



Oil Painting project-Wenqi

💡 Inspiration:

1. Lack cv details in the CV group course (More relative to pictures instead of NLP)
 2. CV2 is interesting and fun to play with (Intel)
-

😊 Goal:

- made a module for the oil painting (packages)
- choosing the artistic type of painting
- the width of the brush
- Make use of OpenCV (a power library of computer vision) <https://opencv.org/>
- the practice of Python coding especially in Numpy and functions

Introduction:

In the field of computer vision, stylizing pictures is a very common application. What we will do today is to redesign an oil painting algorithm in consideration of various methods, and carry it out in multiple directions.

| Original pictures



style=sobel, width=2



| style=sobel, width=5



| style=scharr, width=5



Gradient (Edge detector):

For the direction of the brush, we use an edge detector to find the components in the X and Y directions.

We have four different edge detectors: **Prewitt, Roberts, Scharr, Sobel**

OpenCV tutorial:https://docs.opencv.org/3.4/d2/d2c/tutorial_sobel_derivatives.html



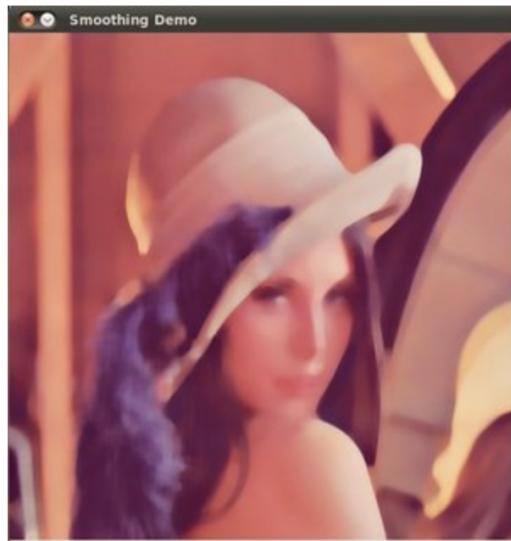
Smoothing:

Next, in order to make the picture look like a painting, we first applied a blur to the picture.

We have options of **Gaussian Blur, Bilateral Filtering, MedianBlur**. I try MedianBlur for better performance.

The algorithm behind smoothing:

https://docs.opencv.org/3.4/dc/dd3/tutorial_gaussian_median_blur_bilateral_filter.html



Draw Order (Make functions):

Originally, we painted from top to bottom, from left to right, but the effect was not satisfactory.

Therefore, we disrupt the order of coloring and randomly decide which plan to draw; and in this step, we decide the color of the stroke (skip some strokes), avoid repeated coloring, and speed up.

Define the function to make the random draw (**use of NumPy**)

Ellipse:

According to the determined order, the ellipse is used to represent the stroke; the length and angle of the ellipse

Refer to the Gradient of the first point (the hypotenuse of the X and Y components is used as the length, and the arctan is used as the angle degree

cv.ellipse https://docs.opencv.org/3.4/d6/d6e/group__imgproc__draw.html

😊 Optional and Bonus:

Make the notebook into a module :

Initiate it by python command line:

```
python main.py --brush_width 5 --path tulip.jpeg --gradient scharr
```

```
▼ parser = argparse.ArgumentParser()
parser.add_argument('--brush_width', default=5, help="Scale of the brush strokes")
parser.add_argument('--path', required=True, type=str, help="Target image path, gif or still image")
parser.add_argument('--palette', default=0, help="Palette colours. 0 = Actual color")
parser.add_argument('--gradient', default='sobel', help="Edge detection type. (sobel, scharr, prewitt, roberts)")
args = parser.parse_args()
main(args)
```

GIF to JPG to GIF:

The Unix command **convert** cuts GIF into JPG.



