## Background:

The Capstone project is A military aircraft detection dataset. This is my understanding of what is included in this project. There are 43 different files of aircraft, each file is for 1 type of aircraft. The amount of pictures per file vaires per file. The smallest folder has 120 images and the largest has 1163 images. The data set is publically available with the following Kaggle link:

<https://www.kaggle.com/datasets/a2015003713/militaryaircraftdetectiondataset?select=crop>

## Questions:

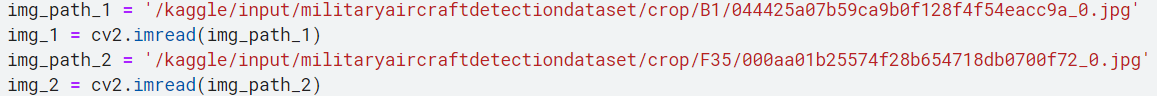
1. Can I display 2 images side by side?
2. Can I modify an image to set the colour of the images to a grey scale?
3. Can I modify an image to flip it vertically?
4. Can I modify an image to rotate it 90 degrees?
5. Can I modify an image to blur it?

## Stage 1 Exploratory Data Analysis for Visual Data:

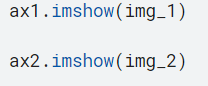
Using a combination of matplot, skimage, sv2 and numpy we can manipulate images of the data to do what we require for our visualisation tests.

### Question 1:

For this test I used 2 images of a B1 plane and a F35 plane, both images were imported from the Kaggle library and read using these lines of code.

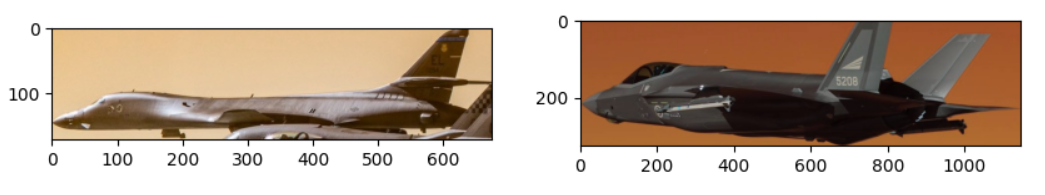


The images were then displayed with the following code.



The result from this test is as follows.

\



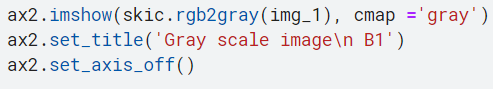
### Question 2:

For this test I used the same 2 images of the B1 and the F35 from the Kaggle library read and displayed with the same method.

The purpose of this test is to display the images in a grayscale. First we need to add some rows to our testing area to display the grey scaled images. The following code allows us to extend the image grid.



To grey scale the images we run a command when displaying the image



The cmap command allows us to set the colour of the image. The set\_title command allows me to add a title to differentiate the images. The result of the test is as follows.



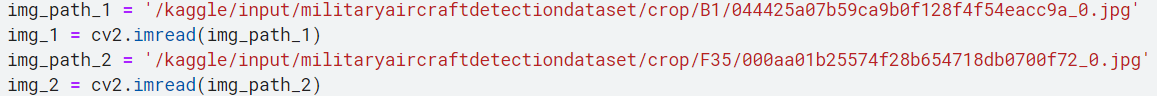
### Question 3:

This test used the same 2 images from the Kaggle data set and the same method was used to read and display the images.

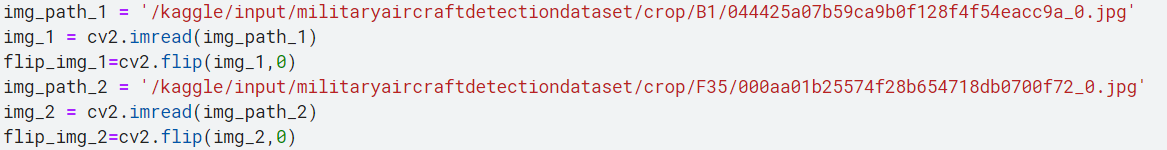
The goal of this test was to see if we can modify the image to be flipped vertically.

To do this we need to add an extra variable into our variables list

Before:



After



These extra variables ask the image to appear flipped vertically.

To display these images we do the same thing as we would the original images as shown below.



The result of this is as shown below.

