**SegFormer Summary, Model Variants, Issues Faced & Solutions, and Advanced Applications**

**🌍 What is SegFormer?**

SegFormer is a **state-of-the-art semantic segmentation model** developed by NVIDIA. It combines the benefits of Transformers (global receptive field) with lightweight MLP-based decoders for high-speed and high-accuracy segmentation.

**Key Characteristics:**

* **No positional encoding**: Uses hierarchical Transformer encoder.
* **Efficient MLP decoder**: No heavy upsampling decoder required.
* **Supports variable input size**.
* Pretrained on **ADE20K** (150-class scene parsing dataset).
* Trained via **pixel-wise classification**.

**Usage:**

from transformers import SegformerForSemanticSegmentation

model = SegformerForSemanticSegmentation.from\_pretrained("nvidia/segformer-b0-finetuned-ade-512-512")

**🔖 Available Model Variants:**

SegFormer has several versions depending on model size and complexity:

| **Model Name** | **Parameters** | **FLOPs (G)** | **FPS (1024x1024)** |
| --- | --- | --- | --- |
| segformer-b0 | 3.7M | 8.4 | 123 |
| segformer-b1 | 13.7M | 20.1 | 85 |
| segformer-b2 | 27.5M | 45.3 | 56 |
| segformer-b3 | 47.3M | 83.0 | 41 |
| segformer-b4 | 64.1M | 127.4 | 33 |
| segformer-b5 | 81.3M | 186.2 | 27 |

**❌ Issues Faced During Training & Inference**

**1. Defect Not Detected**

* **Cause**: Defect mask values were labeled as 2 instead of 1.
* **Fix**: Added remap logic in dataset class:

mask\_np[mask\_np == 2] = 1

mask\_np[~np.isin(mask\_np, [0, 1])] = 255 # Ignore other labels

**2. Segformer Not Predicting Class 1 (defect)**

* **Cause**: Incorrect label mapping + ignore index not set
* **Fix**: Set

model.config.ignore\_index = 255

**3. Invalid Metrics (compute\_metrics error)**

* **Cause**: Wrong format passed to HuggingFace's evaluate
* **Fix**: Removed ClassLabel feature and computed metrics manually using sklearn

**4. CUDA Runtime Errors**

* **Cause**: Torch not compiled with GPU support
* **Fix**: Installed GPU-compatible PyTorch from official <https://pytorch.org/get-started/locally/>

**🔢 Advanced Evaluation Tools Implemented**

* **Validation Set Benchmarking**:
  + mIoU, Precision, Recall, Confusion Matrix using sklearn
* **Visual Sanity Check**:
  + Side-by-side display: image, GT mask, predicted mask
* **Video Inference**:
  + Frame-by-frame segmentation with overlay
  + Export to PNG and JPEG

**🔄 What Can You Do Next with SegFormer?**

**✅ Applications:**

* Quality Control (e.g. Cotton Defect Detection)
* Autonomous Driving (Lane & object segmentation)
* Aerial Mapping
* Agricultural Disease Detection

**💡 Advanced Ideas:**

* **Class-weighted loss**: For imbalance issues
* **Data Augmentation**: MixUp, CutMix, Albumentations
* **Multi-Class Segmentation**: Add more classes (e.g., leaf, stem)
* **Distillation**: Compress model into efficient version for edge devices
* **Convert to ONNX**: For deployment in real-time systems

**🔄 Summary of What You Learned**

1. How to train SegFormer on a custom 2-class dataset (background/defect)
2. Importance of remapping labels & ignore index
3. How to fix evaluation pipeline with sklearn
4. Inference pipeline from single image → video stream
5. How to visualize predictions effectively

📄 Save this document for future reference before retraining or deploying SegFormer.