**Report on Project\_0**

**1. Numpy tutorial**

In this task, I filled out some pre-existing functions to practice some basic matrix and vector manipulation functions of the Numpy library.

**Q1.a:**

Here, I used the **numpy.arange** function to get a numpy vector in a specific sequence. The function takes a single parameter which is an integer (N) and it returns a numpy vector containing a seequence from 0 to N-1.

For example, **numpy.arange(5)** returns [0, 1, 2, 3, 4]

After that, I calculated the dot product between the above numpy vector and another pre-defined vector. The function I used for this task is **numpy.dot()**. It takes two arrays (can be vectors or matrices). It returns the summation of the multiplcation of each same positioned elements of the arrays.

For example,

if x = [10, 15, 20, 25, 30] and y = [0, 1, 2, 3, 4]

then, **np.dot(x, y)** = 10 \* 0 + 15 \* 1 + 20 \* 2 + 25 \* 3 + 30 \* 4

= 15 + 40 + 75 + 120

= 250

**Q1.b:**

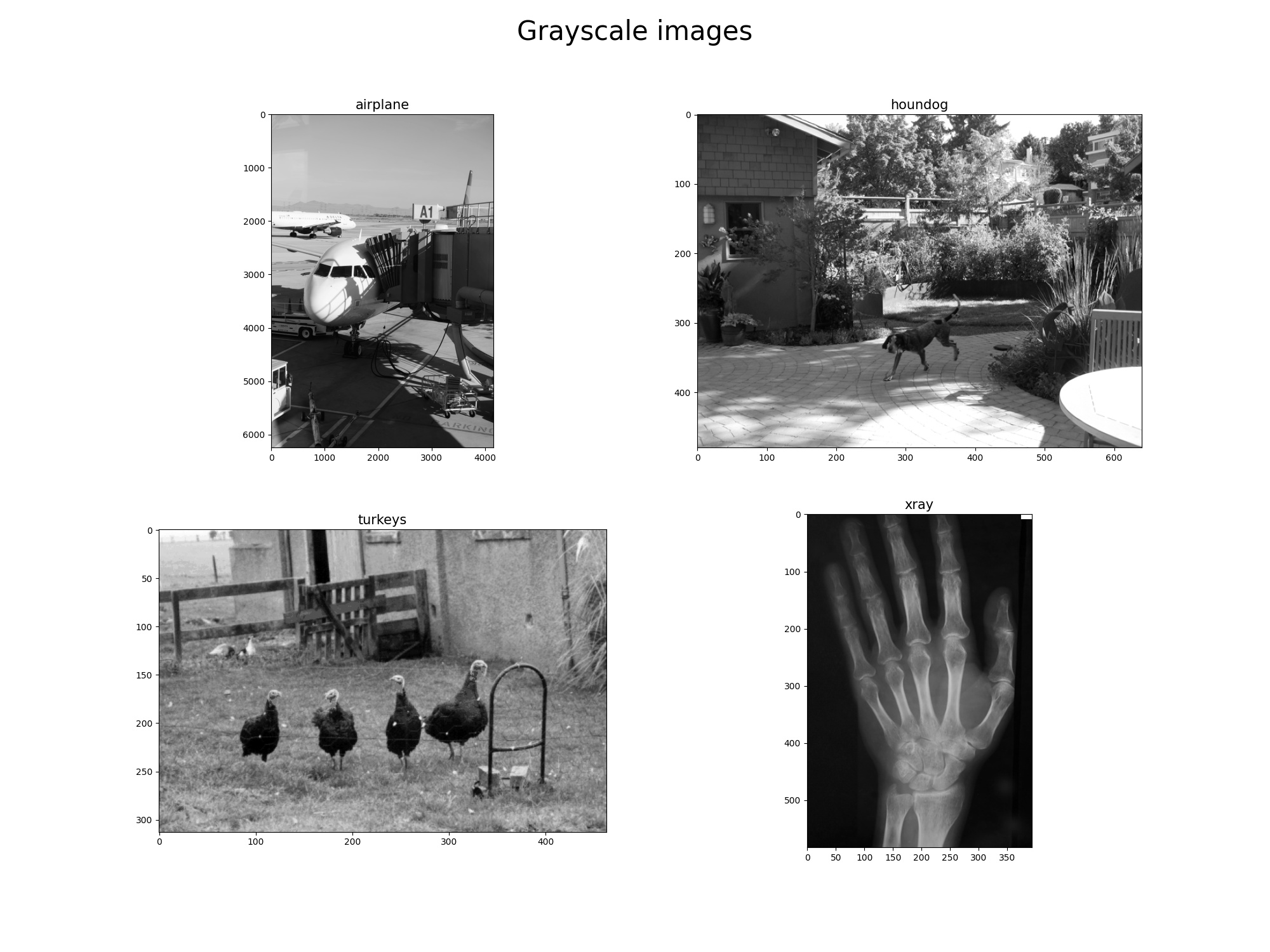
Here, I have used the arange and reshape funtion to create 2d numpy array. Then, I used the numpy.dot function to calculate the dot product of two 2-dimentional arrays.

**Q1.c:**

Here, similar to the previous question, I have used the arange and reshape funtion to create 3d numpy array. Then, I used the numpy.dot function to calculate the dot product of the 3-dimentional array and the 1-d vector of weights. The shape of the 3-d matrix is (10, 10, 3) which mimics the shape of an RGB image. After doing dot product with the weight vector the output becomes 2-d. So this function can be used to convert a RGB image to a grayscale image.

**2. Image I/O**

* First the images are read by using the Matplotlib librariy’s **pyplot.imread** function.
* Then the images are converted to grayscale images by using our own custom function where I checked whether the image is already in grayscale or not by getting the image dimension and then calculated the dot product using the provided weights.
* Then I plotted the images in 2 x 2 grid using the **pyplot.subplots** function.
* After that the grid image in saved in a file using the **pyplot.savefig** function.
* Finally, the grid image is show as output using the **pyplot.show** function.

Fig 1: The converted grayscle images in a 2 x 2 grid.