

# Assignment 2

---

For the second assignment (Total Points: 100), you are required to do the following:

## Prerequisites:

You should already have the following setup:

- A working jupyter notebook environment
- A stable Python environment with OpenCV, Matplotlib and other dependencies.

Setting up a stable environment for coding is your responsibility.

Deadline: April 23rd, 2025 11:59 PM

## Convolutions/Filtering (Total points: 20)

1. Fill in the function, `convolution2D(image, kernel)` from `util.py`. You are only allowed to use Numpy. Use appropriate padding to account for border pixels. Make sure you mention why you use that specific padding technique. (Points: 10). The `util.py` is also your deliverable along with the `.ipynb` file.
2. Test the `convolution2D(image, kernel)` function from `util.py` on the image (NYC skyline) supplied, `NYC_Manhattan.jpeg` with the *embossing kernel* taught in class. You would need to create a jupyter notebook (`<Your_name>_Assignment2.ipynb`) and import your function from `util.py` as shown in `practical03.ipynb`. (Points: 5)
3. Test the `convolution2D(image, kernel)` function from `util.py` on the image supplied (NYC skyline, `NYC_Manhattan.jpeg`) with the *Laplacian kernel* taught in class. (Points: 5)

## Image Sketch (Total Points : 20)

4. Given the image, create a line sketch of the NYC Skyline using methods from Computer Vision, so far taught in class. At every point you should describe your code with comments. Failure to do so would result in penalty.

## Creating panoramas (Total Points : 35)

5. You are required to create a panorama using the images from `office_2/officepic*.jpg`. You will be graded based on the following:
  - Create a function in `util.py` that reads the images reserved for panorama and stores them as a list for processing. Your function should be able to read arbitrary number of image. (Points : 5)
  - Create panorama using OpenCV as shown in tutorial, `practical_03.ipynb` (Points : 10)
  - Test your code using images, from `boats/boat*.jpg` and display the panorama. (Points : 5)

- ###(Mandatory for graduate students)### Create a post-processing function to get rid of stitching artifacts, such as color imbalances, inconsistencies near the border (Points : 15)

## Write-up on RANSAC [Total Points: 25]

6. In your own words, write a paragraph on RANSAC algorithm, and why it is particularly useful for image stitching. You are allowed to use online resources for this answer but the write-up you submit *must* be your own words.

## Deliverables and Submission Guidelines:

There are 3 deliverables for this assignment.

- The jupyter notebook with answers to questions written as markdown (Qs. 1) and displayed on screen (Qs. 2, 3, 4, 5). You should add comments to your code at every step. If you forget to do so, you won't get full score. **Naming convention:** <Your\_name>\_Assignment2.ipynb eg. John\_Doe\_Assignment2.ipynb
- The `util.py` with your defined functions.
- Write-up on RANSAC algorithm (for graduate students only) as a pdf or word file. **Naming convention:** John\_Doe\_Assignment2.pdf or John\_Doe\_Assignment2.docx
- Email your submissions (as a zipped file) to me. If you use github repos for submission, make sure it is not public and add me as a collaborator.
- Always reference/cite your online resources for codes and write-ups. Failure to do so would result in penalty.

## Reference

Boat images from [https://github.com/opencv/opencv\\_extra/blob/4.x/testdata/stitching/](https://github.com/opencv/opencv_extra/blob/4.x/testdata/stitching/)