SVG GENERATOR

Group: Mohg, Lord of Blood

Olivia, Joy, Sergio, and Taylor

Overview

SVG Image Outline Generator

- A generator that transforms images into clean SVG outlines using edge detection!
- Ideal for laser engraving projects such as stamps, custom artwork, or detailed designs.
- Adjustable sliders allow users to fine-tune the clarity and precision to achieve the ideal result for any image.

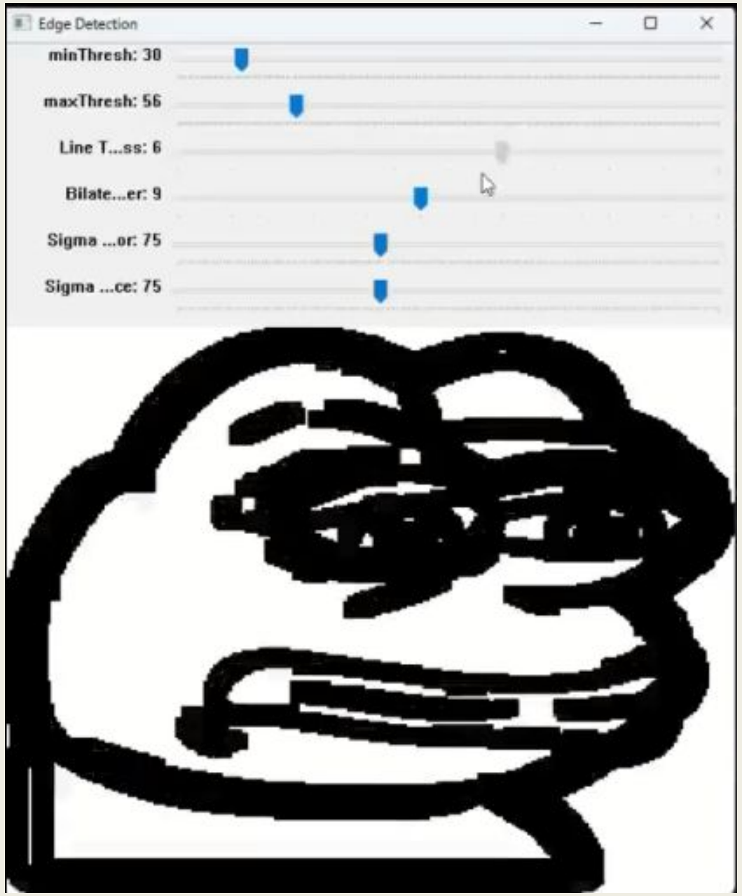
Background

Original Goal: Stencil Generator

- Create stencils through continuous edge detection.
- Eventually decided to focus on general laser engraving since it was more realistic for achieving a well-defined finished product.
- This shift expanded the range of projects that can be created using a laser cutter, including stamps, detailed engravings, decorative designs, custom signage, and various artistic or functional applications.

Mobile and desktop applications

Our project features both a desktop and an Android mobile interface, ensuring accessibility and usability across platforms!



