ECE 5460

Image Processing

Fall 2023

Project 1

Posted: Wednesday, September 20, 2023 **Due:** Friday 23:59, September 29, 2023

We recommend you use the =basic Python notebook discussed in the tutorials. You can run it locally or on Google Colab. To use Colab, upload it to Google Drive and double-click the notebook (or right-click and select Open with Google Colaboratory), which will allow you to complete the problems without setting up your own environment. Once you have finished, make sure all the cells are run before downloading the notebook).

Submission Instructions: Please submit three categories of files on Canvas: (1) the Python notebook with the relevant source code and with all the cells run, and (2) The filtered input output images in jpeg format (3) A typed !! (latex, word, etc) report saved as as a pdf file that includes all input and output images and a brief description of each image

Late Submission Policy: As discussed in the late policy. Due dates will be strictly enforced. 20% penalty for every late day. Solutions will be posted to the course web page in 5 days. After solutions are posted, no credit will be issued for late work.

Problem 1 *Filtering*

This project is intended to familiarize you with Python, NumPy and image filtering. Once you have created an image filtering function, it is relatively straightforward to construct hybrid images.

This project requires you to implement 5 functions each of which builds onto a previous function:

```
cross_correlation_2d
convolve_2d
gaussian_blur_kernel_2d
low_pass
high_pass
```

Image Filtering. We have covered image filtering with various examples in class. Both numpy and opencv 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 to improve your understanding, More specifically, you will implement `cross_correlation_2d`, followed by `convolve_2d` which would use `cross_correlation_2d`.

Gaussian Blur. To implement Gaussian blur, you will implement a function `gaussian_blur_kernel_2d` that produces a square kernel of a given size which can then be passed to `convolve_2d` implemented before, along with an image, to produce a blurred version of the image.

High and Low Pass Filters Recall that a low pass filter is one that removed the fine details from an image (or, really, any signal), whereas a high pass filter only retails the fine details, and gets rid of the coarse details from an image. Thus, using Gaussian blurring, implement `high_pass` and `low_pass` functions.

Stubs for these functions are provided in the `project1.py` file on the canvas page. You should write a test script, that accomplished the following.

1. Load the cameraman. jpg image and convert to floating numbers and scale to [0,1]
2. Take a 7×7 subimage and correlate the images with the subimage you used. Plot the resulting correlation output. Interpret the output.
3. Create,and print Gaussian Kernel of size 7 and sigma=2
4. Low pass filter the image with gaussian filter of size 7 and sigma=2. Plot the resulting filtered output.
5. Low pass filter the image with gaussian filter of size 7 and sigma=2. Plot the resulting filtered output.
6. Filter the image with the horizontal and vertical $3{\times}3$ Sobel Filters and Plot the resulting filtered outputs.

Repeat Steps 2-6, with a *color* image of your choice.

For just this assignment, you are not allowed to use any Numpy, Scipy, OpenCV, or other preimplemented functions for filtering, convolution and correlation. You are allowed to use basic matrix operations like np.shape, np.zeros, and np.transpose. The aim is to use simply multiple for loops or Numpy vectorization to apply a kernel to each pixel in the image using the basic definition of correlation and convolution we reviewed in class.

Problem 2 *Hybrid images*

In this problem you will create hybrid images as described in [1]. Please read the paper before starting this question.

Take two images, A and B, that you'll want to have blend from one to the other. You can use any two images you like, be creative! For a visually pleasing effect, make sure the objects in the two images occupy the same region. Construct a hybrid image from A (to be seen far away) and B (to be seen close up) as follows:

$$\text{out} = \alpha\ \texttt{low\_pass}(\text{A,sigma1}) + (1-\alpha)\ \texttt{high\_pass}(\text{B,sigma2})$$

Where `low_pass` and `high_pass` are functions that low-pass and high-pass filters the images you developed in the previous question . You will have to experiment with various values of α (the mix in factor) as well as sigma1 and sigma2 for a visually pleasing effect. In your report, please specify the t parameters you used. show your input images labelled clearly as A and B, and include the result. You should strive to create the illusion as described in the paper.

Acknowledgment: Project based on versions developed by James Hays, Noah Snavely and Derek Hoiem.

References

[1] Aude Oliva, Antonio Torralba, and Philippe G Schyns. Hybrid images. *ACM Transactions on Graphics (TOG)*, 25(3):527–532, 2006.