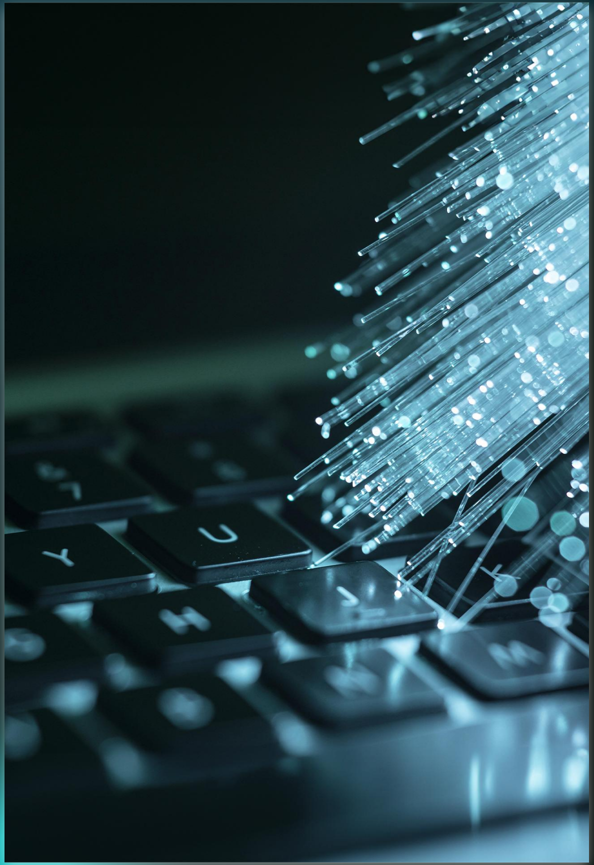
Big Data

Project Review - II

121AD0011 - Shenbagasujan V S
121AD0013 - Madhav Sharma
Mentor - Dr. Srinivas Naik

BigRemoteFormer

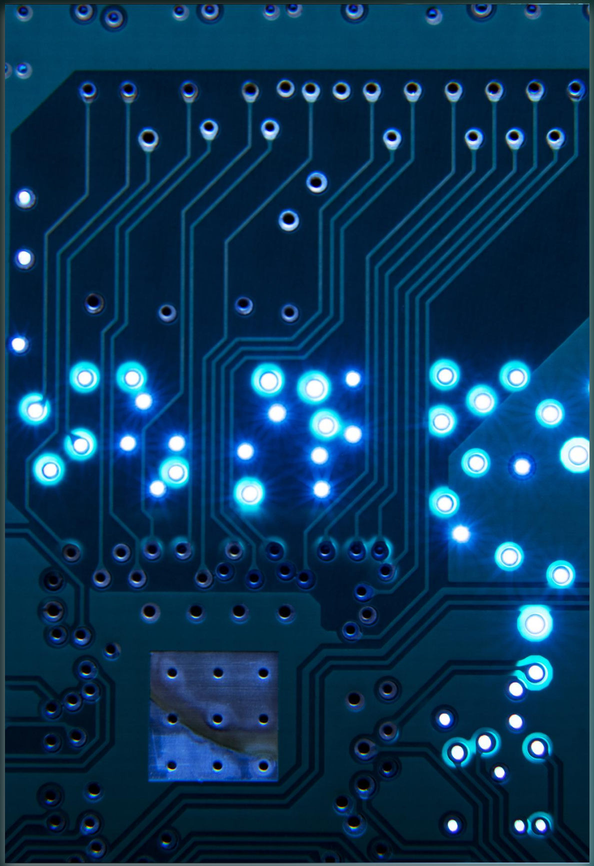
Transformer for Remote Sensing using Big Data



Problem Statement

To train a robust unet based transformer based model on large .tif or .tiff or .jpg (image files) with mask in a parallel manner which can perform semantic segmentation efficiently with good accuracy

