System To Distinguish Birds And Drones

Dheeraj Reddy Pailla, Kshitij Gupta

International Institute of Information Technology, Hyderabad

Abstract

While image classification is a nearly solved problem (if not already solved), many of today's state-of-the-art methods use highly sophisticated and obscure deep neural networks. While these deep neural networks are highly effective, they consume a lot of processing power as compared to classical machine learning models. In this project, we take a comprehensive look at several methods to distinguish between birds and drones using classical machine learning models along with some popular feature descriptors. We propose a pipeline to accurately distinguish between birds and drones while trying to keep up with the current state-of-the-art accuracy levels.

Introduction

Problem Statement

The goal of the project is to build an image classifier that can successfully distinguish between a drone and a bird with sufficient accuracy using only image processing and machine learning techniques. Deep learning networks are used only for a comparative study.

Motivation

The prime motivation to pursue this project is for educational purposes. Using classical machine learning models, we learn to use image processing features and descriptors to build a classifier.

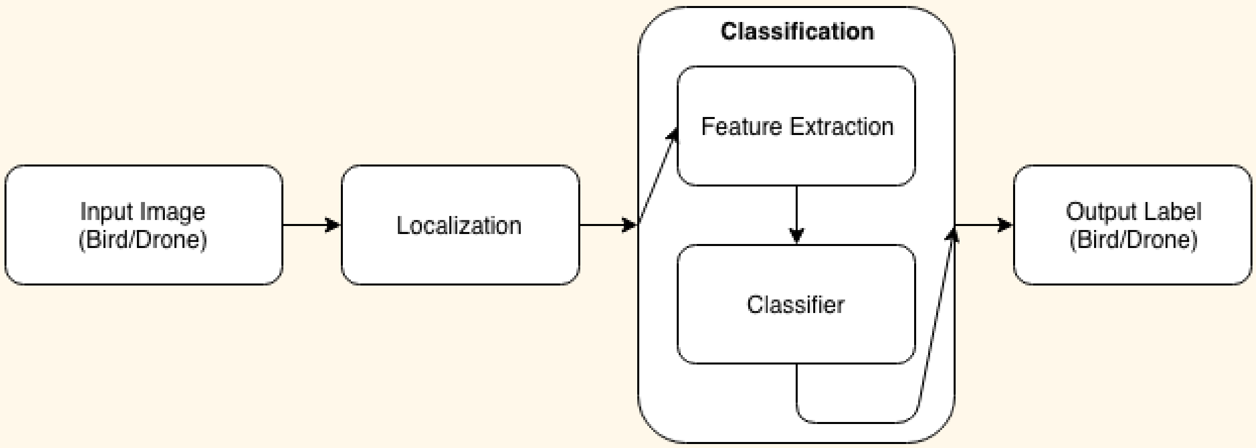
Since deep learning networks usually require huge computation power, another motivation for this project is also to reach high efficiency whilst requiring as little computation power as possible.

The applications for such a classifier include national security. The use of drones, especially military grade, are on the rise, with several of them being designed to look like birds in order to evade security checks. The system proposed in this project can be used in conjunction with existing surveillance systems to improve security.

Overview

In this project, we take a look at SIFT, SURF, and ORB features in conjunction with classical machine learning models such as Logistic Regression, Linear Discriminator Analysis, K

Neighbours Classifier, Decision Tree, Random Forest, Gaussian Naive Bayes and Support Vector Machine. We perform a comparative and exploratory analysis to find which combination of features/descriptors and classifier model perform the best in this particular use case. Before we pass an image through our classifier, we localize the image to our area of interest in order to maximise efficiency of the classifier.

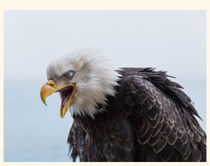
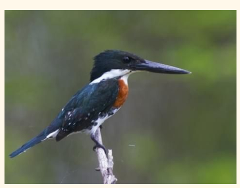
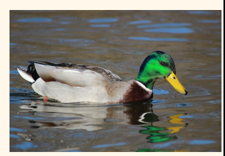
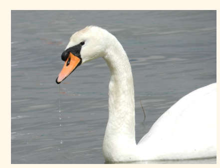


Dataset

About

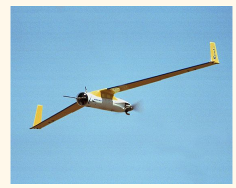
The dataset consists of 10 species of birds, each with approximately 100 images each. Similarly, there are 10 models of drones with approximately 100 images each. Of the birds, 7 are predominantly aerial, and 3 are aquatic species. Of the drones, 8 models are consumer grade drones, and 2 are military grade drones, currently deployed by the US military.

