

The following are the packages used:

1. Pillow - Used for transforming the input image to grayscale and then to assign colours blue and orange to black and white respectively,
2. opencv-python - Used for edge estimation and merging the edges with the re-colored image.
3. numpy - Used for the conversion of a PIL image (RGB format) to cv2-compatible image (BGR format)

User Input Required:

1. Location of the input images: If the user happens to copy from the location from the Windows Desktop, the separator must be changed “/” from “\”
2. Name of the folder to store the output images

Description of the Implementation

1. The code first asks the user to provide the location of the input images which have to be transformed. The images within the folder are then obtained.
2. Given that the folder contains both input and target images, the input images are separated by searching for the string “_input” within the file names. However, if there is a separate file with all of the input images, the corresponding code is enclosed within the comments. The user is then asked for the name of the folder that will contain the processed images.
3. The input images are then processed using a “for loop”. Every image is converted into grayscale using the PIL package. Then the colours black and white are assigned to the colours blue and orange respectively.
4. To enable Edge Detection using Canny, the PIL images in RGB format are converted to BGR format using numpy commands.
5. The image obtained in the previous step is then converted to grayscale. To enable a better “Edge Detection”, the image is subject to median blurring.
6. To enable Dynamic Setting of the thresholds for edge detection, the function *get_auto_edge* is defined which takes the grayscale cv2 image as the input and returns the threshold values.
7. After the edge detection, the grayscale image is converted to BGR format so that the edges can be overlaid on the transformed image.
8. Finally, the detected edges obtained in step 6 and the transformed image obtained in step 4 are combined to get the final image. This is then written to the folder provided by the user in Step 2. The string “_input” in the name of the original image is replaced by the string “_output”. If the names of the files do not adhere to this format, an alternative is mentioned within the comments.