XCI-Sketch: Extraction of Color Information from Images for Generation of Colored Outlines and Sketches

V Manushree * **Sameer Saxena*** **Parna Chowdhury*** **Manisimha Varma***
Harsh Rathod* **Ankita Ghosh** **Sahil Khose**

Manipal Institute of Technology, Manipal
{ manushree.v, sameer.saxena, parna.chowdhury, manthena.varma, harsh.rathod, ankita.ghosh1, sahil.khose }@learner.manipal.edu

1 Introduction

In recent years, image processing and deep learning techniques have been extensively deployed to leverage images to generate sketches, but these sketches have been limited to black and white shades [8, 4, 2]. Often while using grayscale sketches, the overall expressivity falls short, therefore adding color elevates the sketch and efficiently carries across its meaning, background details, and connotation. Our paper focuses on different approaches to convert photographic pictures into colored sketches. We use the Contour Drawing Dataset presented in the work by Li [6] and propose two ways to extract color information from the images and amalgamate it with the corresponding sketches. For the first method, we formulate a process to transfer color onto the existing black and white sketches in the dataset to produce colored outlines. In the second method, colorspace manipulated sketches are used as the training dataset for a generative adversarial network and develop a model which can produce colored sketches from unseen images.

2 Rendering Colored Outlines in Sketches

In this section, we discuss how we transfer color to the black and white sketches of the Contour Drawing Dataset. A digital RGB image can contain $256 \times 256 \times 256$ different colors, but generally, only a few colors are used when an individual draws a sketch. To mimic this while generating colored outlines, we perform quantization [1] of the number of colors in the images. First, we perform three iterations of Gaussian blur [3] to reduce the sharp color transitions inside the images. To achieve quantization, we perform k-means clustering [9] on the image where the pixels are categorized into k clusters according to their intensity value. We know the k-means algorithm clusters data by minimizing the sum of squared distance within the cluster, so to determine this value k, we appoint this criterion, called inertia, as a threshold value. Inertia can be calculated as:

$$\sum_{i=0}^{n} \|c_i - x_i\|^2 \tag{1}$$

where x denotes pixels, x_i denotes value of ith pixel and c_i denotes value of cluster centroid closest to x_i. A threshold value (τ) is fixed at the beginning. We then iterate over values of k for an image at strides of 5 and calculate the inertia at each step. The value of inertia starts at infinity and reduces as the number of clusters increase. We obtain the optimal k value for an image at the point where the inertia becomes lesser than the assigned threshold value. For locating the ideal

*Authors have contributed equally to this work and share first authorship. Link of the code: https://github.com/Sampai28/GeneratedSketches

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

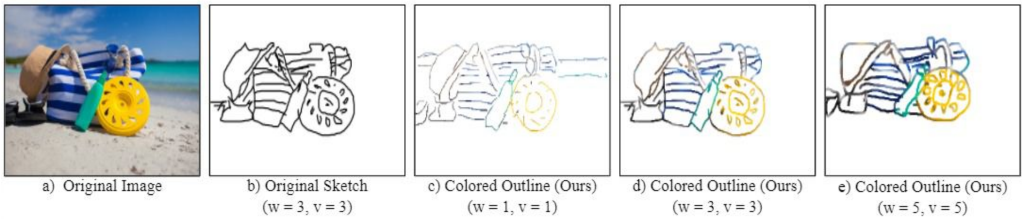

Figure 1: Qualitative results of the rendered colored outlines. w is the stroke width and v is the version of sketch.

threshold value, we test various values empirically and conclude that when the threshold is 70, it gives visually pleasing results. After performing k-means clustering on the images, we apply binary thresholding on the sketch and extract the black outlines to form a mask. We split the channels of the post-processed image and perform bitwise operations between the mask and each channel to obtain the color information. The resultant channels are merged to render the final colored outline sketch.