minThresh: 30

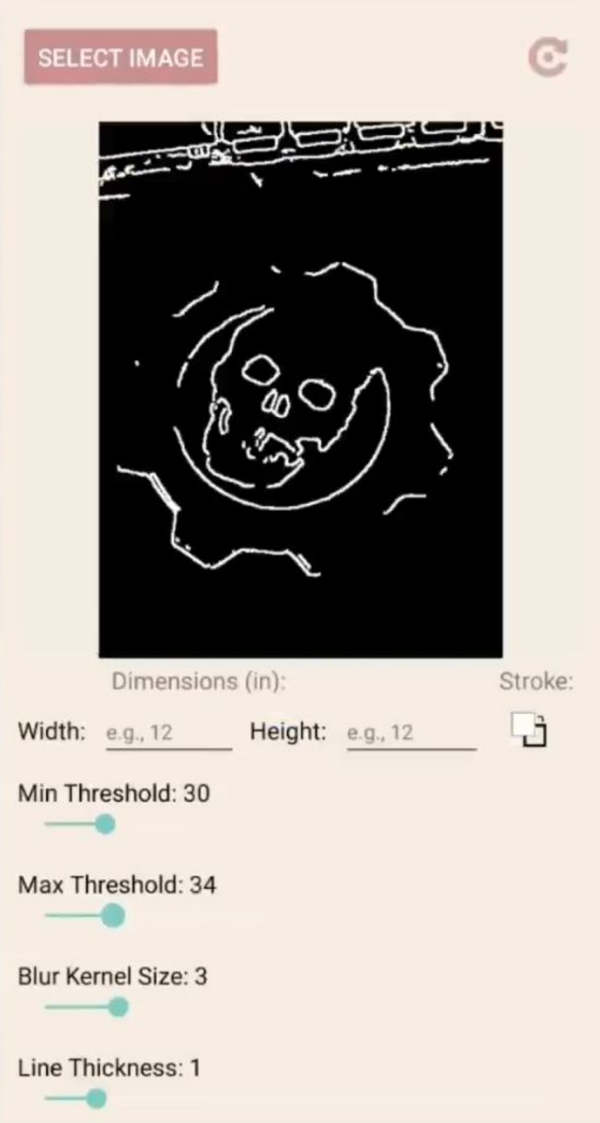
maxThresh: 56

Line T...ss: 6

Bilate...er: 9

Sigma ...or: 75

Sigma ...ce: 75



Width: e.g., 12 Height: e.g., 12

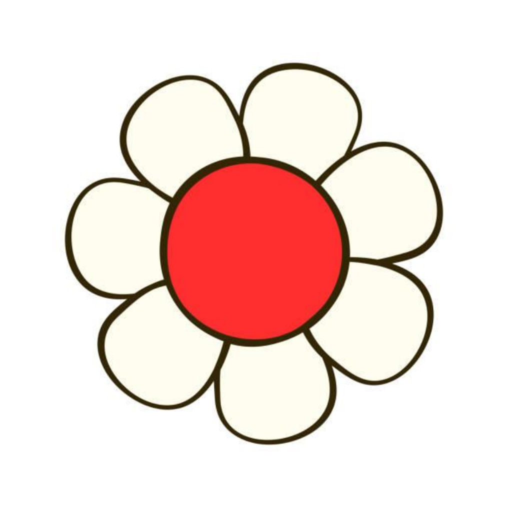
Min Threshold: 30

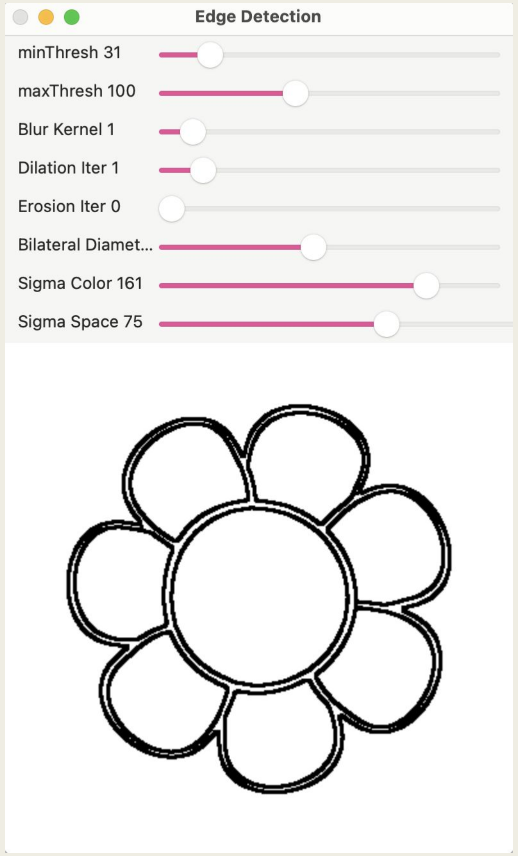
Max Threshold: 34

Blur Kernel Size: 3

Line Thickness: 1

A simple image example





Edge Detection

