CP467: Course Project

Objective:

The purpose of this project is to give you some practical experience in creating professional image datasets along with applying foundational image processing techniques, particularly feature detection, object identification, and image stitching. By crafting coherent visual representations from fragmented datasets, you will enhance your critical thinking, problem-solving abilities, and innovative approaches, ensuring a holistic understanding and practical application in real-world imaging scenarios.

Dataset Preparation:

Dataset for Tasks 1 and 2:

- Select 10 everyday objects such as a book, a toy, a kitchen item, an electronic gadget etc. Take a good quality frontal image of each object. You may or may not manually crop these images to remove the background.
- Store these images in a folder named “Objects” naming them sequentially as “O1.jpg”, “O2.jpg”, …, “O10.jpg”. Also fill the mapping table titles “Object Mapping” in the file “Results.xlsx”.
- Place the first object on a table (or some other setup as you find appropriate) and take three images of the scene image from three different viewpoints – one from the front, and two from an angle (one from left and one from right). Store these three scene images in a folder named “Scenes” naming them as “S1_front.jpg”, “S1_left.jpg”, and “S1_right.jpg”, respectively.
- Keeping the first object in its position, add the second object on the table. Take three images of the scene image from three different viewpoints – one from the front, and two from an angle (one from left and one from right). Store these three scene images as “S2_front.jpg”, “S2_left.jpg”, and “S2_right.jpg”, respectively.
- Repeat this process **8** times, each time adding a new object to the existing objects on the table and storing three images in each case.
- Now you have your dataset in two folders – a dataset of 10 object images and a dataset of 30 scene images.

Note: Your scene dataset must be “reasonably” challenging.

- For scenes containing up to three objects, there must be some clutter and occlusion among the objects and no object should be 100% visible from more than one viewpoint. For scenes with four or more images, at least half of the objects must be partially occluded in each scene image.
- Among 30 scene images, at least 10 images must contain at least one rotated object.

Dataset for Tasks 3:

- Create a new scene image containing all 10 objects. The scene does not have to be cluttered.
- Take 10 images of the scene from different viewpoints. The images are taken such that 3-6 objects are visible in each image.
- For every image, there is at least one other image that has some overlap with that image.
- Create a folder named “Panorama” and store the images naming them sequentially “S1.jpg”, “S2.jpg”, …, “S10.jpg”.

Tasks:

Task 1:

- For each scene image, identify which objects are in the scene. In this task, your system needs to only identify the objects and there is no need to detect the position of the objects.
- Run your code for each scene image, storing your text output in the accompanying file named “testing.txt”. Use the format shown in the sample results in the file.
- Fill the accompanying table of results named “Results.xlsx” showing number of true positives, false positives, true negatives, and false negatives for each scene. For example, if your system detects three objects namely O1, O3, and O6 in the S4_frontal scene image, the number of TP, FP, TN, and FN will be 2, 1, 5, and 2, respectively. TP is 2 because it correctly detected O1 and O3. FP is 1 because it incorrectly detected O6. TN is 5 because it correctly said that O5, O7, O8, O9, and O10 are not present in the scene. Finally, FN is 2 because it incorrectly said that O2 and O4 are not present in the scene.
- Calculate and show the precision, recall, F1-score, and accuracy for the complete dataset.

Task 2:

- For each detected object in Task 1, draw a bounding box around the object. Also annotate the detected objects with the corresponding object name (like "book", "toy") to make evaluation more systematic.
- Store the result in a folder named “Detected Objects” appending “_bb” (for bounding boxes) with the scene image name. e.g., for input “S3_left.jpg”, store the image with detected objects as “S3_left_bb.jpg”.

Task 3:

- Construct a larger panoramic image by stitching together the individual scene images and identifying the location of each object in the stitched scene image.
- Aim for no visible seams or inconsistencies in the stitched image.
- Store the stitched image in the folder “Panorama” naming it “Panorama.jpg”.

- Detect all objects in the stitched image and draw a bounding box around each detected object. Store the image with bounding boxes in the folder “Panorama” naming it “Panorama_bb.jpg”.

Report:

The report includes:

- A statement of originality saying that the code has been written specially for this course. Clearly cite each resource or website from where any part of the code has been adapted, e.g., say “Task x code uses some starting code taken from a feature extraction example available on the official OpenCv tutorials at https://.....”.
- A statement of contribution by each group member mentioned clearly which part of the project they contributed to.
- A small literature review that discusses past and current methods on how to perform Tasks 1,3, and 3 and discusses the advantages and disadvantages of each approach.
- The methods and algorithms you have used for feature detection, matching and image stitching, and why did you select those?
- The challenges faced and how they were addressed.
- A section for possible improvements or future work.

The report should not be more than 5 pages excluding images and tables.

Deliverables

Submit a zip file containing:

- A folder named “Code” with the complete source code, ensuring it's executable on another system without additional modifications. This can include Python files (.py) or Jupyter Notebooks (.ipynb).
- All folders, images, text and tabular results as described in the project description above.
- A PDF file named “Project_Report” for documentation.

Instructions

- Use the folder structure provided with the project description on MLS. Also, make sure to follow all naming conventions and formats including those for the text file(s) and Excel sheet(s) as part of this project may be automatically graded.
- While you are free to employ a variety of techniques to fulfill the project requirements, the use of Machine Learning is not recommended. Should you choose to incorporate a Machine Learning approach, ensure that the model is retrained specifically for the unique objects in your dataset (e.g., a specific 'Gonzalez book' rather than a generic 'book').
- You may use high-level functions from OpenCV, NumPy or other libraries. Provide a requirements.txt file listing all the dependencies.
- Store all images as “.jpg” for consistency.
- Use Python version > 3.6 and OpenCV version > 3.4

- All code that you submit should either be original or, if sourced from the internet, properly cited as indicated earlier.
- Submit all materials to MLS in a single zip file.

Marking Scheme

Task	**Marks**
Quality of Datasets	20
Object Identification	20
Bounding Box Marking	10
Image Stitching	15
Object identification and Bounding Box Marking in Stitched Scene	10
Report	17
Code Structure and Execution	8
Total	**100**

A rubric for detailed marks distribution is attached with the project description on MLS.