

Nashville Instagram Filter

Anders Dahl

Abstract

The purpose of this research is to identify and replicate the effects used by Instagram's Nashville filter. The Nashville filter utilizes level [3], brightness [1], and color [2] adjustments to give it a low contrast, high exposure, and warm temperature. It adds a pastel, slightly pink and pleasant palette to your photo. [5] I attempted to recreate these adjustments in my own custom filter, but at the end, there were still some slight differences.

Contents

1	Introduction	3
2	Methods	4
2.1	Level Adjustment	4
2.2	Brightness Adjustments	4
2.3	Color Ratio Adjustments	5
3	Code	6
3.1	Contrast	6
3.2	Gamma	6
3.3	Tint and Input Color	7
4	Results	7
5	Discussion	8

1 Introduction

The purpose of this research is to identify and replicate the effects used by Instagram's Nashville filter. The Nashville filter utilizes level [3], brightness [1], and color [2] adjustments to give it a low contrast, high exposure, and warm temperature. It adds a pastel, slightly pink and pleasant palette to your photo. [5] The result can be seen below:

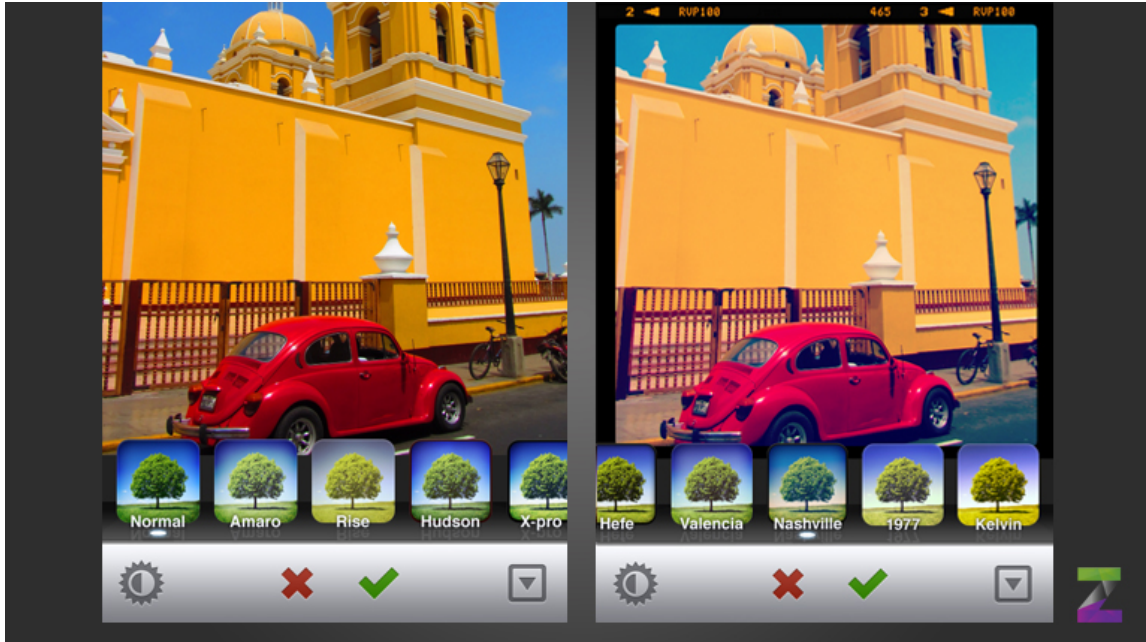


Figure 1: Instagram's Nashville Filter.

I was able to roughly map out what the Instagram filter was doing by looking at the histogram of the RGB channels of the before and after photos. The article 'How to make Instagram Filters in Photoshop: Nashville and 1997' was also an invaluable source that gave me a general direction of what to look for. In the end, I concluded that some level, brightness, and color adjustments were present. The color adjustments caused a tint resembling a maize-like-color.

I attempted to recreate the Nashville filter by altering the contrast, gamma, tint (color ratios), and input color to try to get a similar effect. I compared the original, instagram, and custom image side-by-side with their respective color channels on display in a histogram. Using this technique, I was able to figure out what I needed to adjust in order to get a closer fit. My custom filter was able to get close to the Instagram filter, but there were still some slight differences.

