Problem 1 solution

Python File name: et3 problem1.ipynb & et3 problem1.py

Deliverables:

- **allImages**: A folder which contains the images after collecting them in one folder and eliminating prefixes.
- **imagesInfo.csv:** The csv file which contains the images info in the desired form.

Name	Size (bytes)	Last Modification Date
image1.jpg	28171	Wed Jul 26 13:28:08 2023
image2.jpg	21422	Wed Jul 26 13:28:08 2023
image3.jpg	28564	Wed Jul 26 13:28:08 2023
image4.jpg	23076	Wed Jul 26 13:28:08 2023
image5.jpg	27263	Wed Jul 26 13:28:08 2023

Functions description:

- **refactorFunction (oldPath, newPath):** Function which takes two **parameters:**
  - oldPath: the path of the old images folder.
  - newPath: the path in which images will be collected in.

  Function description:
  - Using **glob** and **os** libraries the function open images folder and sub folders to get images.
  - Using os.rename we move the images to the new path.
- **getImagesInfo(imagesPath):** Function which takes one **parameter:**
  - imagesPath: the path of the collected images.

  Function description:
  - Function which returns a list of key pair values which contains the info of each image (image, size in bytes, modification date).
- **saveToCsv(imagesInfoList, csvFileName):** Function which takes two **parameters:**
  - imagesInfoList: the list which contains the info of images.
  - csvFileName: the name of csv to save data in.

Problem 2 solution

Python File name: objectDetection.py

Deliverables:

- **imagesResult**: A folder which contains the images after applying object detection on them.
- **imagesTxtFiles**: A folder which contains the .txt and .json files of all images.

Functions description:

- **objDetection(images,imageFolder):** Function which takes two **parameters:**
  - images: list which contains all images.
  - imageFolder: the path of images folder.

  Function description:
  - Using **torch vision models,** we use faster RCNN Resnet50.
  - Using **torch.cat,** to concatenate the label tensor with the x, y, width, height tensor.
  - Using **plt.savefig,** to save images after applying object detection.
- **toTxt(result,index):** Function which takes two **parameters:**
  - result: the list which contains the tensor which contains the object label and dimensions of objects.
  - Index: the index of the image.

  Function description:
  - Function which writes the result list into txt file for each image.
- **convertToJson(txtFolder):** Function which takes one **parameter:**
  - txtFolder: The folder of the txt files of images.

  Function description:
  - Function which converts txt files to json and save them in the same folder.