Birds:-

				
Owl	Pigeon	Dove	Falcon	Parrot
				
Eagle	Kingfisher	Goose	Duck	Swan

Goose, Duck and Swan are aquatic birds.

Drones:-

				
DJI Phantom 3	DJI T600 Inspire	DJI Phantom 2 Vision+ V3.0	DJI Phantom 3 Advanced	DJI Phantom 2 Vision+
				
Parrot PF725100 BeBop	Heli-Max 1SQ	DJI Phantom Aerial	Lockheed Martin Stalker	Boeing Insitu ScanEagle

Lockheed Martin Stalker and Boeing Insitu ScanEagle are military grade drones, operated by the US military.

Data Collection

The data collection process for the birds was a relatively straightforward process, with readymade datasets available online such as CUB-200 and ImageNet. However, data collection for drones was a much more difficult task. There currently is no standardised and widely used dataset available for drones. Hence, we decided to build our own dataset. The dataset of drones represented above consist of the 8 most popular consumer level drones of 2017, and 2 military grade drones.

However, the problems with data collection didn't end there. While there are several images of drones in a white background or in a box, It was clear that there are very few images of drones in action. For whatever model we would build, it was imperative that we had several images of drones flying rather than just sitting idle. Without these, the classifier would not be useful for any practical purposes.

Typical Image Found   Ideal Image

We were unable to reach our goal of finding 1000 images of drones due to this constraint. So we took whatever pictures we could find, and took a mirror image of it in order to make up for lost numbers. By the end of our data collection phase, we ended up with around 800 images of drones and 1200 images of birds, for a total of around 2000 images.

Sources

For both birds and drones, images were eventually collected from Google Image Search results. A Google Chrome plugin was used in order to scrape images of each search result's web page.

Related Work

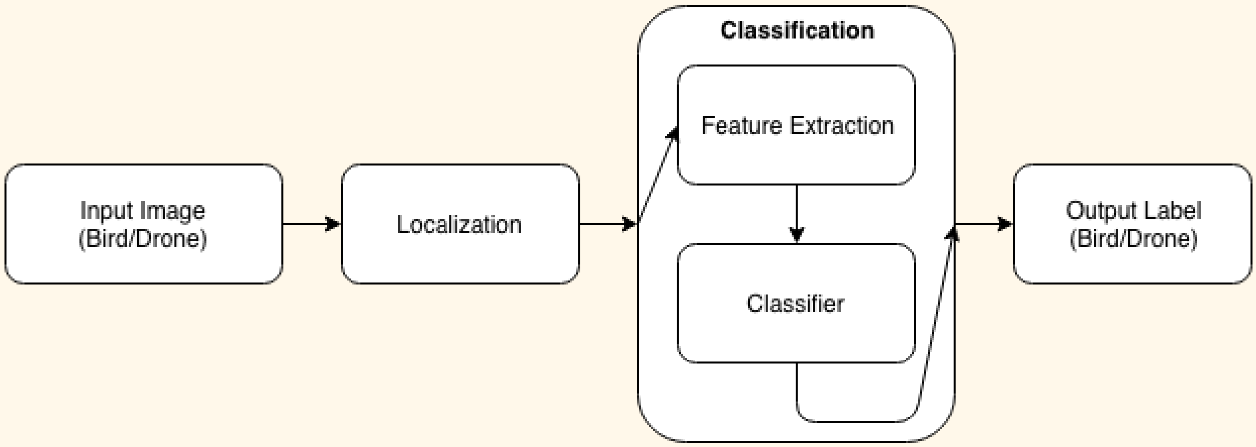
As stated earlier, the current state-of-the-art image classifiers involve deep learning networks. Residual Networks (also known as ResNets) based on the 2016 paper "Deep Residual Learning for Image Recognition" are widely used for image classification. Despite the high number of layers, ResNets still have lesser complexity than the previous state-of-the-art - VGG network. For devices with limited processing power, MobileNets based on the 2017 paper "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications" are widely used.