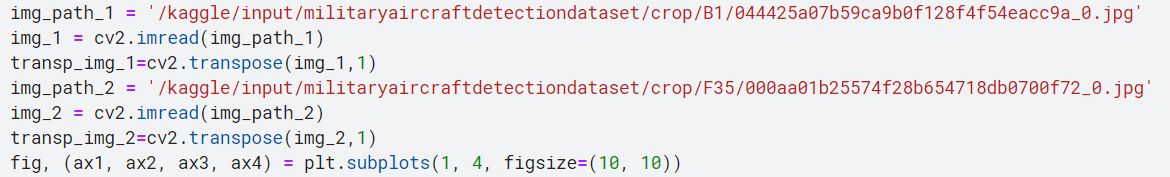
### Question 4:

This test uses the same images as the previous test, all of which were acquired from the Kaggle dataset.

The goal of this test is to see whether we are able to rotate the images 90°. Doing this opens up the opportunity to rotate the images in other degree points.

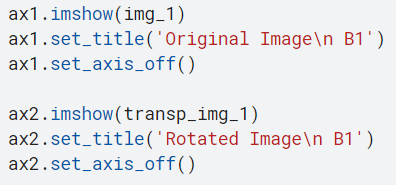
To do this we will be using the transpose command, and this will allow us to rotate the image

The following code is me using the command:

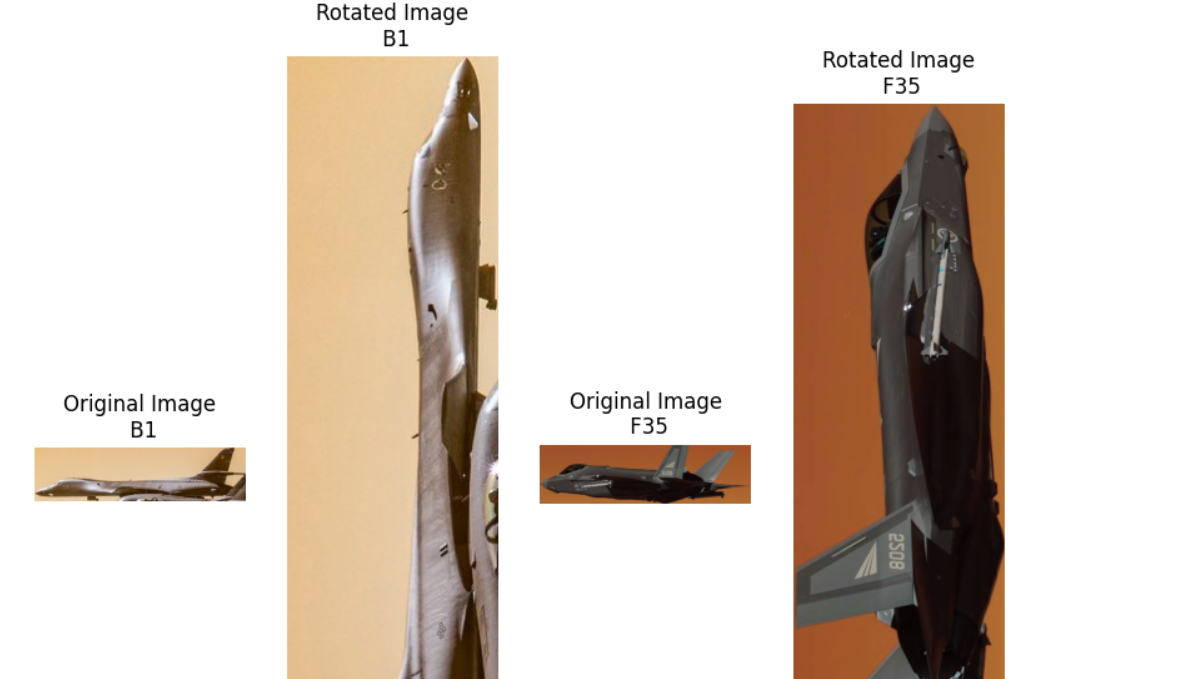


We input it as a variable so we can call upon it later.