minThresh 31

maxThresh 100

Blur Kernel 1

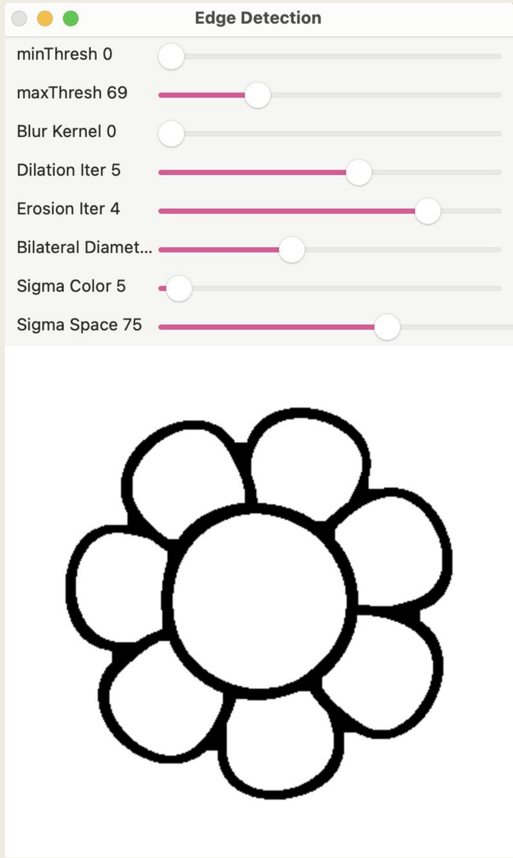
Dilation Iter 1

Erosion Iter 0

Bilateral Diamet...

Sigma Color 161

Sigma Space 75



Edge Detection

minThresh 0

maxThresh 69

Blur Kernel 0

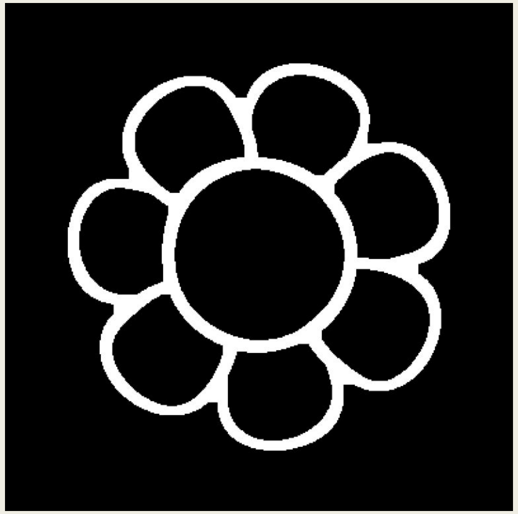
Dilation Iter 5

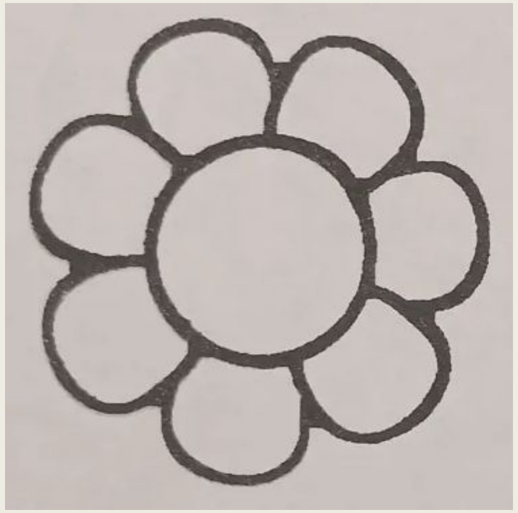
Erosion Iter 4

Bilateral Diamet...

