Assignment

ARI 2129 - Principles of Computer Vision for AI

Deadline: **20th June 2022**
Group Size: **Maximum of 4 students (Equal effort will be assumed)**
Maximum Marks: **100 (50% of the entire study unit)**

Plagiarism: No plagiarism will be tolerated, and all sources are to be adequately referenced both in the code and the documentation respectively.

Marking Scheme:

- 20% for documentation (maximum of 4 pages for Part 1 and 4 pages for Part 2)
- 40% for Part 1
  - 30% for Stage 1
  - 10% for Stage 2
- 40% for Part 2
  - 15% for Task A
  - 15% for Task B-1
  - 10% for Task B-2

Note: Assignments missing any of these components in the marking scheme will be considered null and will be considered a non-submission.

Deliverables:

- 1 Single PDF document with the documentation, clearly separating Part 1 from Part 2. The document should have a maximum of 4 pages for Part 1 and another maximum of 4 pages for Part 2.
- Python source code in a Jupyter Notebook for each Part.
  - These should be presented as a single Zip file with the Surname and Name of the student and the subject code, for example, SeychellDylanARI2129.zip.
  - In the zip file, there must be the documentation in PDF and 2 folders Part 1 and Part 2 containing the notebooks with any images needed.
  - The notebooks need to work as extracted from the zip file without the need for further configuration. This means that any library of images will be either accessed within the folders themselves or via URL.

Rationale:

The aim of this assignment is to provide you with an opportunity to implement a selection of computer vision techniques that can serve as a foundation for more complex applications and their evaluation. You may use packaged OpenCV functions or those you implement yourself from first principles.

Technical Specifications

Overall Description

The first part of this assignment tackles the problem of content blending across different images. The process is outlined in Figure 1 below. You will be using the COTS dataset for this assignment, available for free through the website (www.cotsdataset.info). The COTS dataset presents scenes with an incremental nature.

Part 1: Object Blending

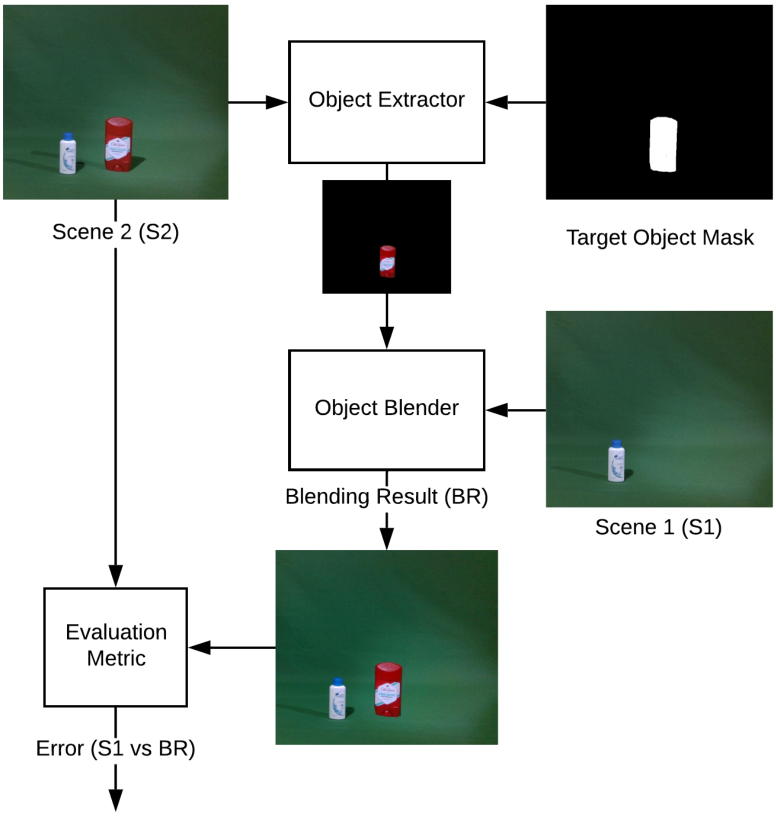

Figure 1: High-level blending architecture (Source: Dylan Seychell)

The **first stage** of this assignment involves implementing the process presented in Figure 1. In the process, Scene 1 (S1) has a single object and Scene 2 (S2) has two objects. The target mask provided in the dataset will be used to extract an object from Scene 2. Following the respective morphological processing of the mask and image subtraction, an image containing only the extracted object will be available.

The next phase is the object blender. This stage involves blending the extracted object from S2 and blending into S1. You are required to use a simple Addition Blending technique[1].

The resultant Blending Result (BR) can be compared against (S2) which will serve as the ground truth for the experiment. Error metrics (SSD and MSE) will be used to compare the quality between the two images, and you are to report on the results.

The Object Blender will be supported by an image filter using the convolution techniques covered in class and tutorials. The respective results of the BR following the use of filters would also need to be compared against the ground truth S2 and evaluated through the error metric.

All these results will need to be presented in the documentation.

The **second and last stage** of the assignment exploits the green background of the COTS dataset. You will be required to remove the green background for both S1 and S2 and replace it with a background of your choice. This will, in essence, implement a chroma-key function that changes the background for both images. The same experiment will then need to be repeated and documented using the four (4) different backgrounds.

