Image Processing

Project title:

Converting images to pencil sketch

Presented by:

Mohamed Yousef Mohamed El-Sayed

Mohamed Alaa El-din AbdelRaouf

Presented to:

Dr/ Marwa Rafaei

Contents

Introduction

In this project, I implemented a system to generate pencil-drawing style images from photographs. The system has two main stages. Given a photo as input, it first generates a stroke layer to represent the shapes on the image, imitating painters sketching the contours. Then it produces tonal textures, imitating the hatching process when painters depict brightness and shades with pencils. Finally the two layers are combined to synthesize a nonphotorealistic pencil drawing.

This system helps people better understand how painters produce pencil drawings, which is one of the most fundamental pictorial languages to abstract human perception of natural scenes. Moreover, rendering real scene in a non-realistic, artistic way in many cases is desirable in game and film industry since artists have learned that by making images look less photorealistic they enable audiences to feel more immersed in a story. Computers thus help saving a lot of human labor as traditionally producing a non-realistic video segment would require artists drawing many pictures by hand. Finally, this kind of algorithms allow ordinary people to "become artists". Being able to transform common photos taken by themselves into unique arts would add much fun to people's life, making this system commercially valuable. [1]

1) Importing Modules:

Let's first start by importing modules into our program. We will be making use of the OpenCV functions and matplotlib module to plot the images.

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. [2]

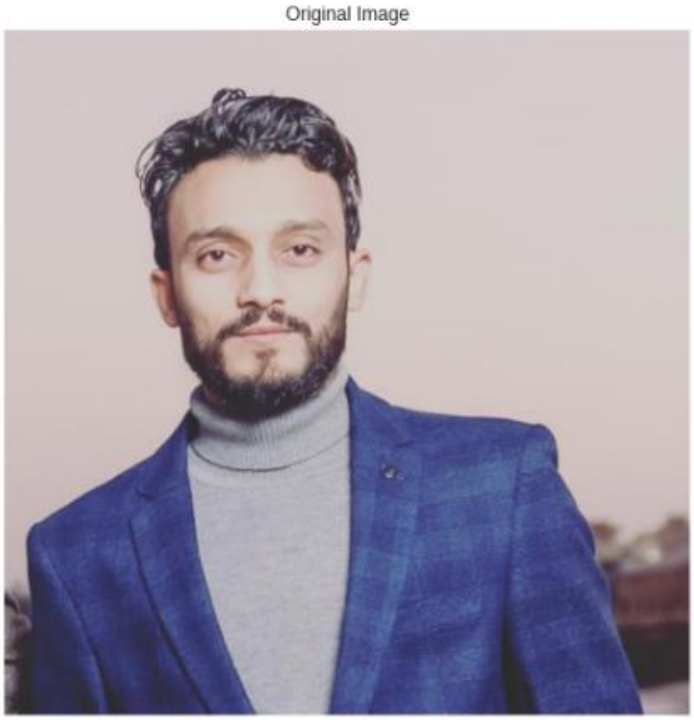
```
import cv2
import matplotlib.pyplot as plt
plt.style.use('seaborn')
```

2) Loading and Plotting Original Image:

Now we will be loading the image file into our program by making use of the imread function which takes the path of the file. Make sure your path to the file is correct.

We will also be changing the color code of the image to RGB format to get the original colors and plot the image using the imshow function.[2]

```
[7] img = cv2.imread("/content/1.PNG")
    img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
    plt.figure(figsize=(8,8))
    plt.imshow(img)
    plt.axis("off")
    plt.title("Original Image")
    plt.show()
```



3) Converting Image to GrayScale:

To decrease the complexity of the image and make the handling easier we will be converting the image to a grayscale image with the help of the cvtColor function. [2]

