Hello,

Upon initial examination of the task, my first idea was to use a neural network for image reconstruction. For this purpose, I turned to a simple implementation of the UNET network. UNET is typically used for segmentation, but in my previous experience, I have used this network for tasks such as removing unnecessary portions of an image and improving image quality.

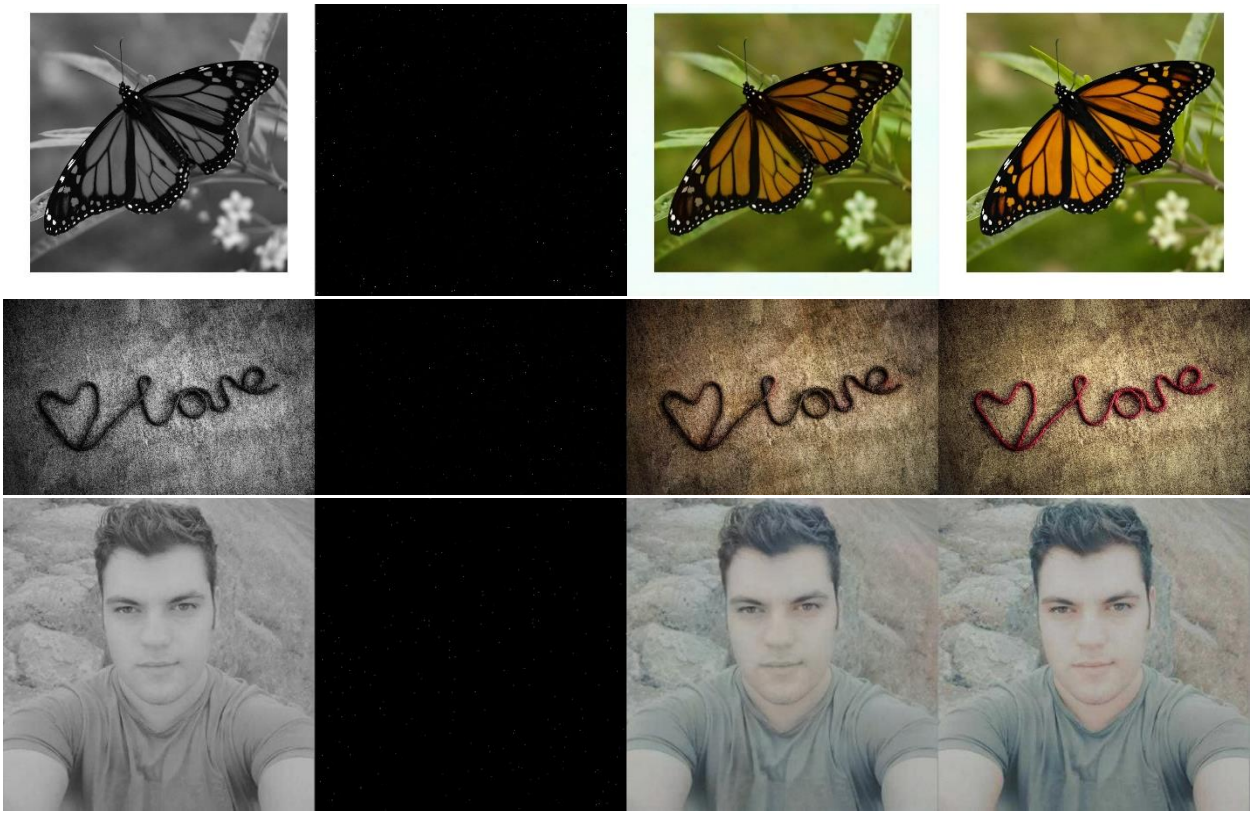
The normal input for the UNET network is a 3-channel image, and the output is a 1-channel image, which is the output of the segmentation network. To perform simple colorization using this network, we can change the output channel to 3, which creates a colored version of the image. However, to perform conditional colorization while having access to only 0.1% of the actual image pixels, I considered the input channel to be 4, which concatenates a grayscale image with a color image with 0.1% of the actual image pixels, and the output channel is a 3-channel image that produces a colorized version of the grayscale image.

To train this network, I used the COCO dataset and wrote a customized Data Loader specifically for this purpose. The output of this Data Loader is a grayscale image, 0.1% randomly selected pixels from the actual image, and the actual image itself as the desired label for the grayscale image.

Due to a lack of access to a GPU, I was unable to train the network extensively, so the implemented network was trained for only 2 epochs.

Outputs :

Gray Scale Image	**Random Pixel**	**Output**	**Ground Truth**



[Ali Fayzi]