However, we will be using deep neural networks only for comparative purposes in this project. The main inspiration for this project comes from the 2007 paper "Evaluating bag-of-visual-words representations in scene classification", which uses a bag-of-visual-words representation for any descriptor to classify images.

A popular implementation of this can be found in VLFeat's Matlab source code here. There is also a Python implementation of the same source code, which can be found here. However, both of these implementations are over 6 years old, with the Python version now obsolete and largely non functional.

Method

As mentioned earlier, our proposed pipeline involves localization of the image to locate the area of interest, followed by feature extraction and then classification.



We use the following feature descriptors - SIFT, SURF, ORB

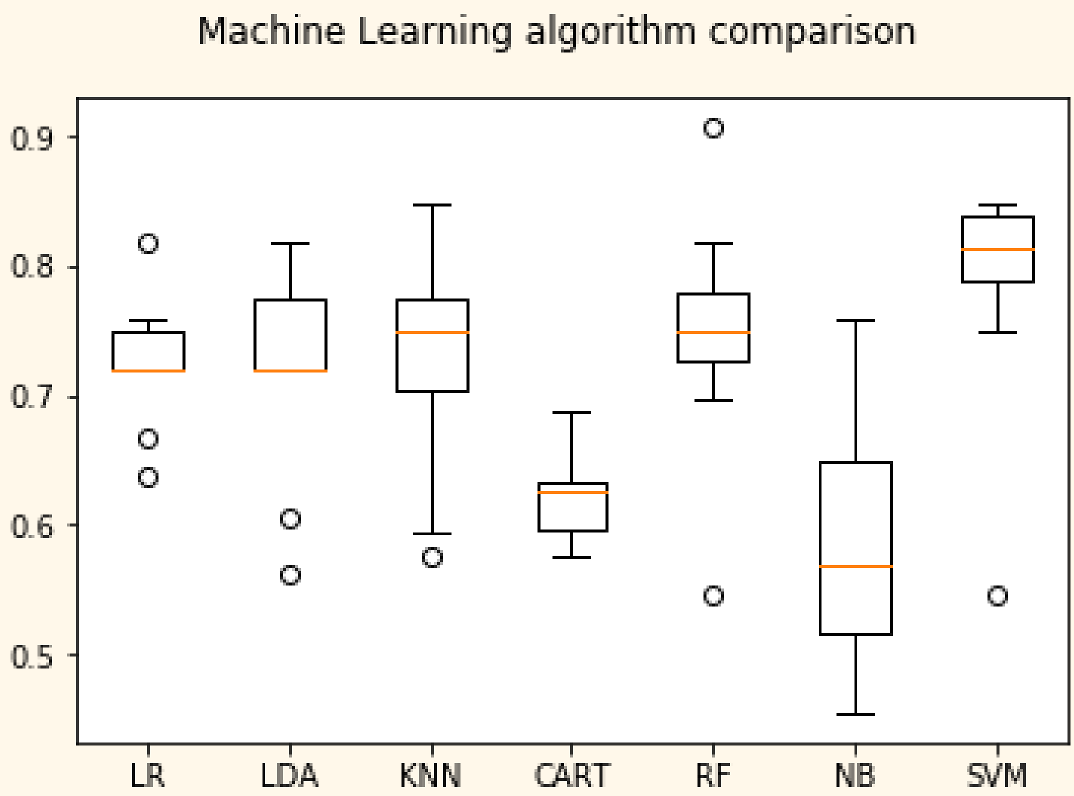
We use the following models - Logistic Regression, Linear Discriminator Analysis, K Neighbours Classifier, Decision Tree, Random Forest, Gaussian Naive Bayes and Support Vector Machine.

We perform 10-fold cross validation for each classifier for each classifier model for each feature descriptor. The dataset is split 40% for training and 60% for testing. The training dataset is split once again in order to generate the validation set.

