprise 5 × 6 × 4 = 120 images of

each UAV. In fact, some additional images were captured for the largest values of h and d because a clear image could not always be captured in these cases. A total of 127 images were captured for each of the three UAVs. Therefore, the dataset is balanced in that there are the same number of images taken under similar circumstances for each of the three UAVs. Balanced datasets are preferred for machine learning to prevent a model being trained with a bias for one or more objects of interest.

It should be noted that there are some other advantages of this methodology for image capture other than simply mirroring what a typical person trying to report a malicious UAV in the field might do. Firstly, the use of different camera zoom levels and different values of h and d values changes:

- the size of the UAV in the image.
- the view angle at which the UAV is captured, thereby exposing different visual characteristics of the UAV.
- the background of the image.

This increases the diversity of the image dataset, which is known to be very important when the objective is to develop a robust image classification model which can make accurate predictions when exposed to new previously unseen images. For example, Figure 3 illustrates the effect of using different zoom levels (in successive shots) on the size of the UAV in the image and the image background.

Figure 3. Effect of Different Camera Zoom Levels On Size and Image Background (UAV: DJI Mavic Air, h = 5 m, d = 10 m, images reduced to 256 × 256 resolution).

Figure 4 illustrates the effect of different UAV elevations on the size of the UAV in the image, the view angle of the UAV in the image and the image background (note: these images were taken some time apart which explains why the backgrounds are different).

Figure 4. Effect of Different Elevations h on Size, View Angle, and Image Background (UAV: DJI Phantom 4, 10× Zoom, d = 10 m, images reduced to 256 × 256 resolution).

Another advantage of the image capture methodology in terms of formally indexing each image by zoom level, elevation h and horizontal distance d, is that it opens the possibility of not just image classification (i.e., predicting the type of UAV in the image), but also elevation prediction and distance prediction. This topic is not covered in this paper, primarily because it would require a much larger dataset, but it is an interesting possibility for the future, particularly as such predictions may facilitate evidence that a UAV was flying illegally (e.g., too close to people).

2.3. Model Theoretical Background

The deep-learning CNN image classification models employed as part of this study are compared and summarized in Table 1. These are all well-known and high performing image classification models. Some of them are part of families, e.g., there are Resnet-18, Resnet-34, Resnet-50 and Resnet-101 models. For this investigation, we generally employed the least complex model in a family (e.g., Resnet-18 in the case of Resnet) because it has the fewest number of parameters to train and therefore is less likely to be overfit when using a dataset that is not particularly large.

Table 1. Deep-learning CNN Image Classification Models Employed in This Study.

Model	Layers with Weights	Parameters	Input Image Resolution
AlexNet	8	~60 million	227 × 227
VGG-16	16	~135 million	224 × 224
Resnet-18	18	~11 million	224 × 224
MobileNet v2	53	~3.5 million	224 × 224

As discussed in the Introduction, we are primarily interested in Resnet-18 because it is relatively lightweight and has such a good reputation in a large variety of image classification tasks. Resnet-18 is a Convolutional Neural Network model which has 18 convolutional and/or fully connected layers in its architecture [15] as illustrated in Figure 5. To understand the structure of a typical convolution layer, consider the convolutional layer with designation "3 × 3, Conv, 128,/2". This uses 128 filters with window size 3 × 3 and a stride of 2. The curved arrows represent skip connections which provide some protection against overfitting.

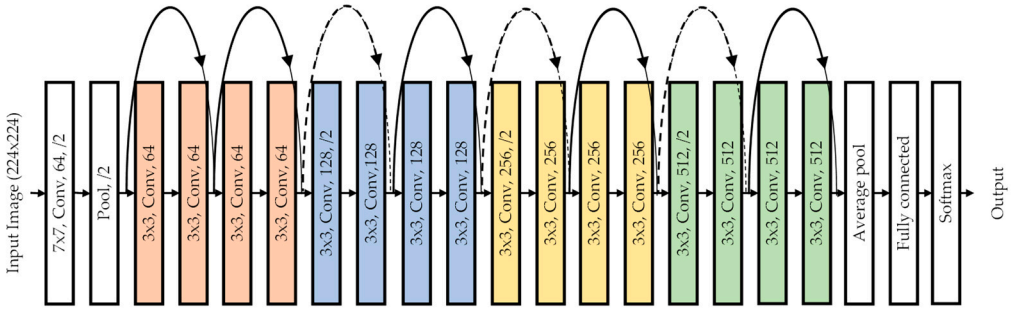

Figure 5. Structure of the Resnet-18 Model.