Calling upon it is the same as flipping the image



The result of the test is as follows:



Additional note:

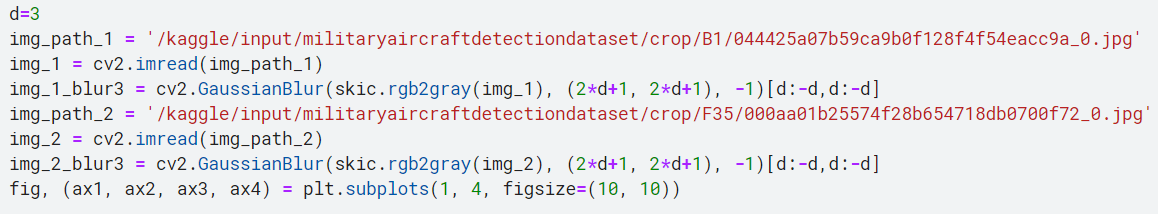
Images appear larger when rotated. A fix should be able to be done quite easily.

### Question 5:

As before the same images are being used of a B1 and a f35. Both have been obtained from the Kaggle dataset.

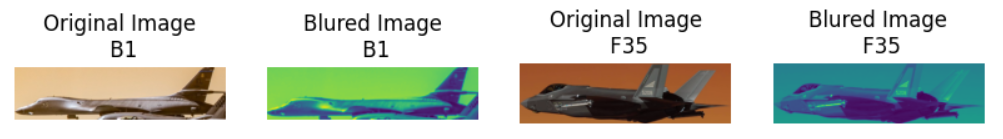
The test I will be performing here is to see whether I am able to blur the images.

To start I will add new variables to our list of variables.



The added variables use Gaussian Blur, this variable is meant to smooth images. Using this we are able to add a blur effect to the image. The variable d is the value of the smoothing, for the current test we have set it to 3.

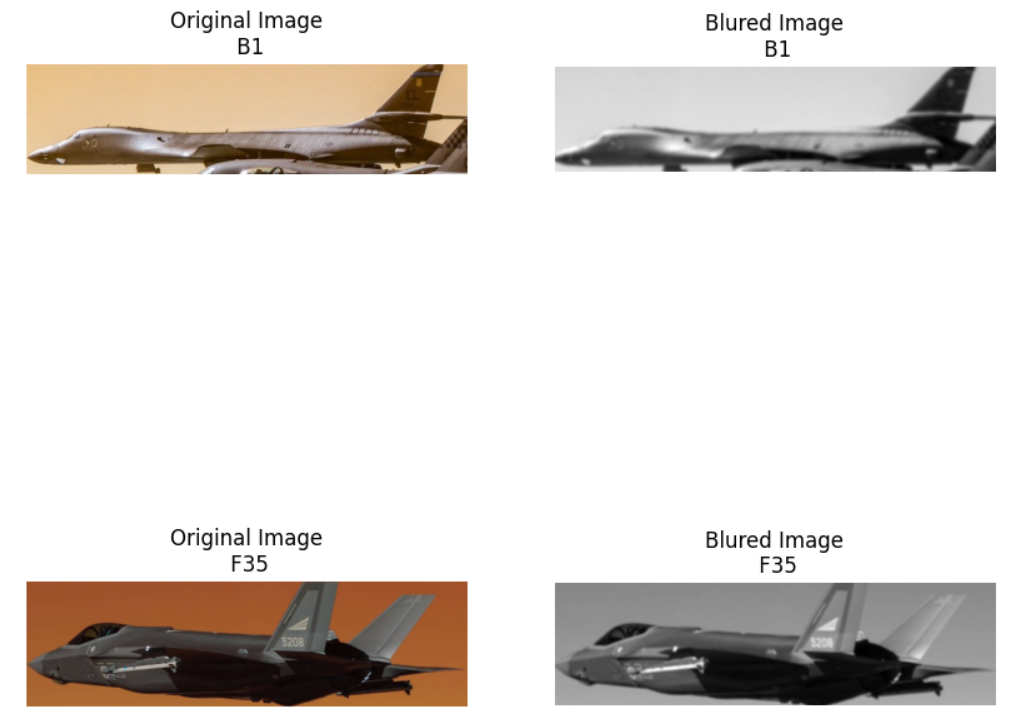
The results of the first test are as follows.



The first result is a failure as the colour of the image has been distorted, we will be changing the functions around and playing with the value until we get it right.