3 Generating Colored Sketches

In this section, we introduce a generative adversarial network framework that converts input images to colored sketches. To add color to the sketches, we propose a technique similar to the Gouache Color Transform stated in [10]. We convert the RGB image and its sketch to the $L^*a^*b^*$ colorspace.The a^* and b^* channels of the sketch are discarded and replaced with the a^* and b^* channels of the corresponding image to transfer color without changing the content. The resultant image is converted back to RGB. To improve the quality of the transferred colors, we increase the saturation by a factor of 1.8 in HSV colorspace.

These newly obtained colored sketches are included in the training data for our GAN model. Our GAN learns a mapping from an input image x to an output colored sketch y, so that $G : x \to y$. We adapt our generator G and discriminator D from the pix2pix work of Isola [5], where the generator resembles the architecture of U-Net [7], which consists of an encoder and decoder network with skip connections. The discriminator is a PatchGAN which penalizes the structure at the scale of local image patches. We optimize the GAN with additional convolutional layers where we use a kernel size of 3×3, apply same padding, and set the stride to 1. This is followed by batch normalization and a leaky ReLU activation function at every downsampling layer of the encoder. The feature space is also increased so that more attributes can be extracted. We apply similar optimizations in the discriminator and introduce max-pooling layers alongside to extract necessary features which can help the discriminator differentiate between real and fake inputs. While passing input data to the discriminator, the post-processing colored sketches are considered real, while the output from the generator is regarded as fake. The original image is stacked upon the sketches before passing them to the discriminator. We train our GAN with the data resized to 256×256 dimensions followed by normalization with mean and standard deviation set to 0.5. To optimize our objective loss function, we follow the standard approach as mentioned in [5], where we oscillate between one step of gradient descent on D and one step of gradient descent on G.The objective loss function, is divided by the batch size, which is 32 in our case. We used a Adam optimizer with a learning rate of 0.0005. The momentum parameters are defined as $\beta_1 = 0.5$, $\beta_2 = 0.999$, and λ is set to 1000.

Figure 2: Sketches generated by our GAN from unrevealed images spanning over a variety of data distribution.

Ethical Implications

The dataset on which the generation model has been trained is biased by certain objects and landscapes, which influence the colors and outlines in the generation of sketches when applied on wide variety of images. The sketches produced by our model attempt to mimic the content of the image at an elementary level. However, assessment of sketches is subjective for every user depending on their aesthetic and expertise but we attempt to quantify their evaluation through a perceptual user study as shown in Figure 5.

References

[1] M. Emre Celebi. Improving the performance of k-means for color quantization. *Image and Vision Computing*, 29(4):260–271, 2011.

[2] Songwei Ge, Vedanuj Goswami, Larry Zitnick, and Devi Parikh. Creative sketch generation. In *International Conference on Learning Representations*, 2021.

[3] Estevão S. Gedraite and Murielle Hadad. Investigation on the effect of a gaussian blur in image filtering and segmentation. In *Proceedings ELMAR-2011*, pages 393–396, 2011.

[4] David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.

[6] Mengtian Li, Zhe Lin, Radomír Mˇ ech, Ersin Yumer, and Deva Ramanan. Photo-sketching: Inferring contour drawings from images. In *WACV*, 2019.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[8] Saul Simhon and Gregory Dudek. Sketch interpretation and refinement using statistical models. In *Proceedings of the Fifteenth Eurographics Conference on Rendering Techniques*, EGSR'04, page 23–32, Goslar, DEU, 2004. Eurographics Association.

[9] Kristina P. Sinaga and Miin-Shen Yang. Unsupervised k-means clustering algorithm. *IEEE Access*, 8:80716–80727, 2020.

[10] Wei Zhang, Guanbin Li, Haoyu Ma, and Yizhou Yu. Automatic color sketch generation using deep style transfer. *IEEE Computer Graphics and Applications*, 39(2):26–37, 2019.