AlexNet was proposed by Alex Krizhevesky [16] in 2012 and is a deep and wide CNN model. This was considered as a significant step in the field of machine learning and computer vision for visual recognition and classification. The AlexNet architecture has 3 convolution layers and 2 fully connected layers. The recognition accuracy was found to be better against all traditional machine learning and computer vision approaches.

VGG-16 is a Convolutional Neural Network (CNN) model which was proposed by Karen Simonyan and Andrew Zisserman [17]. The use of uniform 3 × 3 filters is the standout feature of the VGG technique which reduces the number of weight parameters when compared to a 7 × 7 filters.

MobileNetv2 is a CNN that is based on an inverted residual structure whereby the residual connections are between the thin bottleneck layers [18]. As a source of non-linearity, the intermediate layer utilizes the lightweight depth wise convolutions to filter features. The architecture has a fully convolution layer with 32 filters and 19 residual bottleneck layers.

2.4. Image Pre-Processing and Management

As discussed previously, a total of 127 images were captured for each of the three UAVs. All images were down-sampled from 3024 × 4032 pixel resolution to 256 × 256 pixel resolution since this is a common intermediate resolution prior to training [5]. This also involved some cropping of the image, since the source raw image is rectangular (i.e., non-square) while the output processed image is square. The down-sampling/cropping was performed manually for all images to ensure the UAV was still in the frame of the output processed image.

The processed image data was then pseudo-randomly split into 101 training images and 26 test images for each UAV using the Python split-folders module [19] with a specific seed (2002). This corresponds to approximately an 80/20 split of train/test data, which is quite common when training image classification models.

2.5. Model Training and Validation

The 3 × 101 training images were then used to train each of the deep-learning CNN image classification models specified in Table 1. This number of training samples is not sufficient to train the model from an initial (random) state. Instead, the models were pre-trained with the ImageNet database, i.e., pre-loaded with weights corresponding to the result of training on ImageNet [20], then the last model layer was replaced so as to classify just three objects (i.e., the three UAVs use in this study), and the 3 × 101 training images were employed to further optimize all the weights to apply to the UAV images under consideration. This is known as *transfer learning* [6] and it is a standard technique employed in image classification for relatively small image datasets. As part of the training, we employed data augmentations of a random horizontal flip and a random resized crop to the model input image resolution (see Table 1). Such data augmentations are useful for generalizing the applicability of the trained model to new and previously unseen data.

Table 2 shows the hyperparameters used in the training for all models. The number of epochs (25) was sufficient to train the model to a converged final accuracy in all cases.

Table 2. Image Classification Hyperparameters Used for all Models.

Hyperparameter	Value
Number of epochs	25
Image batch size	40
Optimizer	Stochastic Gradient Descent (SGD)
Initial learning rate for optimizer	0.001
Momentum for optimizer	0.9
Learning rate decay factor	0.1
Learning rate decay period	7 epochs

When training is complete, the accuracy of the model in correctly predicting the UAV type in each image of the test set is given by:

$$Accuracy = \frac{n_{correct}}{n_{total}} \quad (1)$$

$n_{correct}$ is the number of test images for which the trained model correctly predicted the UAV type, and n_{total} is the total number of test images.

2.6. Cross Validation

The previous sections discuss model training and validation for one specific pseudo-random 80/20 training/test split of the processed image data. This is useful to obtain an initial idea about the relative accuracies of the different models, but ultimately the process should be repeated with multiple different training/test splits of the processed image data

to fully characterize the model performance and remove any bias that using just one specific training/test split may result in.