2 Methods

2.1 Level Adjustment

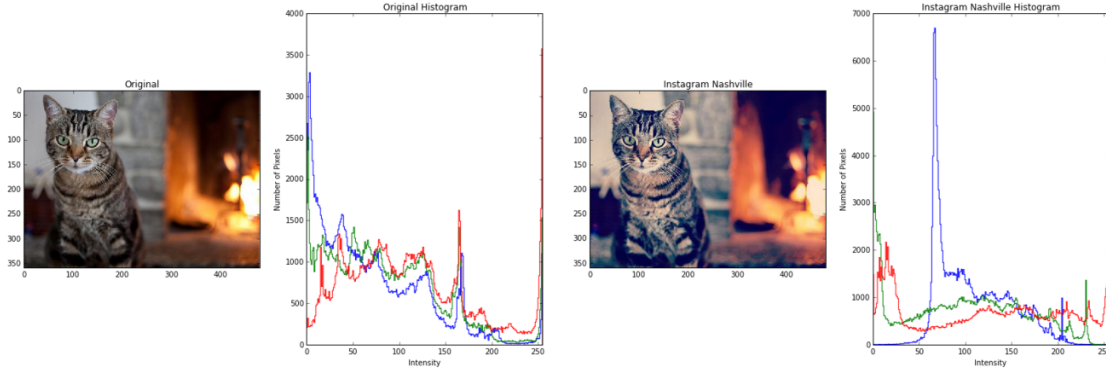


Figure 2: Color Input Adjustments.

You can adjust the level to correct the tonal range and color balance of an image by adjusting the intensity. [3] The input intensity of the Instagram Nashville filter is different; hence, a level adjustment seems to be occurring. This is illustrated in figure 2. For example, the blue channel spikes upwards at an intensity of about 65 compared to the spike at 0 in the original photo. The green channel has a similar adjustment, but it is not as substantial. After testing on multiple images, it seemed like the input color adjustments were close to the following: [5]

Red Channel 0

Green Channel 18

Blue Channel 88

2.2 Brightness Adjustments

Instagram’s Nashville filter adjusts the brightness across the board. The intensity of each channel has increased. In figure 2 we see that many more pixels are in a higher intensity region.

2.3 Color Ratio Adjustments

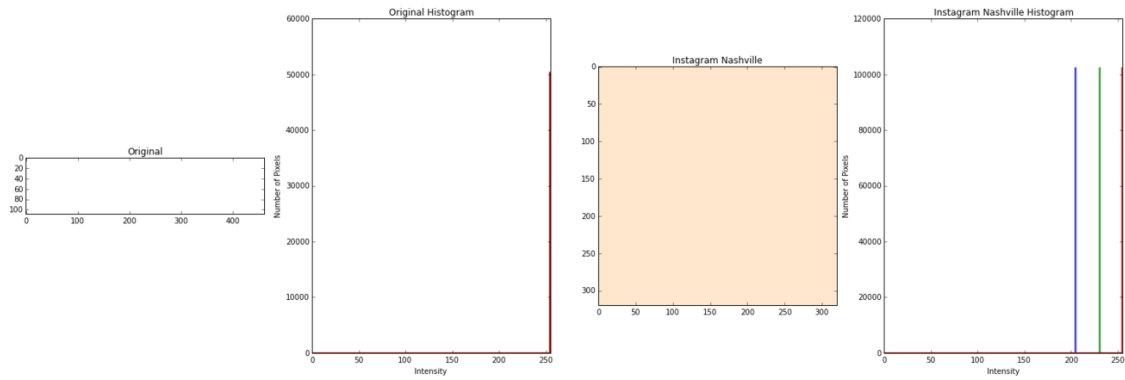


Figure 3: Color Ratio Adjustment

If we run Instagram's Nashville filter on a white image, we notice that a maize-colored tint has been applied. The above figure 3 shows this. The Instagram histogram clearly shows the shift in the red, green, and blue channels. The maize color is close to #F6D8AC or RGB(246, 216, 172). We can approximate the ratio change for each channel by dividing those values by the white, RGB(255, 255, 255).

