CMP_SC 8690 - Computer Vision
HW1A: Hybrid Images in Python
Elizabeth Nash
2/6/2025

Abstract

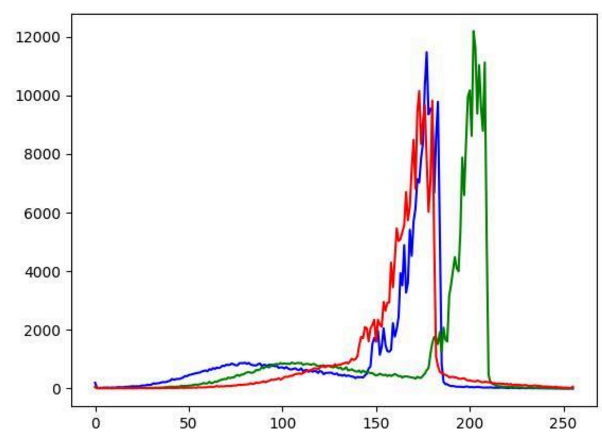
The primary purpose of this introductory assignment is to develop familiarity with Python image processing libraries while producing hybrid images. I accomplished this task using OpenCV for image manipulation and MatPlotlib to generate and save histograms of the resulting images.

Introduction

Hybrid images consist of the low-spatial scale of one image overlaid with the high-spatial scale of another. The resulting image has the blurred colors of the former image with the fine details of the latter. The low-pass image is generated using a Gaussian filter, which is subtracted from the original to generate the high-pass image. This process is applied to two different images, and their high-pass versions are combined with the low-pass of the other to produce the final hybrid image. Additionally, the high-pass image is added back to the input to create a sharpened version.

