CV (Spring 2021)

Name:

**Template Matching with Gaussian Pyramids:**

In this project, you will work with pyramids to implement hierarchical template matching. Use OpenCV PyrDown to create a 3-level Gaussian Pyramids (original image plus two lower levels).

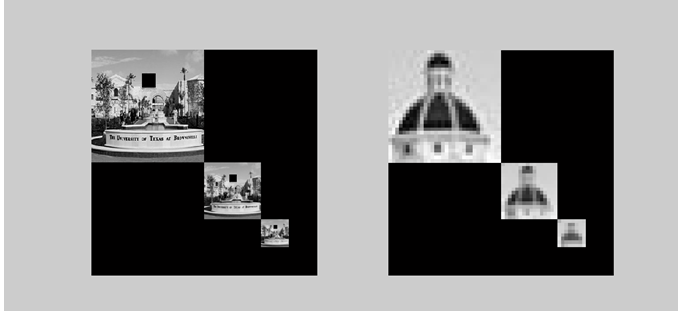
Create a function which will take the *image and the template as parameters*. It should display the matches at every level of the pyramid. The prototype could be something like this:

pyramidTempMatch(image, template, no\_of\_levels, metric)

pyramidTempMatch('UTB\_Gray256x256.jpg','UTB\_Dome.jpg',3, ‘ncc’)

The output should be something like this, where the block boxes indicate where the best template was found at each given level.

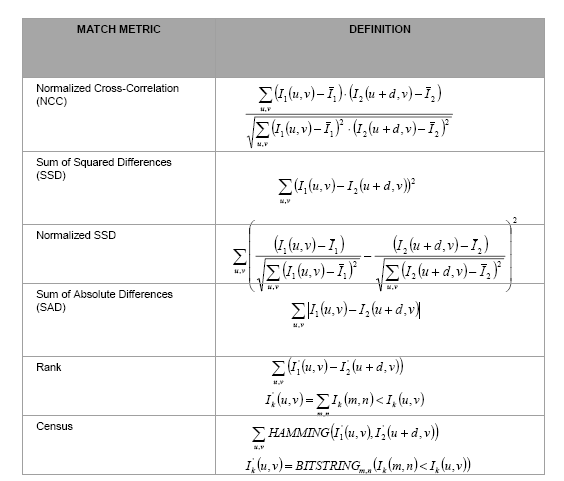
* Resize your input image to be a power of 2 for ease of implementation.
* For the template, extract a 32x32 area of interest from the original image at the lowest level (original resized image).
* For example, I extracted the dome as 32x32 window from the lowest level (the 256x256 utb.jpg image)
  + The image and the template could have pyramids (as shown below). The algorithm in the slide’s appendix (shown at the end of this project as well has only one template of size m x m). Try the one template, and if it doesn’t give good results, then create a pyramid for the template as well.



Do ***full search matching*** on the lower level first (smaller size and lower resolution, once you found the best match at that level, you would go to the next level (finer resolution) and do a region search centered around the best match location from the previous level. The ***regional search*** should take a size indicating how far around the current match should the search be. In the class notes in the appendix, the window size is given as *m x m* for lowest resolution level, and it gets doubled in the next level.

For each level, you will need to keep track of the best match. Since we are using a square template, the best match is the top left corner. Please see the slides (for example the algorithm at the appendix) to see how to implement your matching.

For the metric to use, implement first and the third from the following table.



**Grading and Submission Guide:**

* Must submit the whole project (python folder with code, image dataset, and results) zipped using 7zip tools with the name: LastName\_FirstName\_Project-03.
* For this project, put a sample of screen shots of your program run in the report.
* This is an **individual** project: The work should represent your own: that you acknowledge that have not incorporated into this project any unacknowledged material from the work of another person, including papers, words, ideas, information, computer code, data, evidence-organizing principles, or style of presentation taken from the Internet, books, periodicals, or other sources.