A Example colored outlines and sketches from dataset and generation

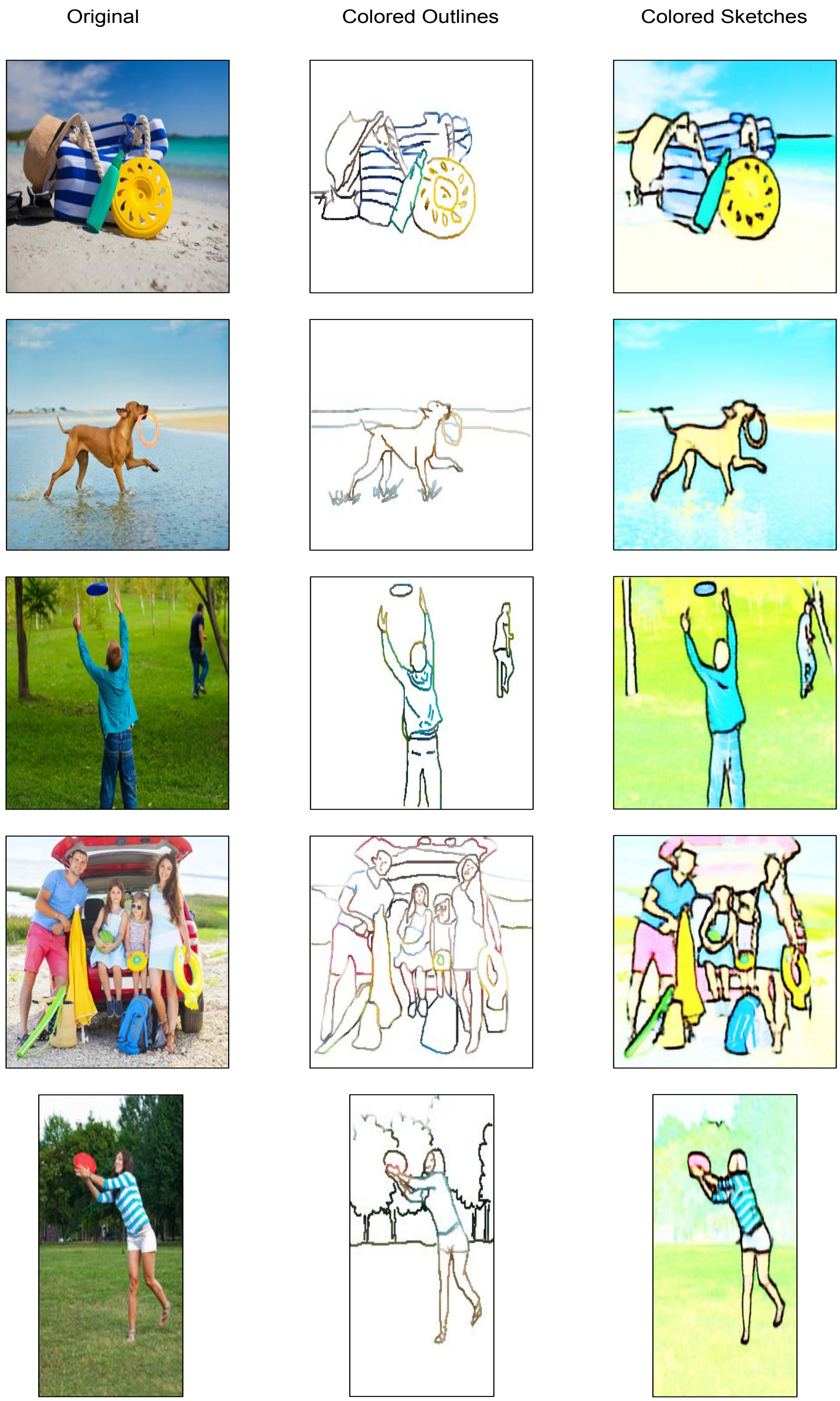

Figure 3: Qualitative results of our model

B Pipeline for Generating Colored Sketches

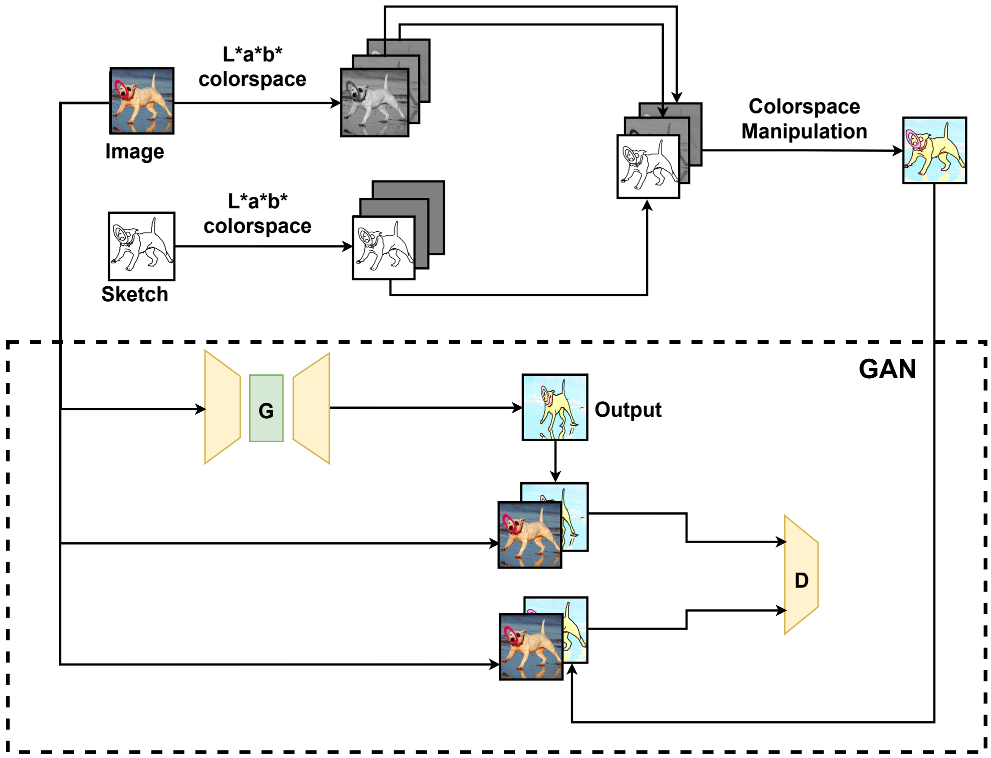

Figure 4: Pipeline of our proposed generation model. We use colorspace manipulation to