For this paper, we used repeated random sub-sampling cross validation (sometimes known as Monte Carlo cross-validation). The processed image data was partitioned 30 times into different 80/20 training/test splits using 30 different random seeds of the Python split-folders module [19]. Note that each such split was balanced in that it contained 101 training images and 26 test images for each UAV, i.e., there were the same number of training images for each UAV and the same number of test images for each UAV. Each model was trained across all 30 training/test splits, and the model accuracy figures averaged.

3. Results and Discussion

3.1. Overall Test Accuracy

Table 3 illustrates the test accuracy of the image classification models in decreasing order for the initial training/test split discussed in Section 2.4. The test accuracy is equal to the proportion of correct predictions (i.e., predicted UAV type = actual UAV type) the trained model makes on the test image dataset.

Table 3. Test Accuracy Using Trained Models.

Model	Number of Test Images (n_{total})	Number of Correct Predictions ($n_{correct}$)	Test Accuracy ($n_{correct}/n_{total}$)
VGG-16	3 × 26 = 78	67	0.859
Resnet-18	3 × 26 = 78	66	0.846
AlexNet	3 × 26 = 78	61	0.782
MobileNet v2	3 × 26 = 78	61	0.782

It is useful to compare the test accuracies represented in Table 3 with a baseline accuracy from a thought experiment. As previously discussed, of the three UAVs represented in the study, the DJI Mavic 2 Enterprise and DJI Mavic Air are visually similar, whereas the DJI Phantom 4 is quite different to the other two in appearance. If we assume a baseline trained model can always correctly predict a DJI Phantom 4 because of its difference in appearance, there will be 26 correct predictions from the test image dataset for this UAV alone. If we further assume that a baseline trained model cannot distinguish between the DJI Mavic 2 Enterprise and DJI Mavic Air, it will correctly predict each of these UAVs only 50% of the time on average, i.e., we will see 13 correct predictions out of 26 test images for the DJI Mavic 2 Enterprise, and 13 correct predictions out of 26 test images for the DJI Mavic Air. This results in a combined number of correct predictions of 26 + 13 + 13 = 52, and a baseline test accuracy of 52/78 = 0.667.

The test accuracies of all four models represented in Table 3 are significantly higher than the baseline reference accuracy of 0.667 from the simple thought experiment. From this perspective, the results are very encouraging. The two best performing models are VGG-16 and Resnet-18. Of the other two lower performing models, AlexNet has an older architecture, dating back to 2012, and MobileNet v2 employs significantly fewer trainable parameters than the other models (see Table 1), which might explain these results.

The test accuracies represented in Table 3 are contextual, i.e., specific to the current study and its design. It is very likely that the actual accuracy values will change significantly if the design of the study is amended. For example, introducing more UAV types will likely result in lower test accuracies since it is more challenging to distinguish between a larger number of object classes. Another example would be that increasing the distance between the observer and UAV when an image is captured will likely result in lower test accuracies since it is more challenging to accurately pick out a smaller object from an image than a larger one.

3.2. Error Analysis

It is important to understand under what circumstances the trained models are making errors in their predictions. In particular, it is useful to know whether the assumption that the models will struggle to distinguish between the DJI Mavic 2 Enterprise and DJI Mavic Air is valid. This can be achieved by examining the confusion matrix for each model, which shows a count of predicted object versus true actual object. Table 4 illustrates the confusion matrix for one of the two best performing models in our study, Resnet-18, and Table 5 illustrates the confusion matrix for one of the other two models in our study, MobileNet v2. The counts value in the main diagonal of the matrix (i.e., from top left to bottom right) represent correct predictions (i.e., predicted UAV type = actual UAV type), whereas the count values in other cells represent an error in prediction.

Table 4. Confusion Matrix for Prediction of Test Images Using Trained Resnet-18 Model.

		Predicted UAV		
	Number of Validation Images = 3 × 26 = 78	**DJI Mavic 2 Enterprise**	**DJI Mavic Air**	**DJI Phantom 4**
Actual UAV	DJI Mavic 2 Enterprise	22	4	0
	DJI Mavic Air	6	20	0
	DJI Phantom 4	2	0	24

Table 5. Confusion Matrix for Prediction of Test Images Using Trained MobileNet v2 Model.