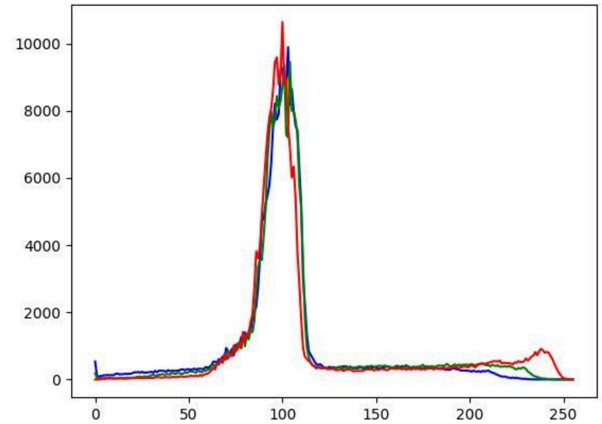
Results and Discussion

Input One



Input One Histogram

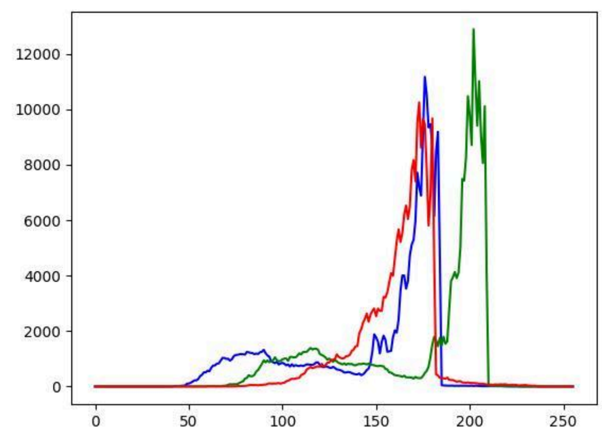
Input Two



Input Two Histogram

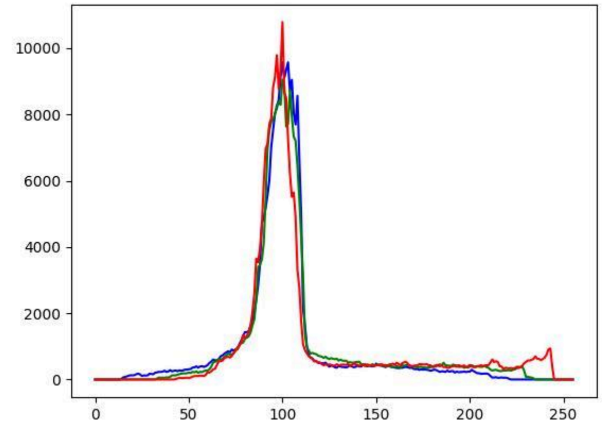
For all images, I produced a histogram using OpenCV's calcHist() to generate the data and Matplotlib to display it. Each channel is represented according to its color. The distribution of each color is roughly equivalent for both images. The cat image's green channel is the only exception, due to the background skewing the average green intensity higher.

Input One Low-Pass
(Sigma 5)



Input One Low-Pass Histogram

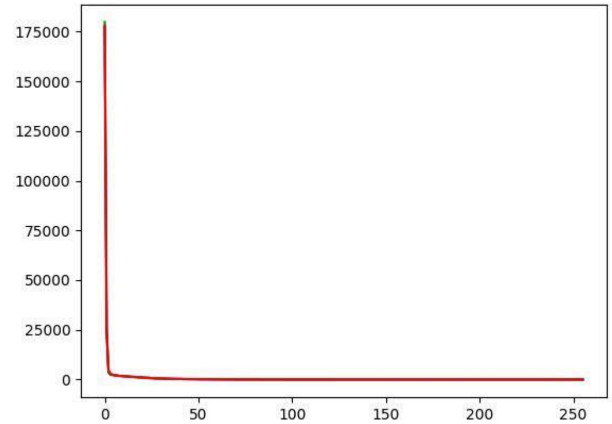
Input Two Low-Pass
(Sigma 5)



Input Two Low-Pass Histogram

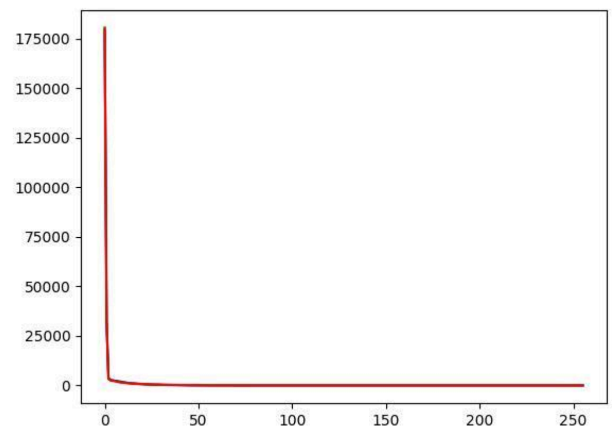
The low-pass filter was implemented using OpenCV's GaussianBlur() function. The details in both images are blurred, making it more difficult to identify the animal's features, but the image histograms are relatively unaffected. The filter essentially applies a weighted average of each pixel's neighborhood, meaning the general distribution of each color remains stable. This does make their trendlines somewhat smoother, best demonstrated by the cat's red channel. The harsh jumps and falls leading to its peak are minimized.

Input One High-Pass
(Sigma 5)



Input One High-Pass Histogram

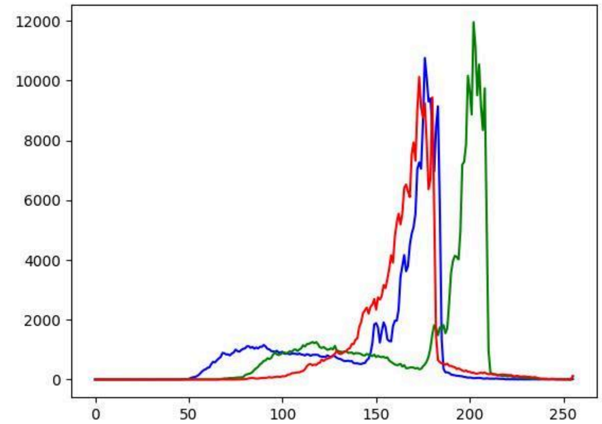
Input Two High-Pass
(Sigma 5)