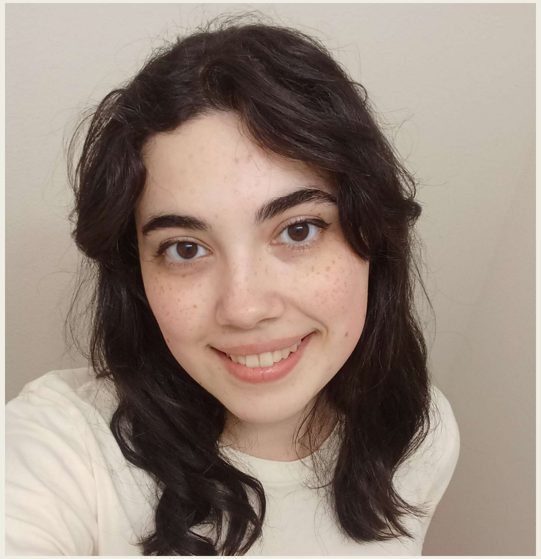
Sigma Color 5

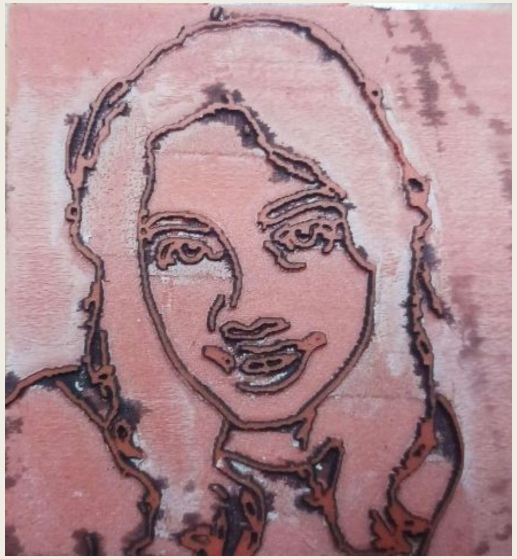
Sigma Space 75

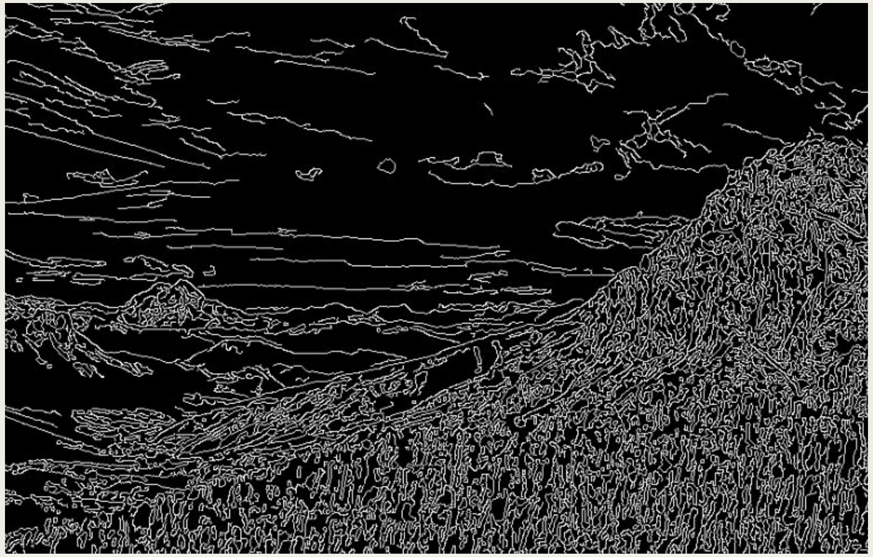


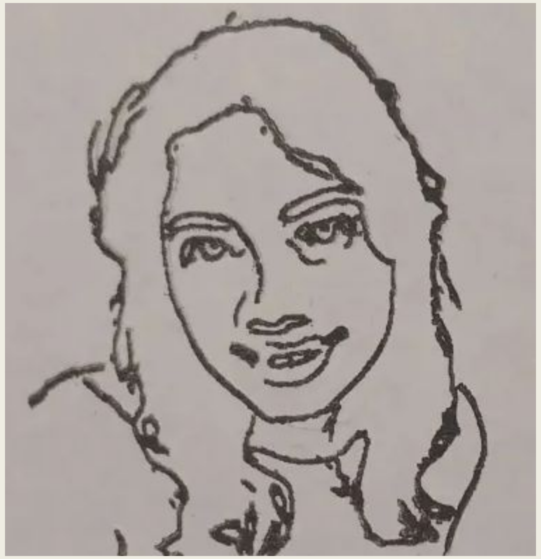