		Predicted UAV		
	Number of Validation Images = 3 × 26 = 78	**DJI Mavic 2 Enterprise**	**DJI Mavic Air**	**DJI Phantom 4**
Actual UAV	DJI Mavic 2 Enterprise	19	6	1
	DJI Mavic Air	6	18	2
	DJI Phantom 4	1	1	24

It can be seen from Table 4, and more so from Table 5, that the majority of prediction errors are one of the two following cases:

- actual UAV is a DJI Mavic 2 Enterprise, predicted UAV is a DJI Mavic Air (row 1, column 2).
- actual UAV is a DJI Mavic Air, predicted UAV is a DJI Mavic 2 Enterprise (row 2, column 1).

Furthermore, when the actual UAV in the image is a DJI Phantom 4, it is almost always predicted correctly by the model (24 out of 26 times in both Tables 4 and 5).

These statistics confirm the previous intuition that the trained models are very good at correctly predicting a DJI Phantom 4 in an image due to its distinctive appearance, but sometimes get confused in distinguishing between the DJI Mavic 2 Enterprise and DJI Mavic Air due to their similar appearance. However, the trained models still correctly predict a DJI Mavic 2 Enterprise and DJI Mavic Air in images more often than they make errors.

3.3. Per UAV Type Metrics

Figures 5 and 6 show the per-UAV type metrics of precision, recall and the F1-score for the Resnet-18 and MobileNet v2 models, respectively.

Precision measures the proportion of predictions for a particular UAV which are correct. For example, taking the confusion matrix for the Resnet-18 model in Table 4, the counts in the 2nd column show that the precision for the DJI Mavic Air is 20/(4 + 20 + 0) = 20/24 = 0.833.

Recall measures the proportions of actual (i.e., true) instances for a particular UAV which are predicted correctly. For example, taking the confusion matrix for the Resnet-18 model in Table 4, the counts in the 2nd row show that the recall for the DJI Mavic Air is 20/(6 + 20 + 0) = 20/26 = 0.769.

The F1-score is a summary statistic that is equal to the geometric mean of the precision and recall.

Figures 6 and 7 confirm the findings of the previous section in that the metrics for the DJI Phantom 4 are significantly better than those for the DJI Mavic 2 Enterprise and DJI Mavic Air, because the DJI Phantom 4 is more visually distinct.

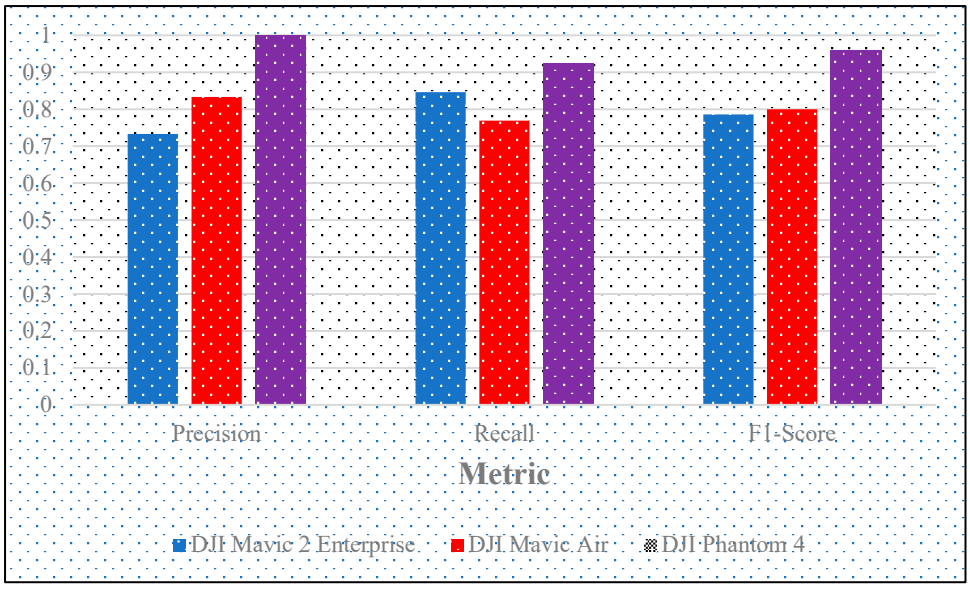

Figure 6. Per Class Metrics for Prediction of Test Images Using Trained Resnet-18 Model.