Input Two High-Pass

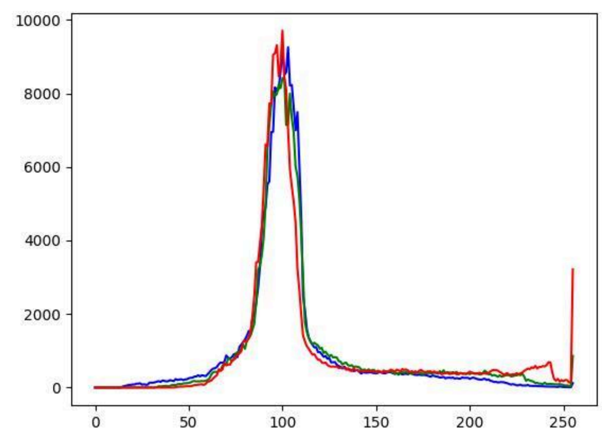
Subtracting the low-pass result from the original input gives the high-pass images seen above. This process essentially extracts the excess value of the pixels most smoothed by the Gaussian filter. Those are the pixels that differ most from their surroundings, which tend to be the fine details. The vast majority of pixels' values only shifted slightly due to the smoothing, meaning subtracting their original value yields a result near zero. Both histograms reflect this.

Low-One High-Two Hybrid
(Sigma 5)



Low-One High-Two Hybrid Histogram
(Sigma 5)

Low-Two High-One Hybrid
(Sigma 5)

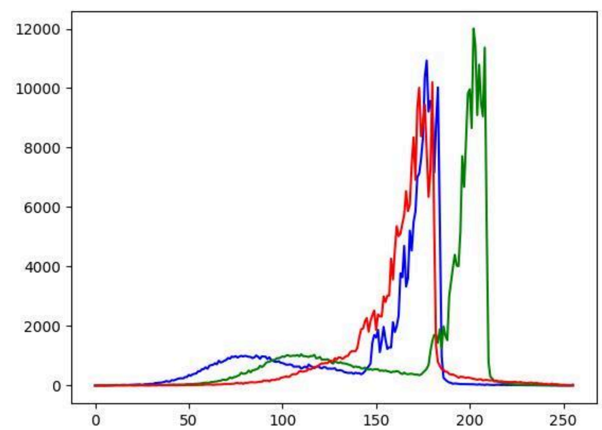


Low-Two High-One Hybrid Histogram
(Sigma 5)

Merging the two images preserves the fine details of one input seemingly overlaid on top of the smooth version of the other. The human brain's visual processes conjure this apparent order, perceiving the high-pass animal as "in front of" the low-pass animal, despite their pixels being added without any bias. The fine details make the high-pass animal appear in focus, causing us to make the assumption that the low-pass animal must be further away.

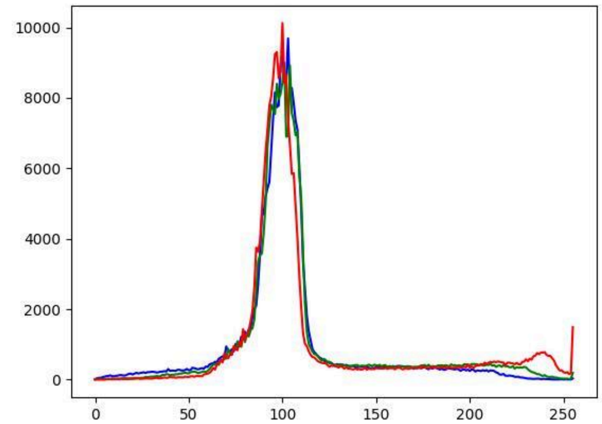
Since the high-pass filter preserves so much less than the low-pass filter, the merged histogram mostly mirrors the smoothed histogram. The main exception to this is the high end of the second image's histogram, where there is a sudden jump right at 255. This is due to the implementation of the merge process, where values are added without any regard to the UInt8 data type. Any areas where the cat's details were added to bright pixels of the dog overflowed the maximum possible value.