Results

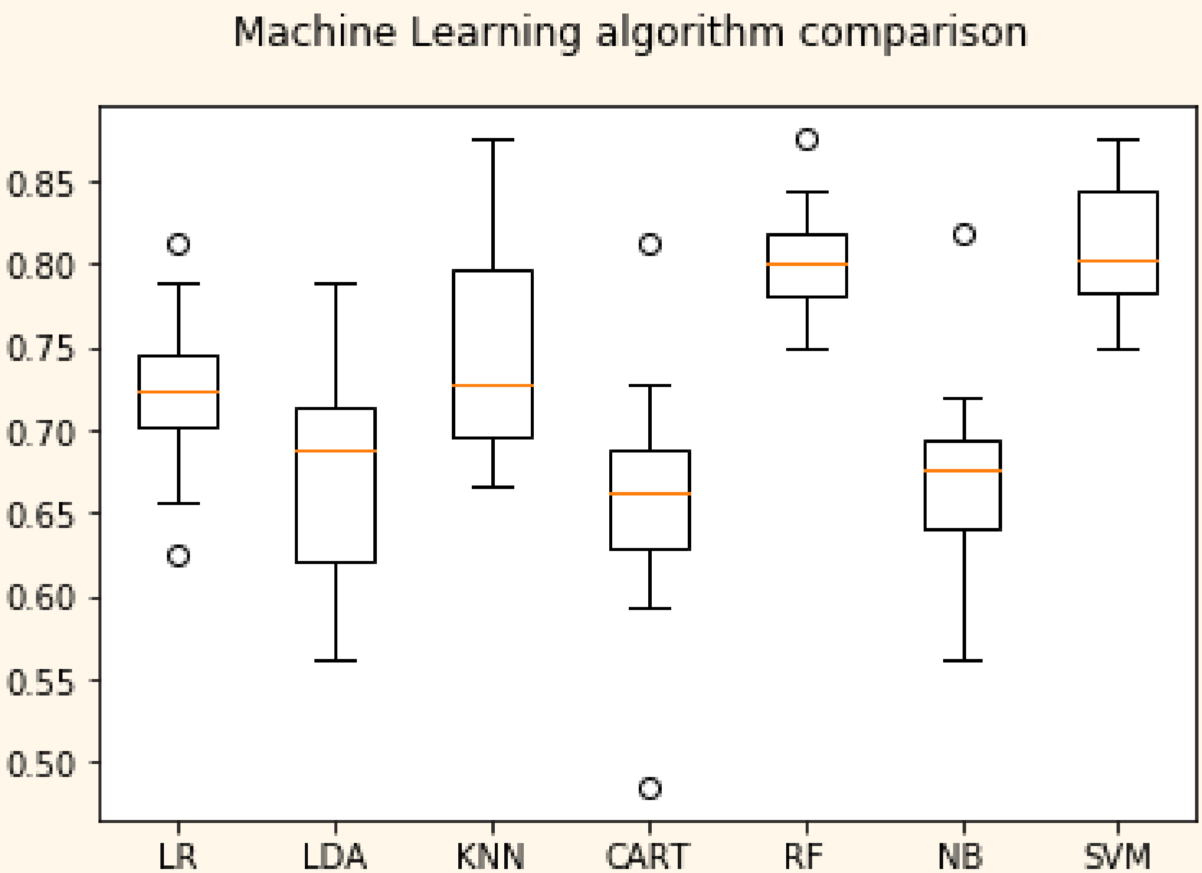
SIFT

Model	Mean Accuracy	Standard Deviation
Logistic Regression	0.725379	0.047337
Linear Discriminant Analysis	0.718845	0.075625
K Neighbour Classifier	0.728504	0.082814
Decision Tree	0.620644	0.033177
Random Forest	0.747348	0.087209
Naive Bayes	0.589773	0.094600
Support Vector Machine	0.787311	0.085703



