**Thermal Image Processing Documentation**

**Introduction**

This Python script demonstrates thermal image processing using the OpenCV library. The code downloads an image from a specified URL, converts it to grayscale, simulates a thermal image using a color map, and performs additional image processing techniques.

**Libraries Used**

**cv2 (OpenCV)**: Used for image processing.

**requests:** Used to download images from the web.

**numpy:** Used for numerical operations on images.

**BytesIO:** Used to handle binary data.

**google.colab.patches.cv2\_imshow:** Used to display images in Google Colab.

**Functionality**

process\_thermal\_image(image\_url)

This function takes the URL of an image as input, downloads the image, and performs the following steps:

**Download and Load Image:**

Downloads the image from the specified URL.

Decodes and loads the image using OpenCV.

**Grayscale Conversion:**

Converts the loaded image to grayscale.

**Thermal Image Simulation:**

Applies a color map (COLORMAP\_JET) to simulate a thermal image.

**Gaussian Blur:**

Applies Gaussian blur to the grayscale image.

**Thresholding:**

Thresholds the blurred image to create a binary image.

**Contour Detection:**

Finds contours in the binary image.

**Contour Visualization:**

Draws contours on the original image.

**Image Display:**

Displays the original, grayscale, thermal, blurred, binary, and contours images using cv2\_imshow.

**Usage**

Replace the image\_url variable with the URL of the desired image.

**python**

Copy code

image\_url = 'https://example.com/your-image.jpg'

Run the script in a Python environment with the required libraries installed.

Observe the displayed images for each processing step.

**Example**

The provided example uses a landscape image from iStockphoto to showcase the thermal image processing. You can replace the image\_url with the actual URL of an image you want to use.

**Screenshots:**









