Colorizing Polygon Shapes using Conditional UNet

Abstract

This project aimed to develop a deep learning model to colorize grayscale polygon images on the basis of a text prompt specifying a color name. A Conditional UNet architecture was implemented in PyTorch, by which the model could understand and coalesce image structure with textual information. After training with a synthetic dataset of simple geometric shapes paired with color names, the model successfully learned to output colored versions of a grayscale input conditioned on the specified color.

Introduction

Conditional image generation, where an image is created or modified depending on additional conditioning inputs such as text or labels, is a critical area of research in deep learning. This project delves into the modeling of the gray polygon images for colorization under a simple prompt of color. It came from the inspiration stemming from the challenge that much different types of data are being combined- images and text- and making a model that learns meaningful associations between the two.

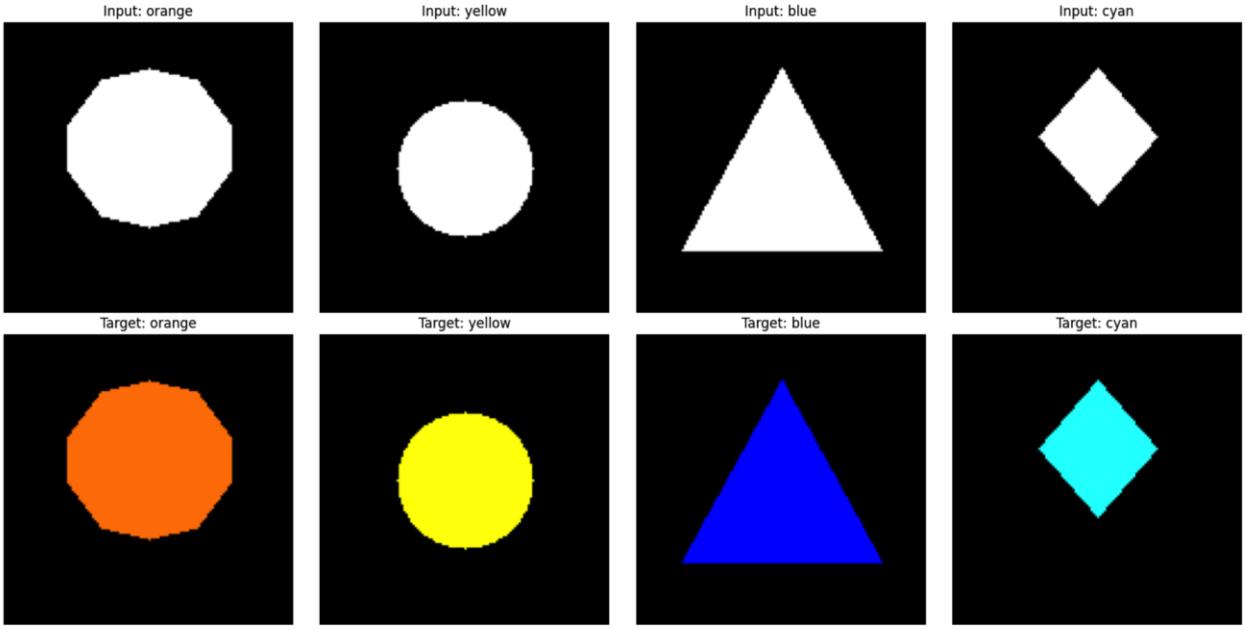
The project utilizes a Conditional UNet-a variant of the UNet-that has been effective in image segmentation and generation. The model is trained to learn the colorization of polygonal shapes such as circles, squares, and triangles against color names given as input, e.g., "red", "green". The conditions provide an easy-to-understand example of conditional generation while offering enough complexity to delve into deep learning design choices.

Methodology

The architecture of this project is a Conditional UNet. In the first part of the network, the encoder takes the input grayscale image and extracts important spatial features from it. The input color name is then embedded as a numerical vector and added into the bottleneck (lowest resolution layer of the network). The decoder reassembles the image with both this information and additional skip connections from the encoder to preserve the shape structure.

The dataset applied here was synthetically generated using Python and contained approximately 1000 images of simple shapes (circles, triangles, squares) in grayscale together with the color labels, which included: red, green, blue, yellow, black, and white. It was 64x64 pixels per image, keeping the dataset easy but at the same time suitable for training.

The model had been implemented using PyTorch, then optimized with the Mean Squared Error (MSE) loss where it compared the generated colored output to the ground-truth. Adam optimizer was chosen for optimization, and the training was performed for 100 epochs. Wandb was configured to track training metrics and display visualization. After completing the training process, evaluation of the model was carried out with the inference notebook, whereby grayscale test images and color prompts were fed into the model to observe if it generated the correct colorized outputs or not.



Conclusion

With this project, I was able not only to understand conditional image generation methods but also to make practical use of this technique. The Conditional UNet was very effective in learning the relationship between grayscale shape structures and text color prompts and then producing outputs that tend to align with the input color and form. I was able to gain further knowledge of UNet architecture, text conditioning methods, and training workflows in PyTorch.

In an ideal scenario, I hope to take this fundamental knowledge and extend it by trying to train this model on more complex shape datasets, natural-language color descriptions, and pretrained embeddings to further push model performance and flexibility. Overall, this project was a wonderful learning experience and gave me the confidence to design, implement, and evaluate custom deep learning models.