

You have 2 free stories left this month. [Sign up and get an extra one for free.](#)

Create Your Own Instagram Filters Using Python

Examples of image filters with code using OpenCV



Sabina Pokhrel

Oct 1, 2019 · 3 min read ★

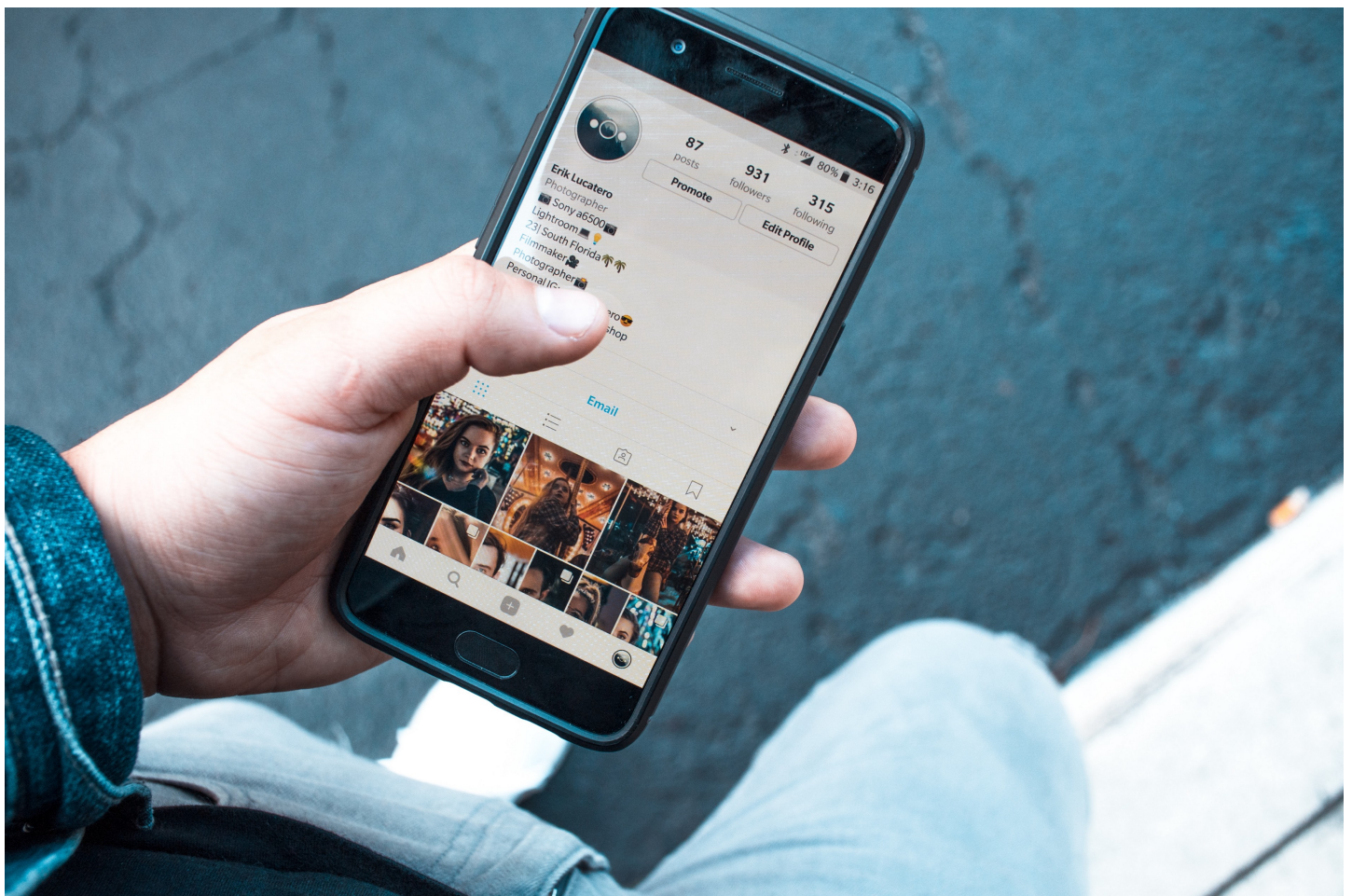


Photo by Erik Lucatero on Unsplash

I am a fan of Instagram filters. They are so cool. I almost always use them

because they are so handy, and with just a click of a few buttons, the photo I want to post is transformed.

Have you wondered if you can create one for yourself?

The answer is **YES YOU CAN!**

In this post, I will show you how you can create some image processing filters, with code and example images, to show how you can create your own filters using Python to transform any image.

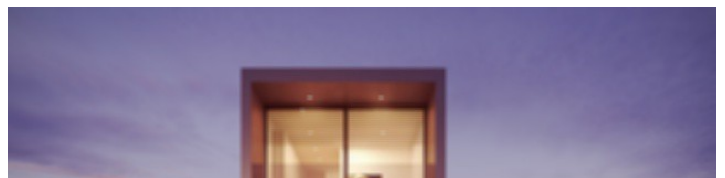
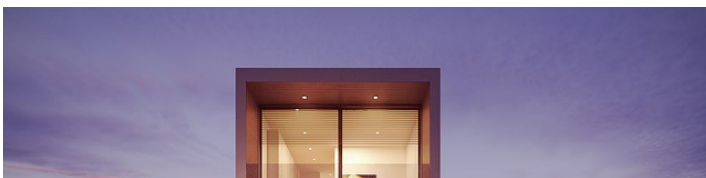
. . .

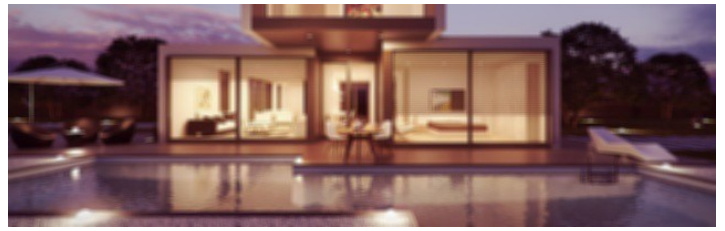
You will need to install the following python libraries if it is not already installed:

```
opencv-python  
matplotlib  
numpy
```

Blur Filter

```
import cv2  
import matplotlib.pyplot as plt  
  
im = cv2.imread('input-image.jpg')  
dst = cv2.GaussianBlur(im, (5, 5), cv2.BORDER_DEFAULT)  
  
plt.imshow(dst)  
plt.show()
```





(Left) Original image of a house (Image by giovanni gargiulo from Pixabay), (Right) Image after applying a blur filter



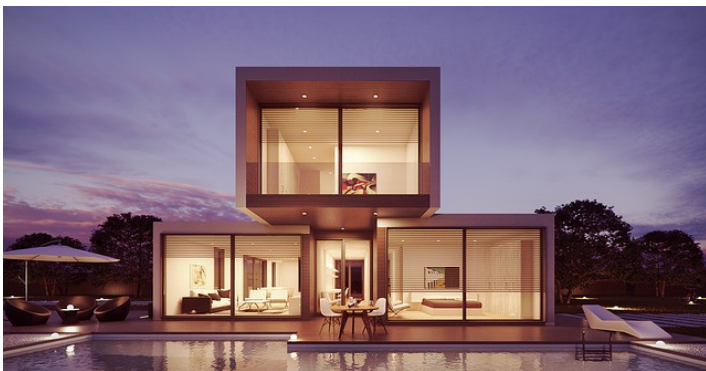
(Left) Original image of pomegranates (Image by Simon Matzinger from Pixabay), (Right) Image after applying a blur filter

Edge Detection Filter

```
import cv2
import matplotlib.pyplot as plt

im = cv2.imread('input-image.jpg')
edges = cv2.Canny(im, 100, 300)

plt.imshow(edges)
plt.show()
```





(Left) Original image of a house (Image by giovanni gargiulo from Pixabay), (Right) Image after applying an edge detection filter



(Left) Original image of a sunflower (Image by Capri23auto from Pixabay), (Right) Image after applying an edge detection filter

Vintage Filter

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

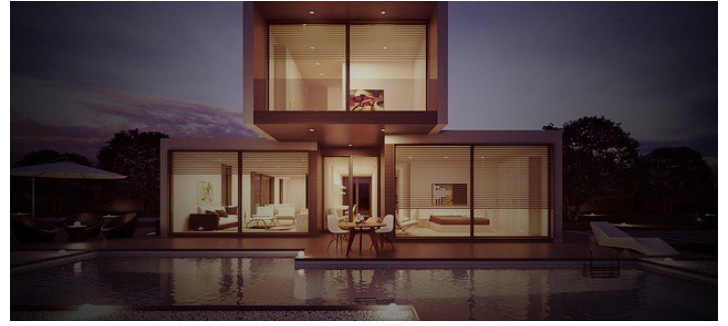
im = cv2.imread('input-image.jpg')
rows, cols = im.shape[:2]

# Create a Gaussian filter
kernel_x = cv2.getGaussianKernel(cols,200)
kernel_y = cv2.getGaussianKernel(rows,200)
kernel = kernel_y * kernel_x.T
filter = 255 * kernel / np.linalg.norm(kernel)
vintage_im = np.copy(im)

# for each channel in the input image, we will apply the
above filter
for i in range(3):
    vintage_im[:, :, i] = vintage_im[:, :, i] * filter

plt.imshow(vintage_im)
plt.show()
```





(Left) Original image of a house (Image by giovanni gargiulo from Pixabay), (Right) Image after applying a vintage filter



(Left) Original image of a sunflower (Image by Capri23auto from Pixabay), (Right) Image after applying a vintage filter

. . .

Image processing using Python and OpenCV was shown in this post with code and example images.

Which filter is your favourite? Leave your thoughts as comments below.

Click here to read my other posts on AI/Machine Learning.

. . .

Sources:

OpenCV

Open Computer Vision Library

opencv.org

OpenCV: Canny Edge Detector

Prev Tutorial: Laplace Operator Next Tutorial: Hough Line Transform
In this tutorial you will learn how to: Use the...

docs.opencv.org

Joshi, Prateek. *Opencv With Python By Example*.

[Python](#)[Image Processing](#)[Computer Vision](#)[Opencv](#)[Programming](#)[About](#) [Help](#) [Legal](#)

Get the Medium app

