**1. Transformation Techniques**

The code applies several image transformation techniques during training and validation. These transformations enhance the robustness of the model by introducing variability in the input data:

**Training Transformations:**

* **Horizontal Flip**: Randomly flips the image horizontally with a probability of 50%.
* **Vertical Flip**: Randomly flips the image vertically with a probability of 50%.
* **Random Gamma Adjustment**: Randomly adjusts the image gamma with a range of (70, 130) to simulate different lighting conditions.
* **RGB Shift**: Randomly shifts the red, green, and blue channels with specified limits to simulate color variations.
* **Normalization**: Normalizes the pixel values using the mean (0.485, 0.456, 0.406) and standard deviation (0.229, 0.224, 0.225), which is typical for pre-trained ImageNet models.
* **ToTensorV2**: Converts the transformed image into a PyTorch tensor.

**Validation Transformations:**

* **Normalization**: Similar normalization as used in the training transformations.
* **ToTensorV2**: Converts the image into a PyTorch tensor.

**2. Model Architecture**

The chosen model architecture is **UNet++** (UnetPlusPlus), implemented using the segmentation\_models\_pytorch library. This is a state-of-the-art model for semantic segmentation tasks, known for its superior performance on medical and natural images.

**Key Features of the Model:**

* **Backbone**: The model uses a **ResNet-34** encoder, pre-trained on ImageNet. This encoder extracts high-level features from the input image.
* **Decoder**: The decoder reconstructs the segmentation mask with rich contextual information and detailed spatial resolution.
* **Input Channels**: The model accepts 3-channel (RGB) images as input.
* **Output Classes**: The model predicts segmentation masks for 3 classes, corresponding to:
  + Background (0)
  + Red regions (1)
  + Green regions (2).

UNet++ enhances the original UNet design with nested and dense skip connections, allowing better feature fusion and improved segmentation accuracy.

Link to my Github repository:

<https://github.com/phoenix301123/Deep-Learning-BKAI>

Wandb charts:

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, Sơ đồ

Description automatically generated

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, Sơ đồ

Description automatically generated

The checkpoint model is the one with these numbers: batch\_size = 8, number of epochs = 250 and learning\_rate = 0.0005

git clone <https://github.com/phoenix301123/Deep-Learning-BKAI>

cd Deep-Learning-BKAI

python3 infer.py --image\_path image.jpeg