Part 2: Image Inpainting

In the second part of this assignment, you are required to implement the evaluation code in the given paper. Figure 2 presents a pipeline for inpainting evaluation that was used in this paper. You will be required to implement the inpainting Algorithm module in Figure 2 using off-the-shelf OpenCV inpainting methods. These can be found in the link https://docs.opencv.org/master/df/d3d/tutorial_py_inpainting.html. You should use the available methods `cv.INPAINT_TELEA` and `cv.INPAINT_NS`.

The evaluation process is illustrated in the paper.

[1] An OpenCV tutorial for addition blending may be found through this URL: https://docs.opencv.org/3.4/d5/dc4/tutorial_adding_images.html

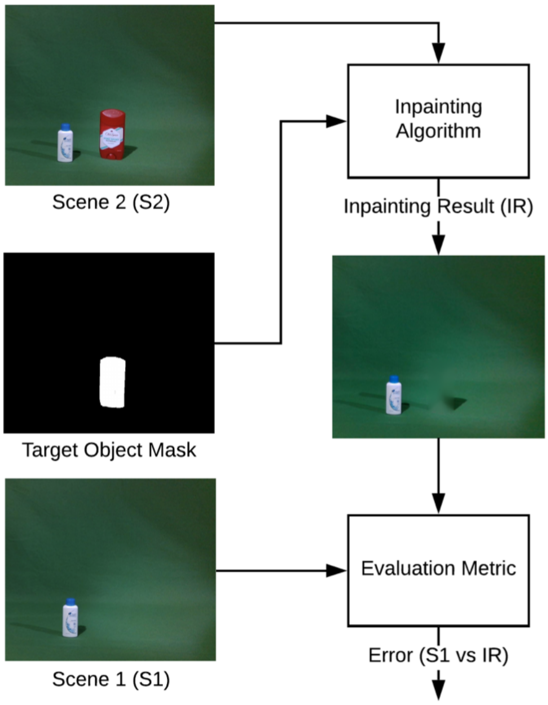

Figure 2: High-level inpainting architecture (Source: Dylan Seychell)

Part 1: Computer Vision Functions

The first part of this assignment is the implementation of computer vision functions that enable the above pipeline. Python and OpenCV will be used for this implementation. Below follows a selection of functions that are needed for the implementation of each stage. You may also develop auxiliary functions to achieve the goal as long as the same pipeline is followed.

The following functions are needed for Stage 1:

1. `ExtractObject( S2, ObjectMask)` and returns ExtractedObject
2. `ApplyFilter( ExtractedObject, FilterIndex)` and returns FilteredExObject
    a. For simplicity, it is advised that FilterIndex is simply a number and the function switches between pre-defined kernels. Index 0 can be used to tell the function to leave the object as is, without applying a convolutional filter.
    b. *Note: you may either use the implementation you have worked on in the first assignment or the OpenCV implementation.*
    c. Three filters need to be implemented.
3. `ObjectBlender(S1, FilteredExObject)` and returns BlendingResult.
4. `CompareResult( BlendingResult S2, metric)` and returns the error value.
    a. The same concept of an index should be used for the choice of error metric.

The following functions are needed for Stage 2:

1. `RemoveGreen(img)` and returns the same image without the green background.
2. `NewBackground(imgNoBg, NewBackground)` returns the updated image with a new background.

Part 2: Implementing a CV Research Paper

In this part of your assignment, you are required to re-implement the code for the two papers specified below. The implementation of code and techniques found in research papers is a very important skill in the research process in order to be able to generate comparative results.

The full paper in question is available on the VLE and is cited as follows:

1. D. Seychell and C. J. Debono, "An Approach for Objective Quality Assessment of Image Inpainting Results," 2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON), Palermo, Italy, 2020, pp. 226-231

Deliverables:

- PDF documentation
- Jupyter Notebook with functions organised in Task A and B respectively.

Task A - Replicate the code and results

Replicate the code in each of the papers together with the respective results excluding any examples or results related to deep learning. The computation of time performance is not required. You are required to use Python 3.6 and OpenCV 3.0 or later. Good use of functions and good practice in coding is expected.

You are to use the COTS dataset available on www.cotsdataset.info and are expected to use your code to generate results similar to those presented in the paper.

Task B - Evaluate a new part of the dataset

Once you have completed the replication of code in Task A of this part of the assignment, you are required to try it out using a new part of the dataset. A new addition to the COTS dataset includes a set of images, constructed in the same way as the original dataset but with a complex background. The masks for each scene are also available in their respective folders. The new part of the dataset can be found on https://github.com/dylanseychell/COTSDataset/tree/master/Part%203%20-%20Complex%20Background

This part of the dataset is organised by topic in a similar way to the original dataset. However, there is a new variant (W - with wind) or (NW - no wind). The sets labelled ‘W’ should have a more varying background.

Deliverables:

1. You are to evaluate the inpainting algorithms in Task A on 6 sets from the new dataset with complex background. This set of six should include different instances that include ‘NW’, ‘W’ and ‘NO’ sets.
2. Experiment with different visualisation methods to demonstrate the effect of a changing background. This can include methods such as background subtraction or any other method that shows differences in the background. The same methods can be also tested using the original dataset, where the changes in the background should be minimal to none.

End of Assignment