Low-One High-Two Hybrid
(Sigma 1)



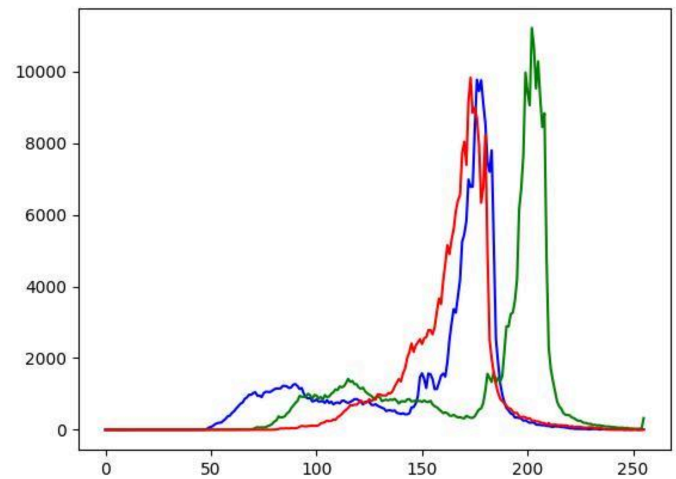
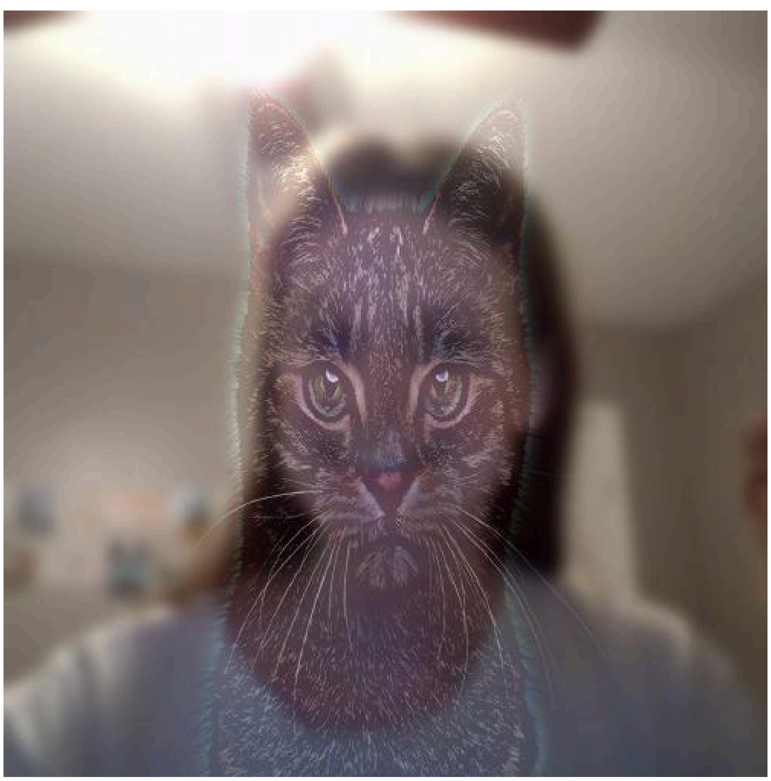
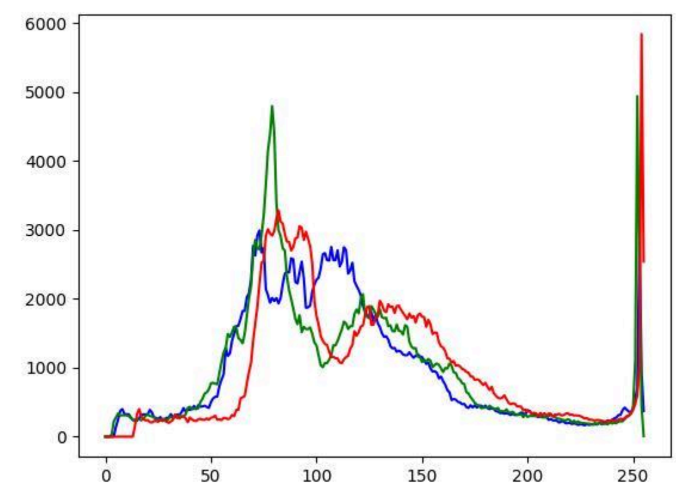
Low-One High-Two Hybrid Histogram
(Sigma 1)

