

Project Report:

Image to Pencil Sketch Converter

Course: Vityarthi Project

Student Name: Aragya Shastri

Registration No.: 25BCE10396

Branch: B.tech CSE Core

Institution: VIT Bhopal University

Year: 2025

1. Introduction

Image to Pencil Sketch Converter is a computer vision project which is developed in Python, implementing the OpenCV library- cv2. This program takes any regular photographic image as input and generates an artistic, monochrome pencil sketch through a series of transformations. This project shows some basic concepts of digital image manipulation, especially those relating to color space, convolution, and emulating blend modes.

2. Problem Statement

It aims to achieve, through digital means, a scientific replication of the aesthetic appearance of a hand-drawn pencil sketch. A pencil sketch can be described by subtle shading, defined edges, and the absence of full color saturation. This calls for a translation of color and detail information from the original image into intensity variations (grayscale), mimicking lead-on-paper textures through a series of defined mathematical operations.

3. Functional Requirements and Code Workflow

This process of conversion is entirely encapsulated within the following successive steps, which are translations of the provided Python code:

ID	Function	OpenCV Method	Purpose
FR-01	Load Image	<code>cv2.imread()</code>	Reads the input image from the local file system.
FR-02	Grayscale Conversion	<code>cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)</code>	Reduces the image channels from three (Blue, Green, Red) to one (Intensity), losing color data.
FR-03	Inversion	<code>255-gray_image</code>	Inverts the grayscale image, turning light areas dark and dark areas light. This produces a "negative" image, which is the necessary input for the next step.
FR-04	Blur Inversion	<code>cv2.GaussianBlur(inverted_image, (21, 21), 0)</code>	Applies a Gaussian blur filter to the inverted image to smooth out details. The resulting image represents the "shading" layer.
FR-05	Final Inversion	<code>255 - blurred</code>	Inverts the blurred image back to normal tones, creating the inverted blurred image (similar to a Color Dodge blending mode input).

FR-06	Color Dodge Blend	<code>cv2.divide(gray_image, inverted_blurred, scale=256.0)</code>	The core mathematical step. Divides the original grayscale image by the inverted and blurred image, scaled by 256. This is the formula for the "Color Dodge" blend mode, which highlights light areas and increases contrast, yielding the final sketch effect.
-------	-------------------	--	---

4. Methodology and Digital Image Processing Techniques

The core methodology of the project embodies some of the usual techniques of image processing, including:

4.1. Grayscale Conversion (Step 2)

All color information is jettisoned, leaving only luminosity data behind. This is necessary because pencil sketches are, in their very nature, monochromatic and rely on the variation of light intensities to provide form and texture.

4.2. Image Inversion (Steps 3 & 5)

Inversion plays an important role in both the blurring and blending stages. The first inversion (Step 3) produces the pencil "negative." The second inversion (Step 5) forms part of the final blending formula.

4.3. Gaussian Blurring (Step 4)

A Gaussian filter is a low-pass filter used in order to reduce noise and detail in an image. In this case, the blur removes high-frequency elements - sharp edges and fine texture - from the inverted image. When this blurred layer is used in the final blend, the lack of detail serves to help simulate the smooth texture of shading pencil work. The kernel size is set at (21, 21) to control the degree of blurring.

4.4. Color Dodge Blend (Step 6)

The last step mathematically emulates the "Color Dodge" blend mode common in graphic design software; the formula used is:

$$\text{Sketch} = \frac{\text{Grayscale}}{\text{Inverted and Blurred Image}} \times 256,$$

This division function brightens and reduces the contrast of the base image where the inverted blurred layer is dark. The result strongly accentuates the edges and emphasizes shading to create the characteristic appearance of a pencil sketch.

5. System Requirements and Execution

- **Language:** Python 3.x
- **Library:** OpenCV (cv2)
- **Execution:** The program is executed from the command line and requires the target image, tiger.jpg or whichever image you decide to use, to be in the same directory in which the script resides. This script opens multiple windows successively to show the intermediate and final images.

Required Setup

```
# Install the OpenCV library  
pip install opencv-python
```

6. Conclusion and Outcome

The Image to Pencil Sketch Converter successfully demonstrates one nontrivial application of the principles of computer vision. This code effectively converts a very complicated photographic input into a minimalist artistic output by successively applying grayscale conversion, inversion, Gaussian blurring, and the Color Dodge blend. One can understand how such basic matrix operations-like division and subtraction-can be combined in such a way as to yield these kinds of sophisticated visual effects.

7. Contact Information

- **Name:** Aragya Shastri
- **Registration No.:** 25BCE10396
- **Email:** aragya.25bce10396@vitbhopal.ac.in
- **Institution:** VIT Bhopal University