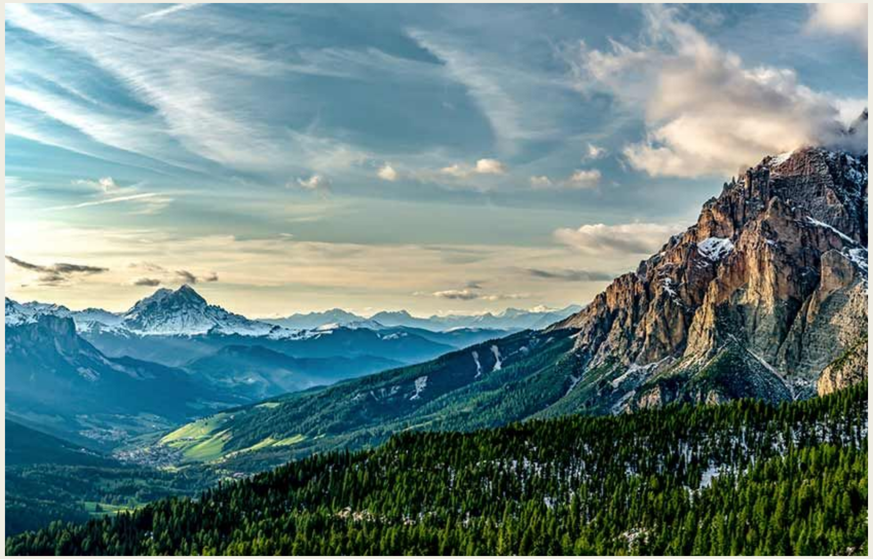
Complex image examples







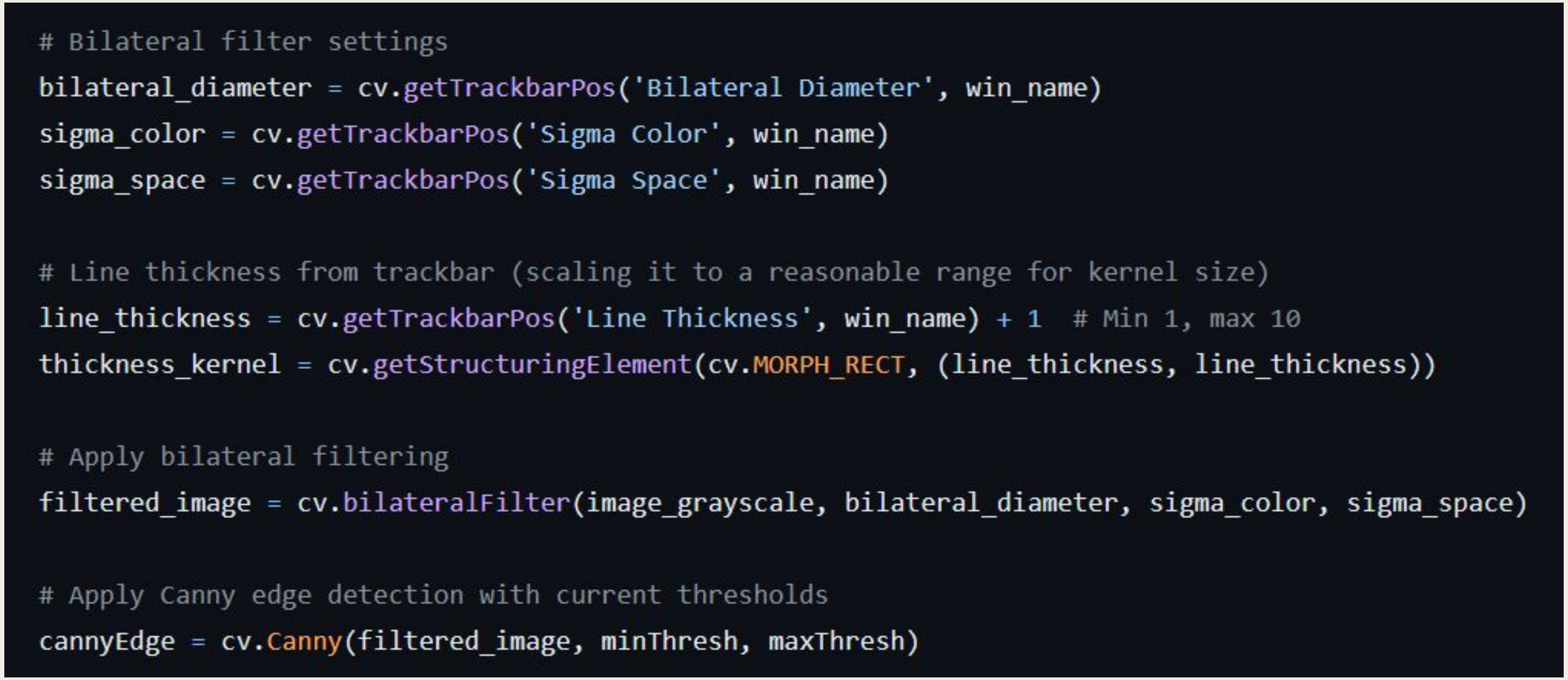




Methodology

Edge Detection

- convert image to grayscale, apply a bilateral filter, and use canny edge detection

