INSIGHTS REPORT

Learning Rate: 1e-4 (stable convergence, less oscillation)

Optimizer: Adam (faster convergence and better generalization)

Batch Size: 8 (balanced between stability and GPU memory)

Epochs: 50

Loss Function: MSELoss (worked best for pixel-level regression)

Augmentations: Random horizontal flip

MSELoss performed better than L1 as the task is to match RGB pixel values precisely.Adam outperformed SGD with momentum by adapting learning rates per parameter.Batch size 8 gave the best training speed vs memory usage trade-off.

UNet Architecture and Conditioning

Final UNet Design:

- **Input Channels**: 6
  → 3 for base image + 3 for polygon mask
- **Output Channels**: 3 (for RGB)
- **Depth**: 4 levels (Encoder → Bottleneck → Decoder)
- **Skip Connections**: Yes (standard in UNet)
- **Activation**: ReLU in intermediate layers, Sigmoid/Linear at output (to maintain RGB range)
- **Normalization**: BatchNorm

Ablations Tried:

- Tried input as single channel mask vs 3-channel → 3-channel preserved structure better.
- Tried deeper UNet (5 levels) → increased training time with minimal gains.
- Skip connection removal → led to blurrier results (spatial info loss).

Training Dynamics

Loss & Metrics:

- Training Loss dropped smoothly with LR=1e-4.
- Validation loss plateaued ~epoch 25.
- No signs of overfitting observed until ~30 epochs.

Qualitative Trends:

- Early epochs: Model learned outlines, colors were random.
- Mid training: Started matching general color regions.
- Final epochs: Captured object-specific colors and boundaries.

Failure Modes:

- Some shapes near image edges were under-colored.
- Rare class polygons (low frequency) were poorly predicted.
- Polygon overlaps caused color bleed occasionally.

Fixes Attempted:

- Added more balanced samples (underrepresented polygons).
- Applied stronger augmentations (flip, scaling).
- Tweaked polygon mask representation from binary to RGB channels for richness.

Key Learnings

1. **Mask Conditioning Matters**
   Representing polygon masks as RGB helped preserve structural information better than binary masks.
2. **Model Size Balance**
   A moderate-depth UNet worked well. Larger models didn't bring much gain, proving that smart input conditioning is more powerful than raw depth.
3. **Data Quality > Model Complexity**
   Improving dataset quality (augmentations, mask detail) had more effect than changing architectures.
4. **W&B Artifacts FTW**
   Using Weights & Biases for model versioning made collaboration and deployment easier, especially for sharing latest models directly.
5. **Inference Notebook Should Be Lightweight**
   Separating training from inference helped keep things modular and clean.