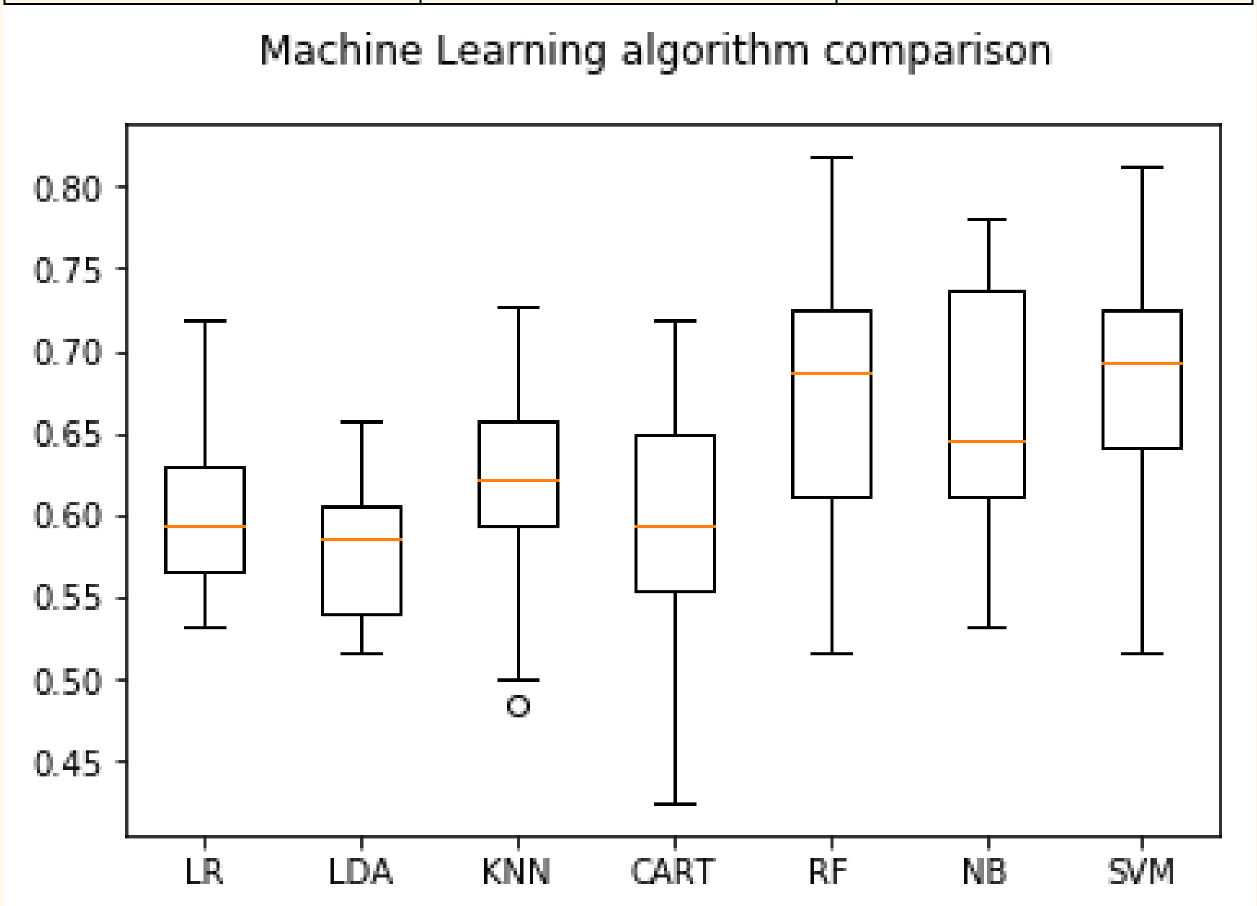
SURF

Model	Mean Accuracy	Standard Deviation
Logistic Regression	0.722064	0.052678
Linear Discriminant Analysis	0.675758	0.067194
K Neighbour Classifier	0.747064	0.063922
Decision Tree	0.657860	0.082106
Random Forest	0.802557	0.036662
Naive Bayes	0.675568	0.063464
Support Vector Machine	0.811742	0.037753



ORB

Model	Mean Accuracy	Standard Deviation
Logistic Regression	0.598580	0.053186
Linear Discriminant Analysis	0.580303	0.043207
K Neighbour Classifier	0.611080	0.070114
Decision Tree	0.596117	0.081487
Random Forest	0.672917	0.083933
Naive Bayes	0.663352	0.077425
Support Vector Machine	0.682197	0.080810



From the above results, it is clear that SVMs clearly outperform all the other classifiers across all the feature descriptors. Therefore, we build and optimise an SVM for this project. For the purpose of comparison, we also build a K Neighbour Classifier for analysis. A grid search was done for hyperparameter tuning.

