**Assignment 1: Image filtering & Hybrid Images (adopted from** <https://www.cc.gatech.edu/classes/AY2016/cs4476_fall/proj1/>)

**Submission Deadline: September 15, 2019 (9 PM)**

**Deliverables:** Project Report in BMVC format and code in Python/Matlab/C/C++/whateverprogramminglanguageyouknow.

You are nevertheless strongly encouraged to code in Python! **If you are coding in Matlab/Python,** **complete the four scripts in ‘Code’ folder and include the image outputs as part of the report (No README is required in this case).** Note that some commands may be incorrect, so correct the commands where required. For other implementations, A README file should accompany your code file, and specify how to run the code. **Your code will not be graded without this README!**

**Objective:** Write an **image filtering** function to create **Hybrid Images** using a simplified version of the SIGGRAPH 2006 paper by Oliva, Torralba, and Schyns.

**Key idea behind Hybrid Images:** High frequency dominates perception when visible, but, at a distance, only the low frequency (or smooth) part of the image signal is visible.

By blending the high frequency components of one image with the low-frequency components of another, you synthesize a hybrid image which induces different interpretations at different distances.

**Specifics:**

**Image Filtering:** Image filtering (or convolution) is a fundamental image processing tool. See chapter 3.2 of Szeliski and the lecture materials to learn about image filtering (specifically linear filtering).

Python has numerous built-in and efficient functions to perform image filtering, but you will be writing your own such function from scratch for this assignment. More specifically, you will implement *my\_imfilter()* which will replicate the default behavior of the function(s) *scipy.misc.imfilter()/scipy.ndimage.filters.correlate()*.

Your filtering algorithm must support (1) both grayscale and color images (2) arbitrary shaped (sized) filters, as long as *both* dimensions are odd (e.g. 7x9 filter size is allowed but not 4x5) (3) pad the input image with zeros and (4) return a filtered image with the same resolution as the input image. You have been provided a script, *proj1\_test\_filtering.py* to help you debug your image filtering algorithm.

**Hybrid Images.** A hybrid image is the sum of a low-pass filtered version of the one image and a high-pass filtered version of a second image. There is a ***free parameter***, which can be ***tuned*** for each image pair, which controls how much high frequency to remove from the first image and how much low frequency to leave in the second image. This is called the **"cutoff-frequency"**. In the paper it is suggested to use ***two cutoff frequencies*** (one tuned for each image) and you are free to try that as well. In the kickstart code provided to you, the cut-off frequency is controlled by changing the standard deviation of the Gaussian filter used in constructing the hybrid images.

We provide you with 5 pairs of aligned images which can be merged reasonably well into hybrid images. The alignment is important because it affects the perceptual grouping (read the paper for details). We encourage you to create additional examples (e.g. change of expression, morph between different objects, change over time, etc.). See the hybrid images project page (<http://cvcl.mit.edu/hybrid_gallery/gallery.html>) for some inspiration. The project page also contains materials from their Siggraph presentation.

**What the report should contain:**

1. Your methodology stated briefly in words, and written in the form of an **Algorithm** (see <https://www.youtube.com/watch?v=l7Z7tvCkQrg> for guidance on how to write an Algorithm on Latex).
2. Hybrid image results for the five pairs, and any interesting observations you could make.
3. Your takeaways and insights you could get from the assignment.

**SUBMIT YOUR CODE, REPORT and OUTPUT IMAGES in the form of a zip file.**