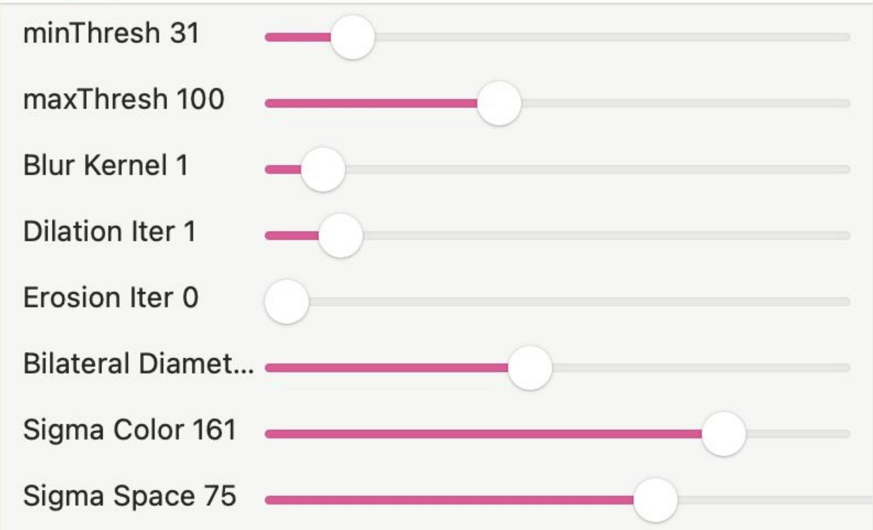


```
# Bilateral filter settings
bilateral_diameter = cv.getTrackbarPos('Bilateral Diameter', win_name)
sigma_color = cv.getTrackbarPos('Sigma Color', win_name)
sigma_space = cv.getTrackbarPos('Sigma Space', win_name)

# Line thickness from trackbar (scaling it to a reasonable range for kernel size)
line_thickness = cv.getTrackbarPos('Line Thickness', win_name) + 1  # Min 1, max 10
thickness_kernel = cv.getStructuringElement(cv.MORPH_RECT, (line_thickness, line_thickness))

# Apply bilateral filtering
filtered_image = cv.bilateralFilter(image_grayscale, bilateral_diameter, sigma_color, sigma_space)

# Apply Canny edge detection with current thresholds
cannyEdge = cv.Canny(filtered_image, minThresh, maxThresh)
```

Methodology

Dimensions and Parameter Fine-Tuning

- Users can specify the width and height of their final SVG.
- Trackbars allows user to change edge-detection settings on a per-image basis.



minThresh 31

maxThresh 100

Blur Kernel 1

Dilation Iter 1

Erosion Iter 0

Bilateral Diamet...