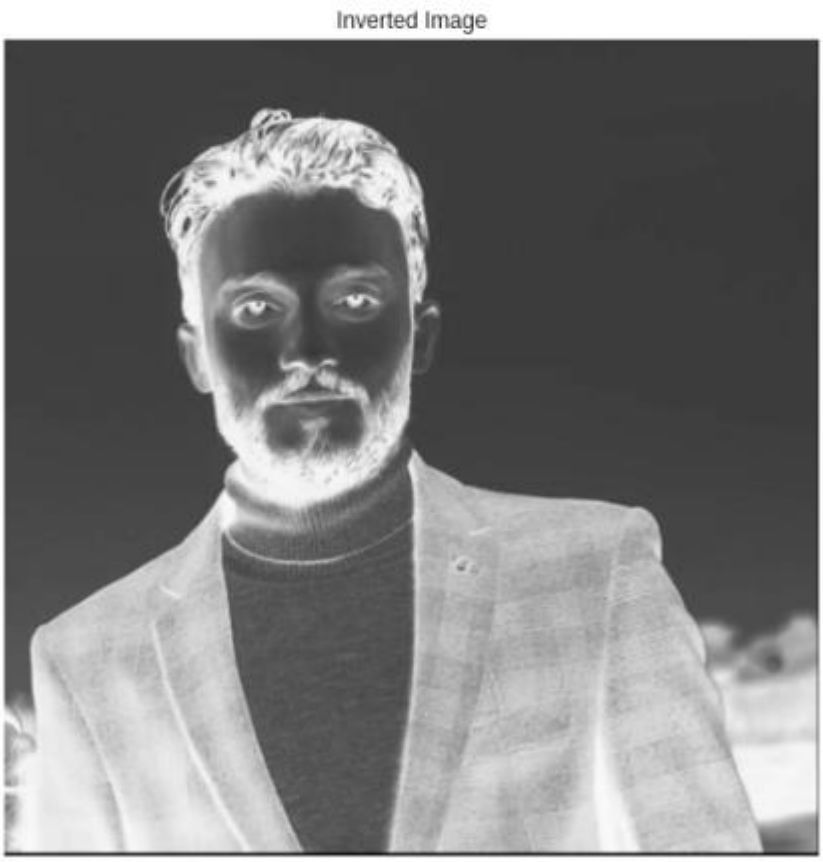
```
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.figure(figsize=(8,8))
plt.imshow(img_gray,cmap="gray")
plt.axis("off")
plt.title("GrayScale Image")
plt.show()
```

4) Inverting the Image:

Now the next step is to invert the image. Now your question would be why to do that? The answer to the same is that when a image is inverted it helps in studying the image with more precision and detail.

The more detailed study is important for the sketch generation as the sketches need to be accurate and good. [2]

```
img_invert = cv2.bitwise_not(img_gray)
plt.figure(figsize=(8,8))
plt.imshow(img_invert,cmap="gray")
plt.axis("off")
plt.title("Inverted Image")
plt.show()
```



5) Smoothing the image:

In addition to this, we will also be smoothing the image to make sure the sketch we obtain is less edgy and smooth.[2]

```
img_smoothing = cv2.GaussianBlur(img_invert, (21, 21),sigmaX=0, sigmaY=0)
plt.figure(figsize=(8,8))
plt.imshow(img_smoothing,cmap="gray")
plt.axis("off")
plt.title("Smoothen Image")
plt.show()
```

6) Converting your images to pencil sketches:

Now that the whole image processing is done, we will be passing the previous outputs to the function and pass the right and accurate parameters to make the appropriate changes to the image.[2]

```
[11]
    final = cv2.divide(img_gray, 255 - img_smoothing, scale=255)
    plt.figure(figsize=(8,8))
    plt.imshow(final,cmap="gray")
    plt.axis("off")
    plt.title("Final Sketch Image")
    plt.show()
```

7) Reference:

1) https://cs.stanford.edu/~yifengj/files/1505_npr.pdf

2) https://www.askpython.com/python/examples/images-to-pencil-sketch

Code:

```
import cv2
import matplotlib.pyplot as plt
plt.style.use('seaborn')
img = cv2.imread("/content/1.PNG")
img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
plt.figure(figsize=(8,8))
plt.imshow(img)
plt.axis("off")
plt.title("Original Image")
plt.show()

img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.figure(figsize=(8,8))
plt.imshow(img_gray,cmap="gray")
plt.axis("off")
plt.title("GrayScale Image")
plt.show()
img_invert = cv2.bitwise_not(img_gray)
plt.figure(figsize=(8,8))
plt.imshow(img_invert,cmap="gray")
plt.axis("off")
plt.title("Inverted Image")
plt.show()
img_smoothing = cv2.GaussianBlur(img_invert, (21, 21),sigmaX=0, sigmaY=0)
plt.figure(figsize=(8,8))
plt.imshow(img_smoothing,cmap="gray")
plt.axis("off")
plt.title("Smoothen Image")
plt.show()

final = cv2.divide(img_gray, 255 - img_smoothing, scale=255)
plt.figure(figsize=(8,8))
plt.imshow(final,cmap="gray")
plt.axis("off")
plt.title("Final Sketch Image")
plt.show()
plt.figure(figsize=(20,20))
plt.subplot(1,5,1)
plt.imshow(img)
plt.axis("off")
plt.title("Original Image")
plt.subplot(1,5,2)
plt.imshow(img_gray,cmap="gray")
plt.axis("off")
```

```
plt.subplot(1,5,3)
plt.imshow(img_invert,cmap="gray")
plt.axis("off")
plt.title("Inverted Image")
plt.subplot(1,5,4)
plt.imshow(img_smoothing,cmap="gray")
plt.axis("off")
plt.title("Smoothen Image")
plt.subplot(1,5,5)
plt.imshow(final,cmap="gray")
plt.axis("off")
plt.title("Final Sketch Image")
plt.show()
```

Output: