ET3 Task documentation

Problem 1:

- First importing the required libraries: os → for managing directories, shutil→ for copying images from source to destination, csv and pil→ to deal with images and datetime → to deal date format
- Then specifying where the dataset path is as well as where the destination to where it should be saved afterwards and the csv file that will be generated.
- if not os.path.exists(destinationDairies):
  os.makedirs(destinationDairies) , We use this to make sure the destination folder is existing and the system can find it.
- Then the extract_image_name function is used to cut out the prefix from the images name and would be called every time that code runs through an image. The function splits everything before the '-' because this is its sign that the prefix is ending and returns the new image name.
- Then a new file.csv is made as a report and its rows are specified using ,writerow and given 3 rows: new image name, image size in megabytes and the date of the last time the image was modified.
- A for loop then iterates over all folders and subfolders using the os.walk function and checks for all files with .png/jpg/gif/jpeg extensions, when it does it specifies the destination and source paths and uses the shutil.copy function to copy the image into the destination path.

To make the image size in megabytes we used : image_size =round(os.path.getsize(destination_path) / (1024 ** 2), 2) to etxtract the size and then divide it by 1024*1024 to turn in it to megabytes.

- After getting the name and date we write them into the report to be saved.
- A folder with the modified images would be made and report csv will be uploaded.

Problem 2:

- First we specify the Input and Output Paths.
- Then we initialize the JSON Data Structure that we want which is json_data = { "image rotation": 0,"value": []}
- So a dictionary called json_data is created to hold the data that will be written to the output JSON file. It has two initial keys: "image rotation" and "value". The "image rotation" key is set to 0 (assuming no rotation for all objects). The "value" key is initially an empty list, which will store the object-specific data.
- We read the Lines from the .txt file using : with open(txt_file_path, 'r') as txt_file: lines = txt_file.readlines()
- Then a loop iterates over each line in the lines list, representing the data for each detected object. The line is stripped of leading/trailing whitespace and split into individual values using the split() method.The values are then assigned to variables: x, y, width, and height. These values are accessed using their corresponding indices (1, 2, 3, and 4).The label variable is set to the label of the detected object.
- Json data is written to output file.
- The output JSON file is opened in write mode using a with statement. The json.dump() function is used to write the json_data dictionary into the file. The json.dump function takes care of serializing the dictionary into JSON format and writing it to the file.