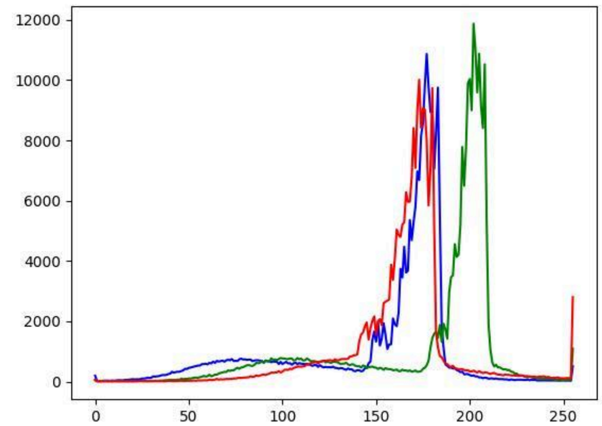
Low-Two High-One Hybrid
(Sigma 1)

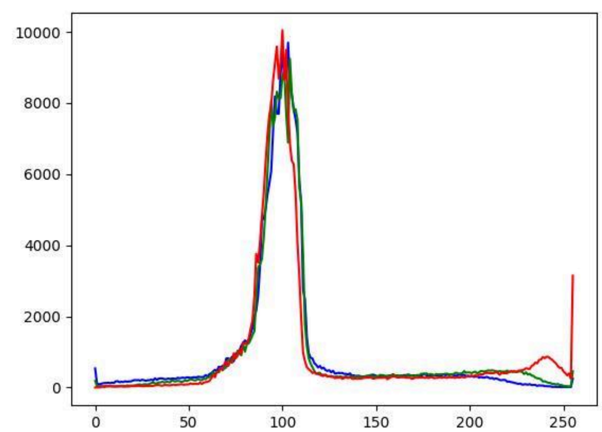


Low-Two High-One Hybrid Histogram
(Sigma 1)

I repeated the merging process using a Gaussian filter with a low sigma, which barely smooths the image. The low-pass result looks similar to the input, while the high-pass detail is barely perceptible. The cat’s fine details are easier to distinguish, which is two-times the result of its higher level of visual complexity. Firstly, it gives more details to focus on in extracting its overall face. Secondly, the details in its low-pass version hide the dog’s features, as they can be interpreted as variations in the color of the cat’s fur.

 Low-One High-Ellie Hybrid (Sigma 5)	 Low-One High-Ellie Hybrid Histogram (Sigma 5)
 Low-Ellie High-One Hybrid	 Low-Ellie High-One Hybrid Histogram (Sigma 5)

<table>
<tr><td>(Sigma 5)</td><td></td></tr>
<tr><td colspan="2">Additionally, I merged the image of a cat using a picture of me. The result is much more disturbing, and much less like an Animorphs cover, than I hoped. The image of me was taken in low light, which allows the cat's details to be far more distinguishable when laid on top of me. This does not impair the high-pass filter, though.</td></tr>
<tr><td>
Sharpened One
(Sigma 5)</td><td>

Sharpened One Histogram</td></tr>
<tr><td></td><td>

Sharpened Two Histogram</td></tr>
</table>

Sharpened Two (Sigma 5)	
The sharpened versions are produced by adding the high-pass result on top of the input, which makes each animal's fine details brighter. The histograms are mostly unchanged, save for the recurring overflow on the dog image.	