extract color information for the images and amalgamate it into black and white sketches. We use these colored sketches to train our GAN.

C Evaluation

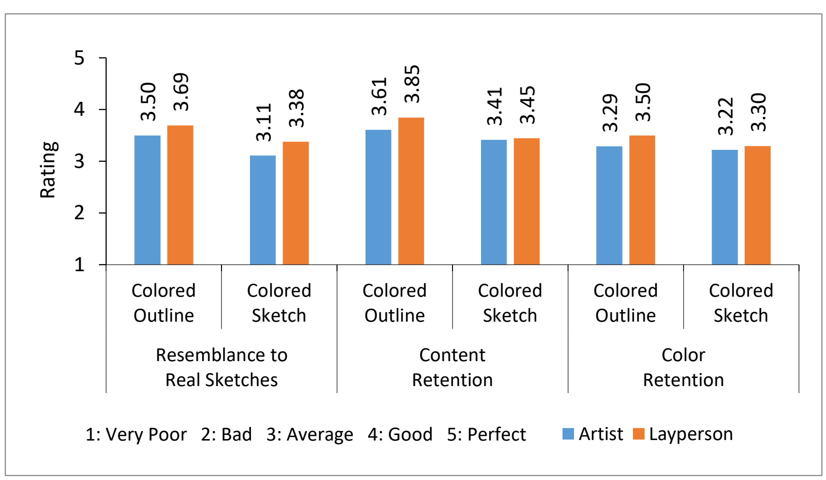

Figure 5: Results of our perceptual user study, for each category rated on a scale of 1 to 5, collected from 30 artists and 70 laypeople.