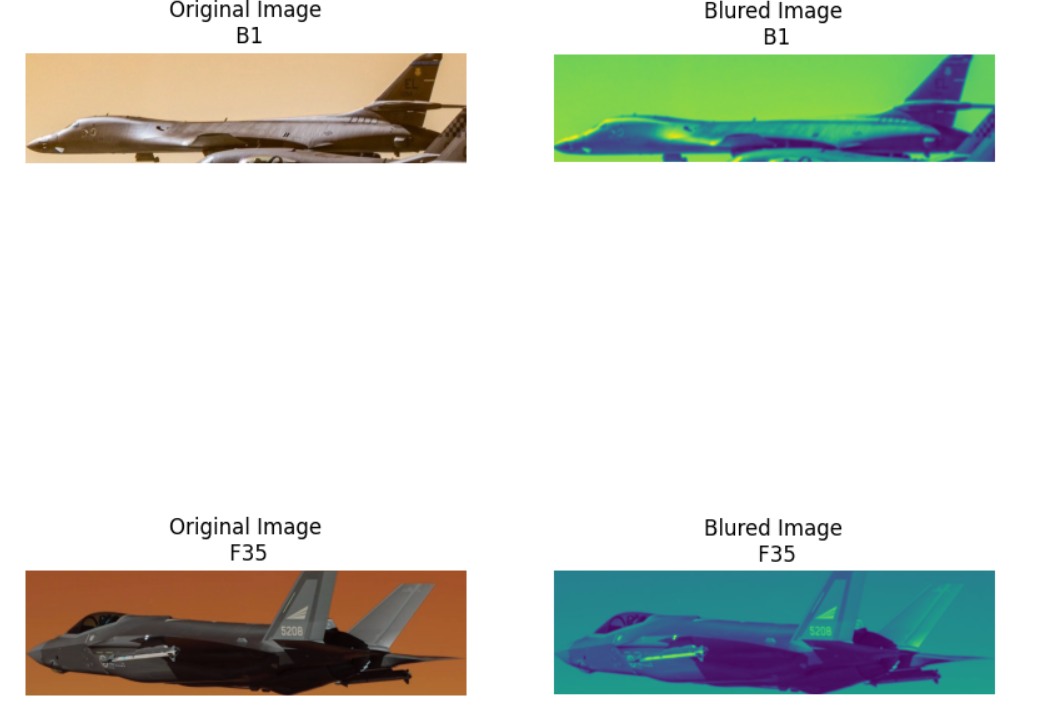
This next test we will grey scale the images to see if it changes anything.

The d variable has been set to 6, I also put the B1 jet and F35 jet into different rows for greater clarity. Finally the images have been grey scaled. Here is the result.



Through close inspection it does appear that the images are blurred, though something messes with the image colour when not grey scaled.

Upon reverting the changes while keeping the clearer picture quality I can see the images are actually blurred.



## Stage 2, Trained Model:

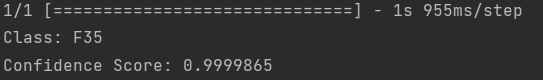
The model was trained using Google’s Teachable Machine software, this was used to quickly train an image detection model, this model was then downloaded along with the supplied code. This was put into PyCharm and the image path was replaced as in the image below.



This image path will be changed when handing in to be withing the folder i am submitting. The code supplied is via console, the supplied image looks like this:



This image is of a F35 and this is what the model identified it as.

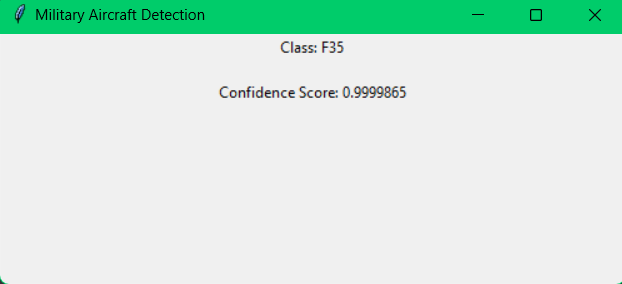


It has a 99% confidence score of it being an F35.

## Stage 3, Implementation and Deployment:

The above model was then deployed as a tkinter GUI, i imported tkinter into the main document and added labels to display the results of the test.

Using the same image i tested it via the GUI and this was the result.



The GUI displays the same information as the console. The GUI was also implemented so it doesn’t remove the console result.