SVM

<table>
<tr><th>Feature Descriptor</th><th>Accuracy</th><th>Confusion Matrix</th></tr>
<tr><td>SIFT</td><td>0.726038338658</td><td><table><tr><th></th><th>Drone</th><th>Bird</th></tr><tr><th>Drone</th><td>318</td><td>108</td></tr><tr><th>Bird</th><td>235</td><td>591</td></tr></table></td></tr>
<tr><td>SURF</td><td>0.75</td><td><table><tr><th></th><th>Drone</th><th>Bird</th></tr><tr><th>Drone</th><td>327</td><td>99</td></tr><tr><th>Bird</th><td>214</td><td>612</td></tr></table></td></tr>
<tr><td>ORB</td><td>0.575878594249</td><td><table><tr><th></th><th>Drone</th><th>Bird</th></tr><tr><th>Drone</th><td>366</td><td>60</td></tr><tr><th>Bird</th><td>471</td><td>355</td></tr></table></td></tr>
</table>

KNN

<table>
<tr><th>Feature Descriptor</th><th>Accuracy</th><th>Confusion Matrix</th></tr>
<tr><td>SIFT</td><td>0.803514376997</td><td><table><tr><th></th><th>Drone</th><th>Bird</th></tr><tr><th>Drone</th><td>222</td><td>204</td></tr><tr><th>Bird</th><td>42</td><td>784</td></tr></table></td></tr>
<tr><td>SURF</td><td>0.784345047923</td><td><table><tr><th></th><th>Drone</th><th>Bird</th></tr><tr><th>Drone</th><td>303</td><td>123</td></tr><tr><th>Bird</th><td>147</td><td>679</td></tr></table></td></tr>
<tr><td>ORB</td><td>0.575079872204</td><td><table><tr><th></th><th>Drone</th><th>Bird</th></tr><tr><th>Drone</th><td>317</td><td>109</td></tr><tr><th>Bird</th><td>423</td><td>403</td></tr></table></td></tr>
</table>

Successes

For both kNN and SVM classifiers, it was clear that most images were getting classified accurately (approximately 80%). The optimal feature descriptor to classify the images was SURF. It is clear that birds are getting classified accurately.

Failures

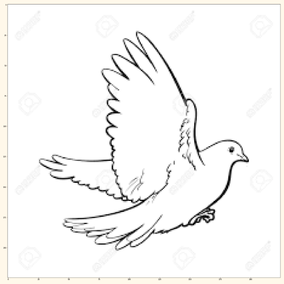
Despite the decent accuracy, some images were inaccurately classified for both classes, with the drones class suffering the most. This is down to the inefficiency of the localization module. Some images that were taken in the dark (such as drones with lights), could not be identified accurately. The dataset is also too sparse, so we didn't have enough images of some types of drones(night shots, flying). Flying ones usually always classified as birds.

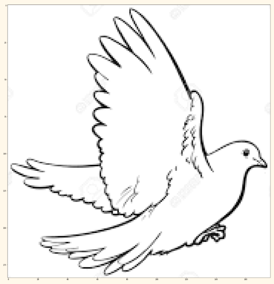
Manual Feature Extraction

We observed that most drones have flat surfaces and straight edges in their images. So, we decided to detect all the straight edges in an image and use them as features.

To detect straight edges, We did the following:

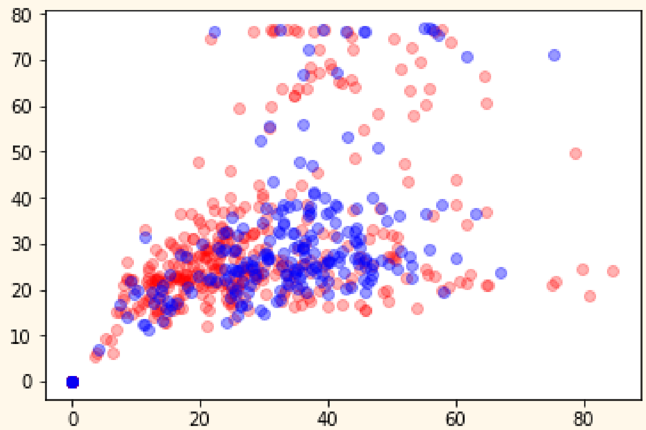
1. **Detect edges:** We used Gaussian filter followed by a Prewitt filter to detect edges and then we used thresholded the edge image to get a binary image required for running morphology operators.
2. **Filtering edges:** To further clean the edge space, We used closing to connect nearby components. Then we ran thinning to get edges of uniform thickness which can be used by Hough Transform. To allow rough but straight edges, We dilated the edge space with a 2x2 structuring element.
3. **Localisation:** We use the filtered edges for localisation, by stripping from all sides. We get the following results:



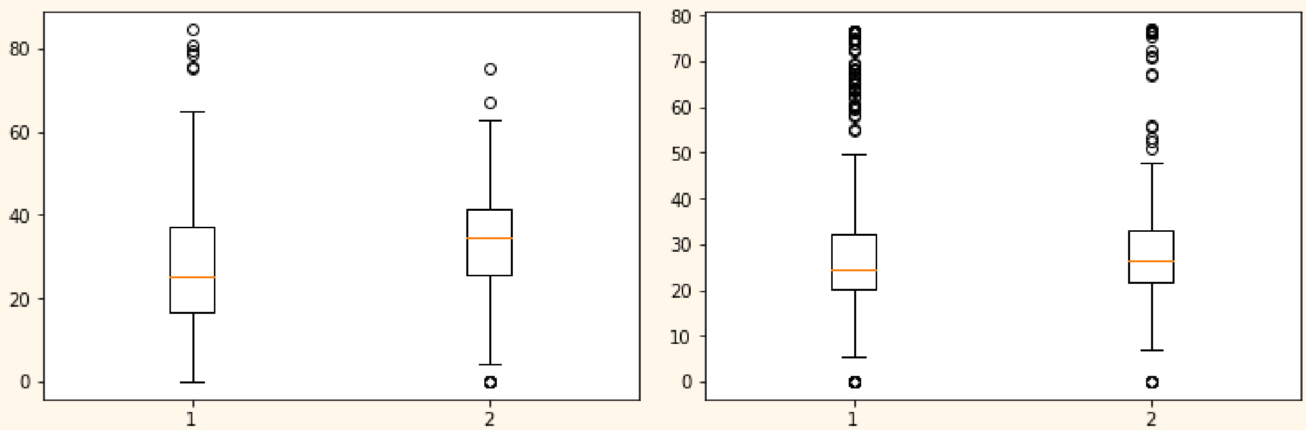


4. **Detecting Straight Edges:** We used Probabilistic Hough Transform to get the straight lines. We used the length of the longest line and sum of all edges as two parameters for classification.

5. **Plot Feature Space:** We visualised the feature space to check if they form separable clusters.



Feature Space (Feature 1 vs Feature 2)



Box Plots of Feature 1 and Feature 2

6. **Classification**: We used 9-Nearest Neighbour classifier to predict the class of a test image.

Results: We were able to get an accuracy of *62.24%* with 85%-15% Training-Test split.

Conclusion: We observed that there were many bird outliers which resulted because image boundaries.

Comparative Analysis

The major advantage of using classical machine learning models with image processing features is that decent accuracy can be achieved even with a limited dataset. It is quite clear that in this particular problem statement, data collection turned out to be a huge stumbling block. Since deep learning frameworks usually require large amount of data during its training phase, it makes much more sense to use an SVM or a kNN as shown in this project.

For a fair comparison in terms of processing power, we chose to train a MobileNet 0.50 classifier on the same dataset. As expected, the classifier gives a much higher accuracy than all of our classifiers - 96.4%. It was clear that the MobileNet was able to learn features for the drone dataset much better than all of our classifiers.

For future improvements to our classifier, it is imperative to increase the quantity and diversity of the drones dataset. Once this has been done, we can also experiment with mixing and matching various feature descriptors to see if accuracy can be improved.

Task Division

- Dheeraj Reddy Pailla
    - Drone dataset collection, cleaning
    - SIFT, SURF and ORB feature descriptors
    - SVM Classifier
- Kshitij Gupta
    - Bird dataset collection, cleaning
    - kNN Classifier
    - CNNs and state-of-the-art comparison