OBJECTIVES OF THE PROJECT



OBJECTIVES

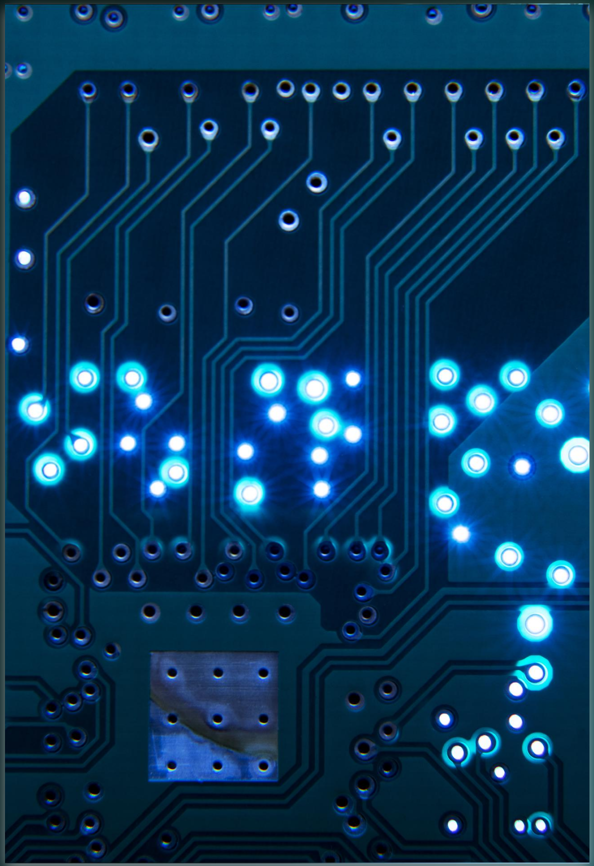
OUR AIM

The idea of our project is to extend the transformers to Remote Sensing
The victims of the “single big .tif ” must be known of the torment when they get a single large image along with mask!
They have to do the task using different softwares!

THE GOAL

Our code will do pacification , semantic segmentation, and evaluation of the semantic segmentation problem
The implementation of the project will be done using big-data approach in a parallel manner

Proposed Solution



Problem - I Single Large Image

Parallelizing Image Data: Convert image data into NumPy arrays and parallelize them into RDDs.
Splitting Image into Patches: Define a function to split images into smaller patches.
Applying Patch Splitting Function: Use Spark.flatMap() transformation to apply the patch splitting function in parallel.
Collecting Patches: Collect the resulting patches from all partitions into the driver node.
Outputting Patch Information: Iterate over collected patches and output relevant information.

Problem - II Generating Dataset

- The dataset used here is oxford pets dataset, the dataset contains the “ground truth” or the mask and the images of various varieties of pets
- The task is to normalize the images and to make all images of standard size
- The other challenge we faces was the preservation of class labels , to ensure this we use K-Nearest Neighbors approach which will match the image with corresponding masks
- This all must be ensured in generating dataset
- Then to make model more robust we introduce image augmentation by rotating and flipping the images, this makes model more robust and less prone to outliers

Problem - III Defining Model Architecture

- We need to define a model such that model must get some idea about the domain so we are using transfer learning approach to import the MobileNetV2 model
- The UNet architecture is used while Transformer layers are embedded as upsampling layers*
- Downsampling of the samples are done then upsampling is done then masks are generated*
- The UNet is very popular architecture in Machine Learning Domain[7]

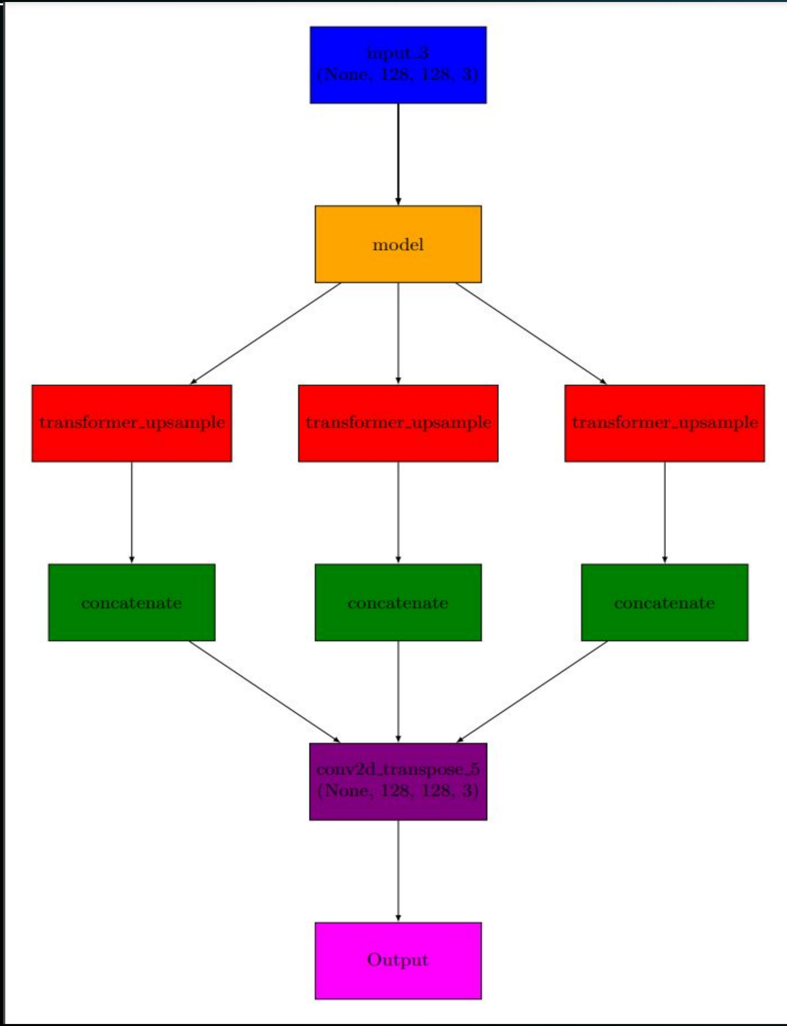
*Madhav : These ideas are unique not taken from any source