Sigma Color 161

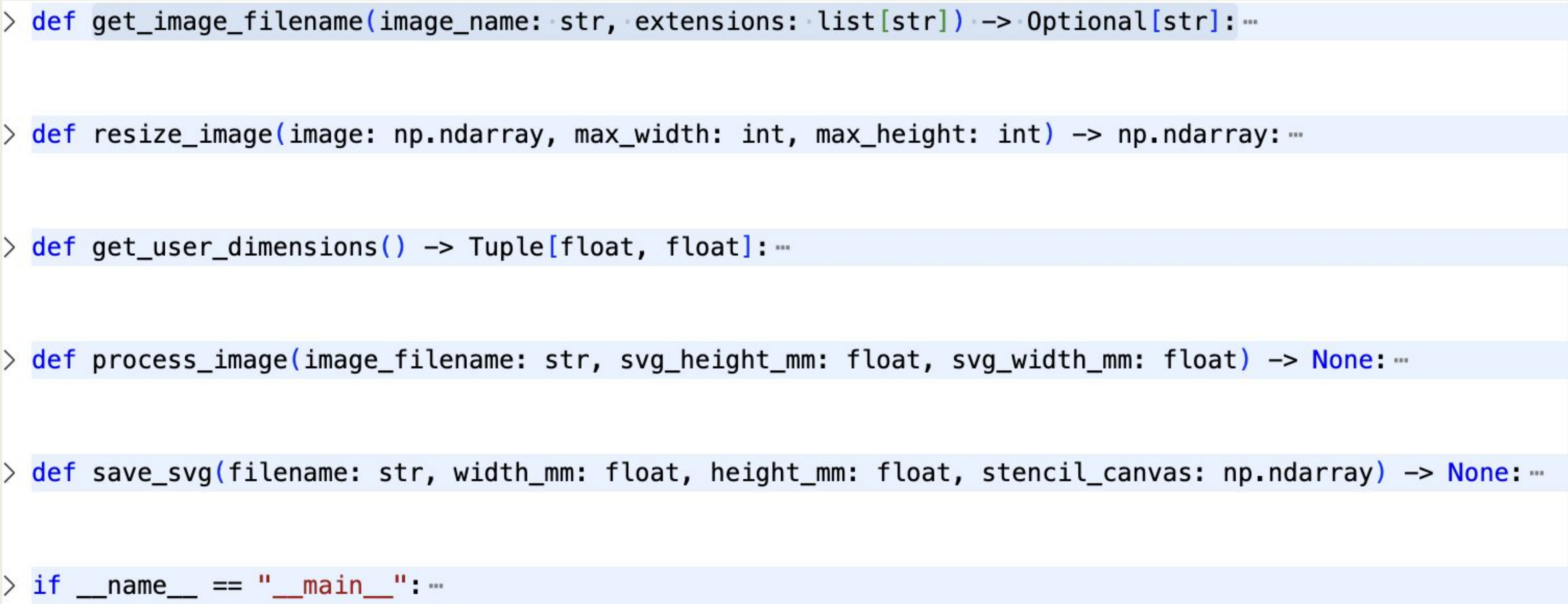
Sigma Space 75

Methodology

Understanding SVGs and Using a Laser Cutter

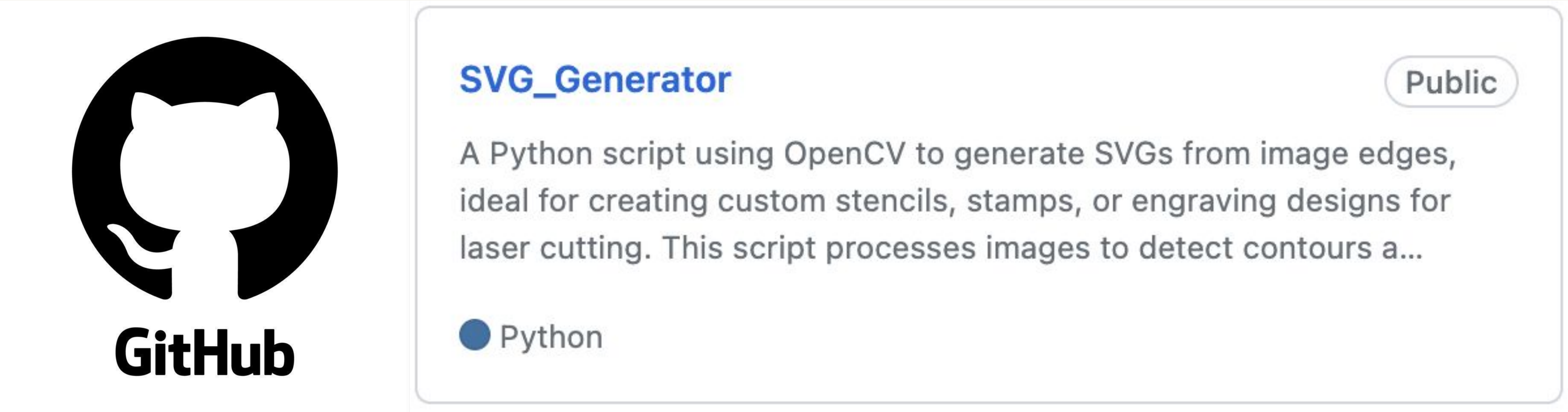
- Gained an understanding of how laser engraving works and how to create effective SVG files for laser cutter applications.
- Learned to apply colors and borders to guide the laser cutter effectively.
- Laser cutter training at Alkek Makerspace
  - Tested various machine settings to find a balance that consistently produced detailed stamps and engravings on different materials.

Code Implementation



```
def get_image_filename(image_name: str, extensions: list[str]) -> Optional[str]:⋯

def resize_image(image: np.ndarray, max_width: int, max_height: int) -> np.ndarray:⋯

def get_user_dimensions() -> Tuple[float, float]:⋯

def process_image(image_filename: str, svg_height_mm: float, svg_width_mm: float) -> None:⋯

def save_svg(filename: str, width_mm: float, height_mm: float, stencil_canvas: np.ndarray) -> None:⋯

if __name__ == "__main__":⋯
```

GitHub



SVG_Generator

Public

A Python script using OpenCV to generate SVGs from image edges, ideal for creating custom stencils, stamps, or engraving designs for laser cutting. This script processes images to detect contours a...

● Python

https://git.txstate.edu/omm19/SVG_Generator

Demo

- Play Me