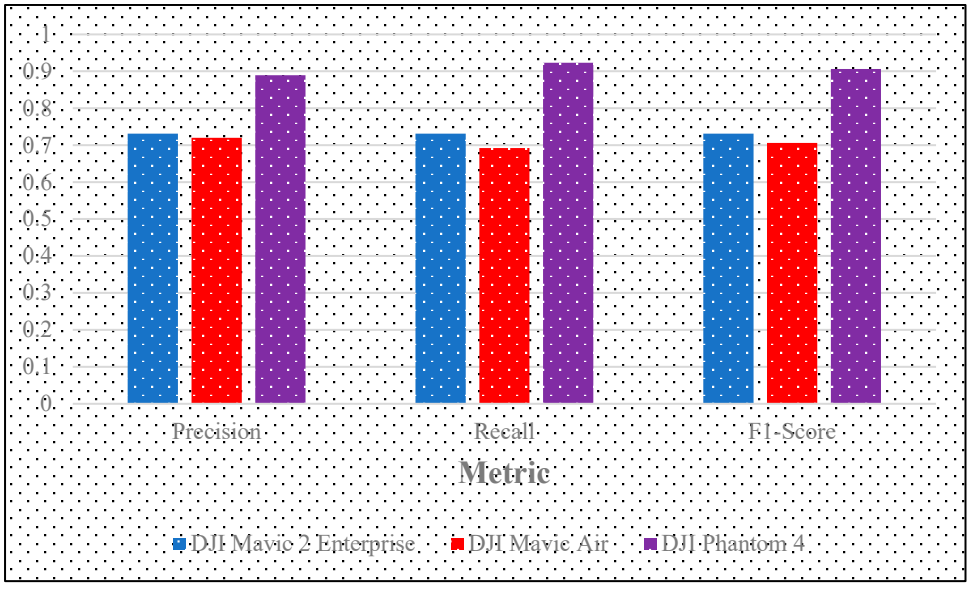

Figure 7. Per Class Metrics for Prediction of Test Images Using Trained MobileNet v2 Model.

3.4. Cross Validation

Table 6 illustrates the model test accuracy results of the cross validation on the 30 pseudo-randomly training/test splits. When multiple training/test splits are considered, Resnet-18 offers the best mean and median model accuracy, followed by VGG-16, MobileNet v2 and AlexNet in order.

It is interesting to note that some specific data splits result in consistently good or poor accuracy across all models. For example, consider data split index 15, for which the accuracy of all 4 models is poor. On examination of this dataset, it transpires that a larger proportion than usual of the images in the test set were of distant shots of UAVs, for which is much more difficult to make a correct prediction about the type of UAV.

Table 6. Results of Cross Validation Showing Mean and Median Test Accuracy.