Upsampling

- Increases image size by adding new pixels.
- Techniques include nearest neighbor, bilinear, and bicubic interpolation.
- Used for image resizing, super-resolution, and data augmentation.

Downsampling

- Reduces image size by removing or averaging pixels.
- Techniques include average pooling, max pooling, and subsampling.
- Used for image compression, feature extraction, and reducing complexity.

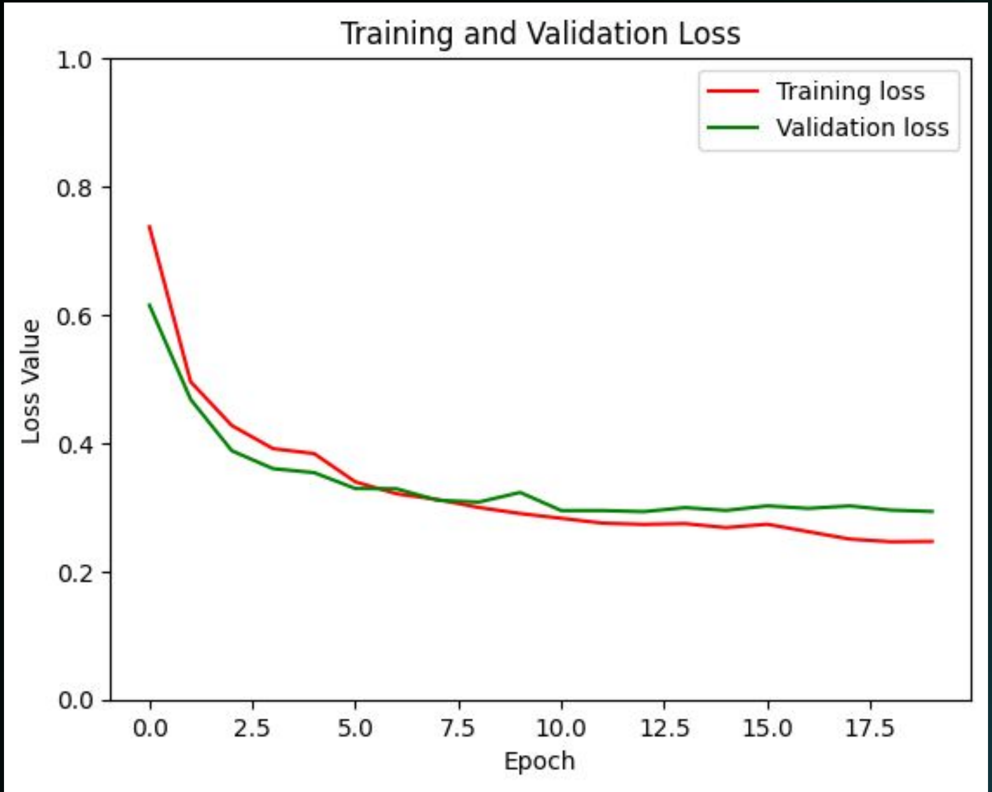

input_3
(None, 128, 128, 3)
model
transformer_upsample
transformer_upsample
transformer_upsample
concatenate
concatenate
concatenate
conv2d_transpose_5
(None, 128, 128, 3)
Output

