U-Net for Polygon Colorization

**Project Summary**

Training a UNet model from scratch to generate an image of a coloured polygon. The model should take two inputs:

* A **polygon image** as input.
* A **colour name** as a condition.

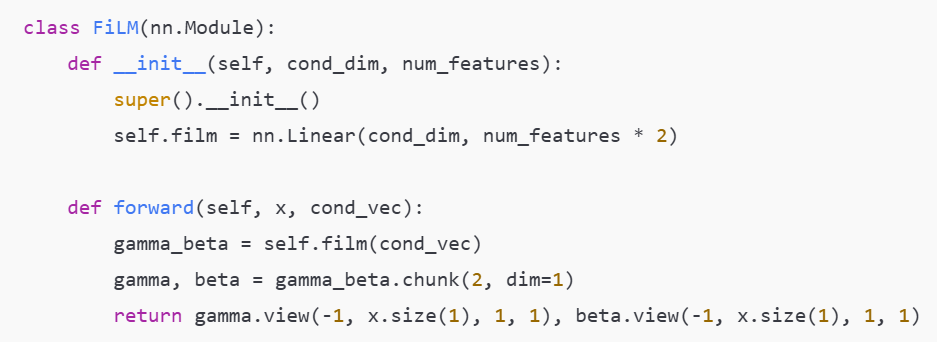
The goal is to generate an RGBA image that correctly fills the polygon with the target colour.

**Architecture: U Net + Conditioning (Fi LM)**

The model is a custom-built U Net architecture from scratch, enhanced with:

* **Fi LM Layers (Feature-wise Linear Modulation)** for conditioning
* **Group Norm** instead of Batch Normalization for better generalization.
* **Si LU (Swish)** activation function for smooth learning
* **Colour embedding** from a one-hot or RGB input

**Condition Fi LM Mechanism**



Fi LM is applied after normalization in each Conv Block to modulate features based on the colour condition.

**Hyperparameters**

| **Hyperparameter** | **Value** |
| --- | --- |
| Image Size | 192×192 |
| Batch Size | 32 |
| Learning Rate | 1e-4 |
| Optimizer | Adam |
| Loss Function | 0.7 \* L1 + 0.3 \* MSE |
| Conditioning Input | One-hot colour name or RGB (dim = 9) |
| Epochs | 80 (No overfitting or Underfitting) |

**Training Dynamics:**

The model was trained using both **L1 loss** and **MSE loss** to ensure sharp yet consistent outputs.

* **Training Loss** decreased steadily.
* **PSNR** improved during early epochs, then plateaued.

**Sample Predictions:**



**Key Learnings:**

1. Fi LM conditioning is very effective when used across all Convolution Blocks.
2. Group Norm + Si LU works well with small batch sizes.
3. Learned that U Net performs better than CNN.
4. Tracking via Weights & Biases was critical for debugging but graphs helped to check if model was trained correctly.