Data Split Index	Seed	Test Accuracy			
		VGG-16	**Resnet-18**	**AlexNet**	**MobileNet v2**
1	3	0.795	0.846	0.705	0.756
2	42	0.808	0.769	0.821	0.782
3	75	0.859	0.821	0.795	0.821
4	126	0.808	0.795	0.769	0.821
5	527	0.782	0.744	0.667	0.756
6	603	0.769	0.782	0.667	0.782
7	820	0.782	0.808	0.808	0.846
8	991	0.782	0.782	0.692	0.718
9	2002	0.859	0.846	0.782	0.782
10	4565	0.744	0.846	0.782	0.744
11	6425	0.833	0.872	0.795	0.833
12	9476	0.795	0.846	0.756	0.744
13	13,869	0.821	0.808	0.782	0.833
14	24,658	0.744	0.833	0.808	0.795
15	44,941	0.718	0.667	0.692	0.641
16	59,740	0.718	0.795	0.782	0.769
17	61,086	0.833	0.833	0.782	0.756
18	73,952	0.833	0.821	0.756	0.808
19	84,248	0.821	0.821	0.705	0.756
20	91,033	0.821	0.897	0.821	0.769
21	163,576	0.795	0.769	0.718	0.808
22	356352	0.795	0.795	0.718	0.859
23	406,538	0.795	0.846	0.679	0.769
24	459,208	0.897	0.833	0.795	0.833
25	565,642	0.769	0.808	0.795	0.744
26	755,484	0.846	0.795	0.769	0.769
27	887,546	0.885	0.897	0.769	0.833
28	943,457	0.705	0.756	0.744	0.718
29	1,418,519	0.821	0.795	0.795	0.782
30	5,641,860	0.718	0.795	0.821	0.718
	Mean	**0.798**	**0.811**	**0.759**	**0.778**
	Median	**0.795**	**0.808**	**0.776**	**0.776**

3.5. Training Time Analysis

Table 7 illustrates the mean and median training times for the various models when using the hyperparameters specified in Table 2. The model training was conducted with

PyTorch scripts running on Google Colab with Graphics Processing Unit (GPU) acceleration. As Google Colab is a shared cloud-based service, service levels can change over time and so the values represented in Table 7 are subject to such variations. However, it can be clearly seen that VGG-16 is the slowest model to train, while AlexNet and MobileNet v2 are the quickest to train.

Table 7. Training Times for Various Models.

	VGG-16	Resnet-18	AlexNet	MobileNet v2
Mean	184.1 s	113.7 s	68.0 s	78.5 s
Median	184.0 s	120.0 s	67.0 s	78.5 s

4. Conclusions and Further Work

In this paper, we have described the collection of an image dataset for 3 popular UAVs at different elevations, different distances from the observer, and using different camera zoom levels. This UAV image dataset has been modelled using four CNN image classification algorithms, comprising AlexNet, VGG-16, Resnet-18 and MobileNet v2. The accuracy of the trained models on previously unseen test images is up to approximately 81% (for Resnet-18). This is encouraging given that two of the UAVs, the DJI Mavic 2 Enterprise and the DJI Mavic Air, are visually similar, and given that some of the photos of UAVs were taken at relatively large distances from the observer. The main anticipated application of this work is the automatic identification of the manufacturer and specific product identification of a UAV contained in an image which is part of a malicious UAV report. However, it could equally be employed in real time by security cameras (e.g., on buildings or other infrastructure) which identify an unwanted or even illegal UAV in the vicinity.

The main limitations of this image classification based technique for UAV identification are (1) it can be difficult to distinguish between UAVs which are of similar appearance and/or have similar flight characteristics, and (2) UAVs which are far from the observer/camera will appear small in the image, thus complicating identification via image classification. Therefore, it may be useful to combine this technique with other methods of UAV identification (such as radar or acoustic signature), although this may not always be feasible depending upon the scenario.

Given the encouraging results to date, we plan to expand the image dataset to include more UAV types, and more UAV usage scenarios, e.g., UAVs carrying loads, UAVs in motion and UAVs that are part of swarms. In addition, all the images collected to date were taken from below the UAV, because the main application for the work is an observer manually observing a (malicious) UAV from the ground. However, there is also the possibility that the images can be captured from above the UAV, e.g., by security cameras on tall buildings or even by another UAV. Therefore, we also plan to take photos of UAVs from above. The set of candidate image classification models may also be expanded; given the anticipated increased size of the image database, more complex models such as Resnet-34 may be considered. Finally, we would also like to expand the work to include object detection of UAV type in video streams using a YOLO variant as the object detection algorithm.