Red Channel Ratio Change .965

Green Channel Ratio Change .847

Blue Channel Ratio Change .6745

3 Code

After analyzing the photos, I started coding. After some manual testing I found three functions that were essential in creating a similar custom filter. I used the Skimage Exposure library with Python to achieve this, and all my code can be found in the iPython Notebook provided.

1. Contrast
2. Gamma
3. Tint and Input Color

3.1 Contrast

Contrast

```
# http://scikit-image.org/docs/dev/api/skimage.exposure.html  
  
# Performs a contrast adjustment. This function transforms  
# the input image pixelwise according to the equation  
#  $O = 1/(1 + \exp(\text{gain} * (\text{cutoff} - I)))$  after scaling each  
# pixel to the range 0 to 1.  
def adjust_contrast(image, cutoff=.5, gain=1):  
    return exposure.adjust_sigmoid(image, cutoff=cutoff, gain=gain, inv=False);
```

Figure 4: Contrast Adjustment

Run a contrast adjustment. This function transforms the input image pixelwise according to the equation $O = 1/(1 + \exp(\text{gain} * (\text{cutoff} - I)))$ after scaling each pixel to the range 0 to 1. [4] It can be used to adjust the brightness and sharpness of the image. After some experimenting, I found that a gain of 8 worked well. This brightens the image.

3.2 Gamma

Gamma

```
# http://scikit-image.org/docs/dev/api/skimage.exposure.html  
  
# Performs Gamma Correction on the input image. This function transforms  
# the input image pixelwise according to the equation  $O = I^{**}\text{gamma}$  after  
# scaling each pixel to the range 0 to 1.  
def adjust_gamma(image, gamma = 1, gain = 1):  
    return exposure.adjust_gamma(image, gamma=gamma, gain=gain);
```

Figure 5: Gamma Adjustment

Performs Gamma Correction on the input image. Also known as Power Law Transform. This function transforms the input image pixelwise according to the equation $O = I^{**}\text{gamma}$ after scaling each pixel to the range 0 to 1. [4] It can be used to adjust the brightness and sharpness of the image. After some experimenting, I found that a gamma of .68 worked well. This brightens the image.

3.3 Tint and Input Color

Tint and Input Color

```
# Add a color tint to the image and adjust the input color start.
# For example, if the color start is 20 for the red channel then
# it will take the current red values in the range of 0..255 and adjust
# them to be in the range 20..255
def tint(img, red_scale = 1, green_scale = 1, blue_scale = 1,
        red_input_color = 0, green_input_color = 0, blue_input_color = 0):

    tinted_img = img.copy()
    width, height, _ = np.shape(tinted_img)
    for x in range(width):
        for y in range(height):
            tinted_img[x][y][0] = (float(tinted_img[x][y][0]) * 255 / 255 + red_input_color) * red_scale
            tinted_img[x][y][1] = (float(tinted_img[x][y][1]) * 237 / 255 + green_input_color) * green_scale
            tinted_img[x][y][2] = (float(tinted_img[x][y][2]) * 167 / 255 + blue_input_color) * blue_scale

    tinted_img = img_as_ubyte(tinted_img)
    return tinted_img
```

Figure 6: Tint and Input Color Adjustment

This function adds a color tint to the image and adjusts the input color. For example, if the color input is 20 for the red channel then it will take the current red values in the range of 0..255 and adjust them to be in the range 20..255. In our case, we want to run the function with the tint ratio and color input values calculated previously in the methods section. This will give us something close to a maize-colored #F6D8AC with the appropriate input color values.

4 Results

The results are formatted as original, Instagram, and custom. Here are the results:

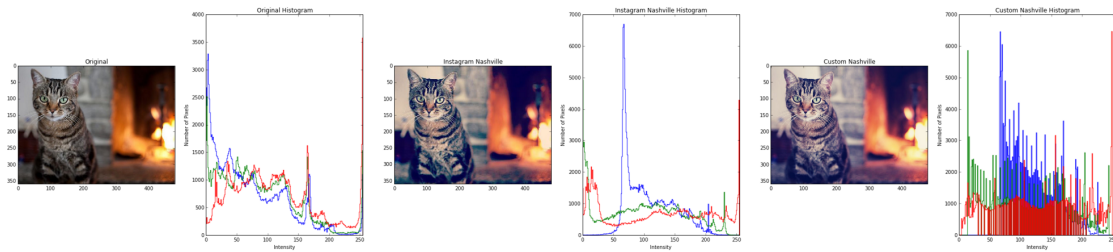


Figure 7: Result 1

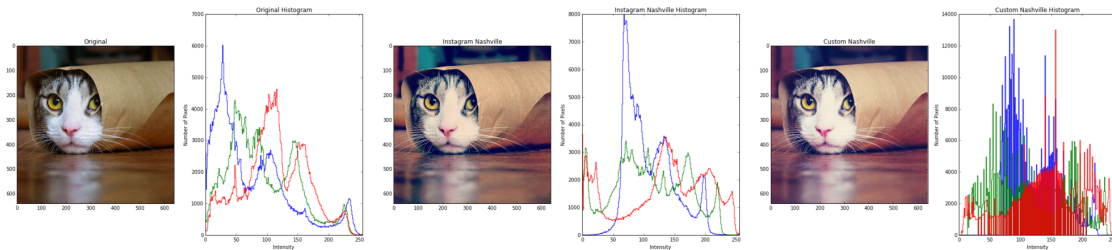


Figure 8: Result 2

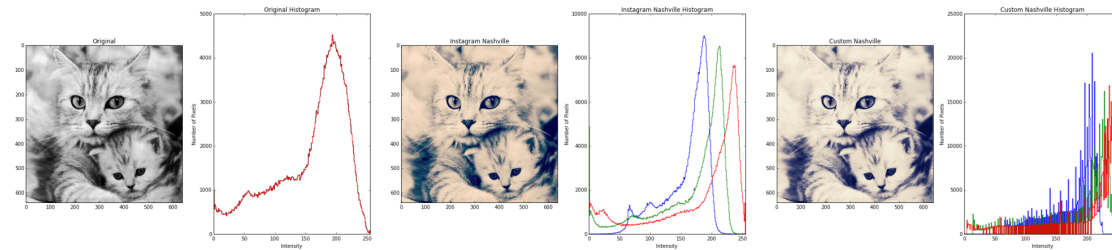


Figure 9: Result 3

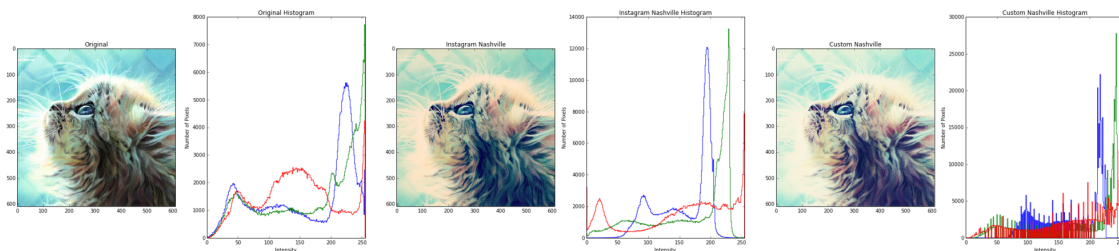


Figure 10: Result 4

5 Discussion

In conclusion, it was hard to get a perfect match with Instagram's Nashville filter. There's a lot more going on under the hood than what I have implemented. Even though my custom filter gets close, it is by no means an exact match. I hypothesize that Instagram does some color adjustments based on the intensity level. My tint and input level function is the same across the intensity range, and that makes for a much less complicated function.

References

- [1] Photoshop help / apply a brightness/contrast adjustment. <https://helpx.adobe.com/photoshop/using/apply-brightness-contrast-adjustment.html>.
- [2] Photoshop help / understand color adjustments. <https://helpx.adobe.com/photoshop/using/color-adjustments.html>.
- [3] Photoshop help / levels adjustment. <https://helpx.adobe.com/photoshop/using/levels-adjustment.html>.
- [4] Image processing in python using skimage exposure. <http://scikit-image.org/docs/dev/api/skimage.exposure.html>.
- [5] YOUNG, N. How to make instagram filters in photoshop: Nashville and 1997. <http://photodoto.com/instagram-filters-in-photoshop-nashville-1997/>, August 2014.