Ayna ML Internship Report

---

1. Objective

This project aims to develop a deep learning model capable of generating colored polygon images from grayscale inputs and a given color name. This approach simulates multimodal learning by combining visual (grayscale image) and textual (color name) inputs to predict an RGB image output, fulfilling a use case relevant to generative visual systems.

---

2. Dataset Overview

- **Inputs**:
    - Grayscale polygon image (1 channel, 64x64).
    - Color name (e.g., “red”, “green”, “blue”).
- **Output**:
    - Colored RGB polygon image (3 channels, 64x64).
- **Dataset Format**:
    - Provided via PyTorch Dataset class.
    - Color names embedded using torch.nn.Embedding.

---

3. Methodology

3.1 Preprocessing:

- Grayscale and RGB images normalized to [0, 1].
- Color names encoded as indices and mapped to embeddings.
- Resized images to 64x64 (if needed).

3.2 Model Architecture:

A custom **UNet** model was implemented with modifications to incorporate color embeddings:

- Encoder:
    - 3 convolutional blocks with increasing filters (16 → 32 → 64).

- Decoder:
  - Transpose convolutions to upsample and reconstruct the image.
- Color Conditioning:
  - Color embedding vector expanded spatially and concatenated with image feature maps.
  - Allows the model to "paint" the grayscale image according to the color name.

3.3 Loss Function:

- Mean Squared Error (MSE) loss between predicted and ground truth RGB images.

3.4 Optimizer:

- Adam optimizer with learning rate of 0.001.

---

4. Training Details

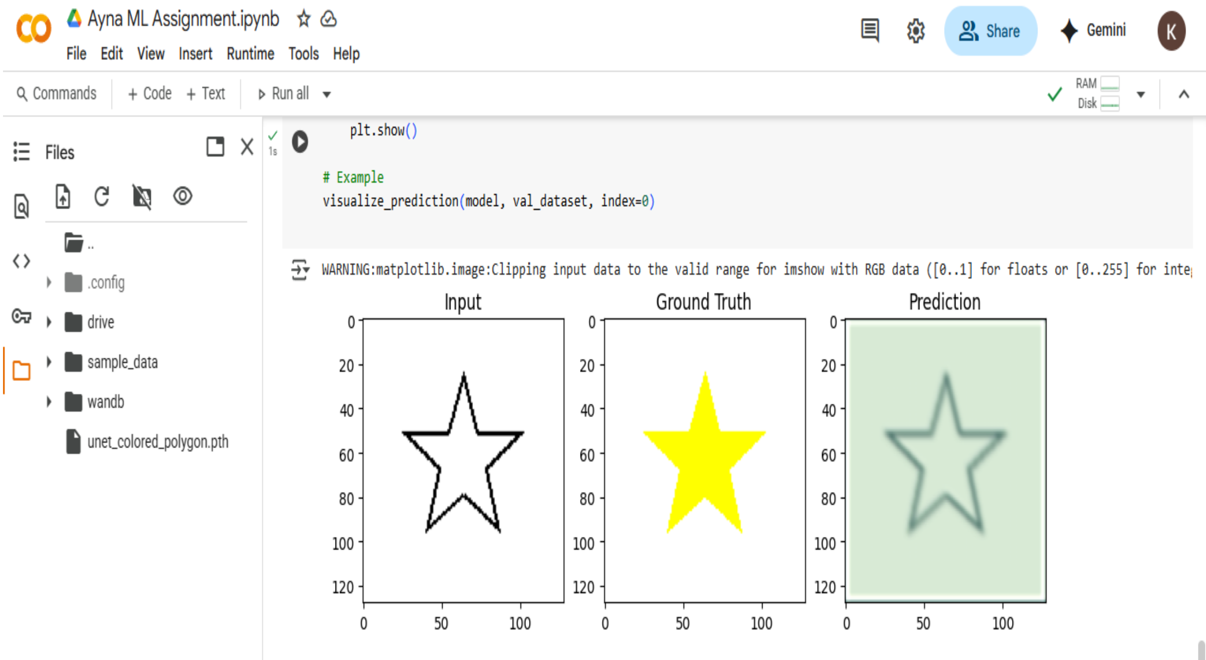
- **Epochs**: 20
- **Batch size**: 16
- **Framework**: PyTorch
- **Tracking**: Weights & Biases (W&B) used to log:
  - Training and validation loss.
  - Sample outputs.

---

5. Evaluation and Results

- Visual comparison of predicted vs. ground truth images shows:
  - Accurate color transfer aligned with the provided color name.
  - Correct structure and placement of the polygons.
- Validation loss converged steadily, indicating proper learning without overfitting.

Sample Visual Output:



---

6. Code Highlights

- CustomDataset class to load grayscale image, color index, and RGB target.
- UNet model modified to integrate color embeddings into decoder pathway.
- Visualization script to save predictions and display outputs side by side.
- Model saved in .pth format.

---

7. Challenges Faced

- Integrating textual embeddings into image-based models.
- Ensuring spatial broadcast of embeddings while maintaining tensor alignment.
- Limited dataset size required careful tuning to avoid overfitting.

---

10. Conclusion

This project demonstrates the power of multimodal learning by conditioning an image generation model with textual input. The successful use of a UNet model modified for color guidance shows potential for broader applications in vision-language systems, such as art generation, colorization, and educational tools.