Author Contributions: Conceptualization, J.B., Z.G. and N.R.; methodology, J.B. and Z.G.; software, J.B.; validation, J.B., Z.G. and N.R.; data curation, J.B. and Z.G.; writing—original draft preparation, J.B.; writing—review and editing, J.B., Z.G. and N.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to ongoing data collection and management.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Maghazei, O.; Lewis, M.A.; Netland, T.H. Emerging technologies and the use case: A multi-year study of drone adoption. *J. Oper. Manag.* **2022**, *68*, 560–591. https://doi.org/10.1002/joom.1196.
2. Vattapparamban, E.; Güvenç, I.; Yurekli, A.I.; Akkaya, K.; Uluağaç, S. Drones for smart cities: Issues in cybersecurity, privacy, and public safety. In Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC) , Paphos, Cyprus, 5–9 September 2016; pp. 216–221. https://doi.org/10.1109/IWCMC.2016.7577060.
3. McTegg, S.J.; Tarsha Kurdi, F.; Simmons, S.; Gharineiat, Z. Comparative Approach of Unmanned Aerial Vehicle Restrictions in Controlled Airspaces. *Remote Sens.* **2022**, *14*, 822. https://doi.org/10.3390/rs14040822.
4. Yaacoub, J.P.; Noura, H.; Salman, O.; Chehab, A. Security analysis of drones systems: Attacks, limitations, and recommendations. *Internet Things* **2020**, *11*, 100218. https://doi.org/10.1016/j.iot.2020.100218.
5. Wang, W.; Yang, Y.; Wang, X.; Wang, W.; Li, J. Development of convolutional neural network and its application in image classification: A survey. *Opt. Eng.* **2019**, *58*, 040901. https://doi.org/10.1117/1.OE.58.4.040901.
6. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proc. IEEE* **2020**, *109*, 43–76. https://doi.org/10.1109/JPROC.2020.3004555.
7. Taha, B.; Shoufan, A. Machine learning-based drone detection and classification: State-of-the-art in research. *IEEE access*. **2019**, *7*, 138669–138682. https://doi.org/10.1109/ACCESS.2019.2942944.
8. Jamil, S.; Abbas, M.S.; Roy, A.M. Distinguishing Malicious Drones Using Vision Transformer. *AI* **2022**, *3*, 260–273. https://doi.org/10.3390/ai3020016.
9. Singha, S.; Aydin, B. Automated Drone Detection Using YOLOv4. *Drones* **2021**, *5*, 95. https://doi.org/10.3390/drones5030095.
10. Jamil, S.; Rahman, M.; Ullah, A.; Badnava, S.; Forsat, M.; Mirjavadi, S.S. Malicious UAV detection using integrated audio and visual features for public safety applications. *Sensors* **2020**, *20*, 3923. https://doi.org/10.3390/s20143923.
11. Liu, H.; Wei, Z.; Chen, Y.; Pan, J.; Lin, L.; Ren, Y. Drone detection based on an audio-assisted camera array. In Proceedings of the IEEE Third International Conference on Multimedia Big Data (BigMM), Laguna Hills, CA, USA, 19 April 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 402–406. https://doi.org/10.1109/BigMM.2017.57.
12. Hasan, M.J.; Islam, M.M.; Kim, J.M. Acoustic spectral imaging and transfer learning for reliable bearing fault diagnosis under variable speed conditions. *Measurement* **2019**, *138*, 620–631. https://doi.org/10.1016/j.measurement.2019.02.075.
13. Zabin, M.; Choi, H.J.; Uddin, J. Hybrid deep transfer learning architecture for industrial fault diagnosis using Hilbert transform and DCNN–LSTM. *J. Supercomput.* **2022**, 1–20. https://doi.org/10.1007/s11227-022-04830-8.
14. Özel, M. Drone Dataset (UAV). 2020. Available online: https://www.kaggle.com/datasets/dasmehdixtr/drone-dataset-uav (accessed on 3 December 2022).
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
16. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. https://doi.org/10.1145/3065386.
17. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556. https://doi.org/10.48550/arXiv.1409.1556.
18. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. https://doi.org/10.48550/arXiv.1801.04381.
19. Python Package Index. Split-Folders 0.5.1. 2022. Available online: https://pypi.org/project/split-folders/ (accessed on 29 October 2022).
20. ImageNet. 2022. Available online: https://www.image-net.org/ (accessed on 29 October 2022).