# Automatic Colorization of Line Sketch Images

Lilly Kumari, Narendra Shivaraman

University of Washington

## 1 Problem Statement

Automatic colorization of outline images has potential applications in areas ranging from image compression to digital art. In this project of image-to-image translation, we aim to generate acceptable and visually appealing colorized images given an outline image.

## 2 Dataset Used

We use the UT Zappos50K shoes 256x256x3 image dataset [1] [2], along with their line sketch images. Because of computational memory limitations, we have downsized the images to 128x128x3. The training set comprises of 48,025 image pairs while the validation set comprises of 200 image pairs. The dataset is uniform with respect to object orientation, luminance, and background color.

## 3 Methodology

In order to generate colorized version of outline shoes, we train two different models- a **Convolutional Neural Network** (CNN) and **Conditional Generative Adversarial Network** (cGAN). The architecture of both models are shown in the poster. The colorization task is treated as a regression problem and in both models, we experiment with three different losses: **L1, L2 and Huber loss** (alongwith the GAN loss in case of cGAN model).

The idea behind using two diversified models was to observe the difference in their performance wrt our task at hand. While the CNN model mostly colored everything in a similar fashion without making much distinction between dense edges, the cGAN models does well by coloring the shoes more vividly while recognizing the edges well.

## 4 Qualitative Results

Figure 1 shows some examples of colorized images generated using different model and loss combination.

## 5 Quantitative Results

We use structural similarity (SSIM) as a measure of performance as it takes into account the contrast, luminance and structure of the images [3]. The SSIM of the generated colorizations are evaluated with respect to the original images.

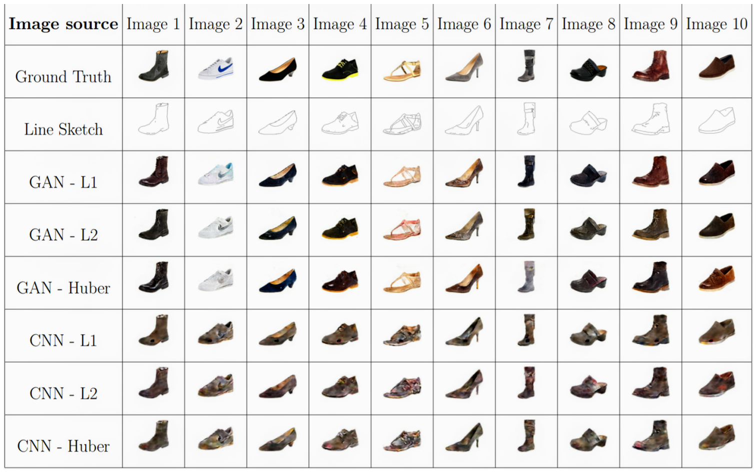


Figure 1: Generated Colorized Images

$$SSIM(x,y) = \frac{(2\mu_{\mathrm{x}}\mu_{\mathrm{y}} + C_1)(2\sigma_{\mathrm{xy}} + C_2)}{(\mu^2{}_{\mathrm{x}} + \mu^2{}_{\mathrm{y}} + C_1) + (\sigma^2{}_{\mathrm{x}} + \sigma^2{}_{\mathrm{y}} + C_2)}$$

| **Model** | **SSIM value (base: 0.5416)** | **Relative SSIM value** |
|---|---|---|
| CNN - L1 | 0.5814 | 0.1598 |
| CNN - L2 | 0.5798 | 0.1566 |
| CNN - Huber | 0.5774 | 0.1519 |
| GAN - L1 | 0.706 | 0.3035 |
| GAN - L2 | 0.7037 | 0.2994 |
| GAN - Huber | 0.6855 | 0.2657 |

Table 1: Evaluation of SSIM values

It can be observed from Table 1 that the **GAN-L1 model** has the highest relative SSIM value.

# 6 Conclusions

- Images generated by the GAN model are quantitatively and qualitatively better than the images generated by the CNN models. This suggests that the GAN models are robust, even though training for these models is tricky.
- The outputs generated using L1 loss are more sharp while those generated using L2 loss are comparatively blurry.
- It can be observed that the coloring in both the CNN and GAN generated images are generally within the outline of the input images. This can be attributed to the uniformity in the orientation of the images in the dataset.
- In the case of sports shoes, the GAN gets confused by the presence of too many edges near the laces region. Hence, it fails to recognize them sharply and mixes it up with the background color patches.
- Since the *edges2shoes* dataset has classes and subclasses metadata available, we plan to learn a classification based colorization model to compare improvement across classes.

## 7 Implementation

We implemented the CNN model in *Keras* (using tensorflow as backend) while the cGAN model was implemented in *Tensorflow*. We set the hyperparameters, for example learning rate, kernel size, stride, weight decay coefficient, dropout, L-norm loss weight coefficient ($\lambda$), number of generator & discriminator filters as proposed in [4]. We also implemented the utility functions for preprocessing and data loading. Finally, we used the python library *scikit-image* for SSIM metric evaluation.

## References

[1] A. Yu and K. Grauman, "Fine-grained visual comparisons with local learning," in *CVPR*, 2014.

[2] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of IEEE International Conference on Computer Vision*, 2015.

[3] Z. Wang and E. P. Simoncelli, "Translation insensitive image similarity in complex wavelet domain," in *ICASSP*, 2005.

[4] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017.