Problem - IV Model Evaluation

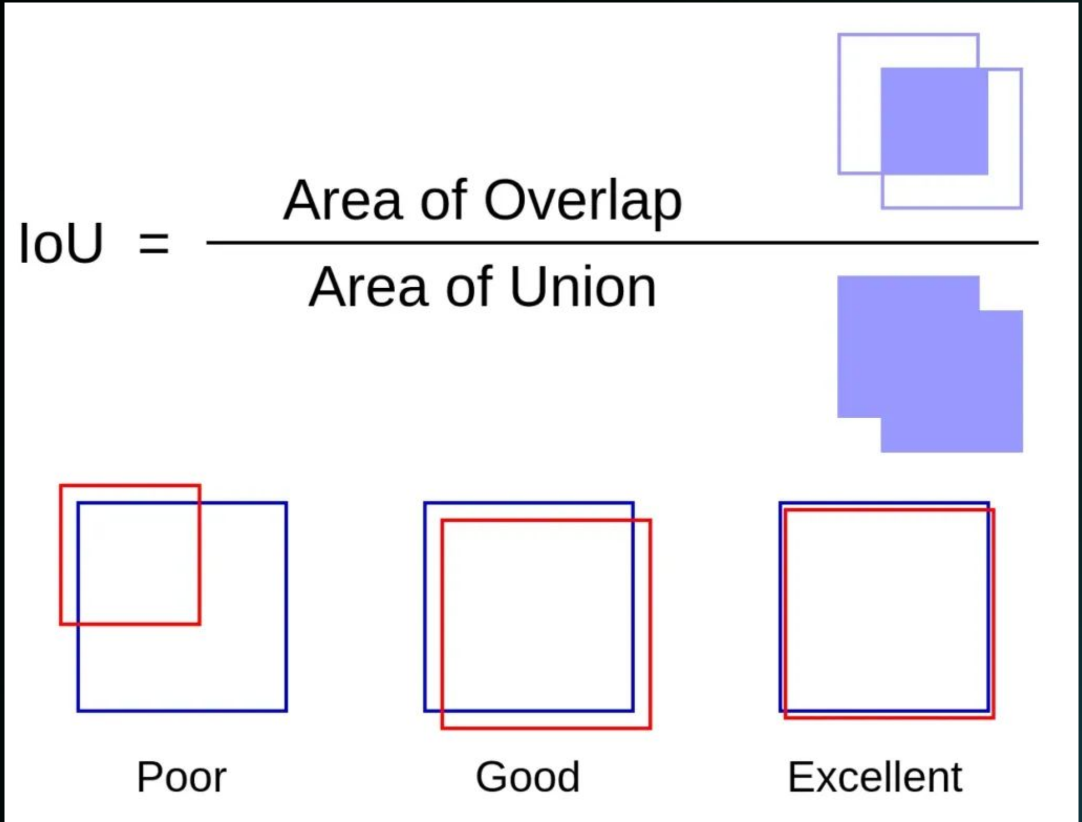
- Model Evaluation can be done using Categorical Cross Entropy loss function
- The IoU score is used to check the overlapping between predicted and actual masks
- The dice loss is used for handling class imbalance problem
- The visual results are quite visible in next slides

Visualizing the Results

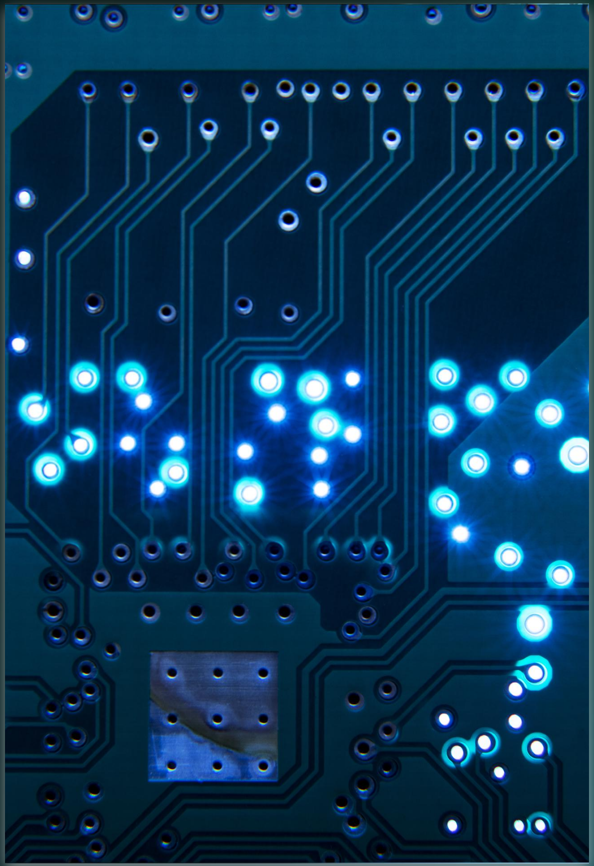
Visualizing Loss Function



Calculating IoU Score



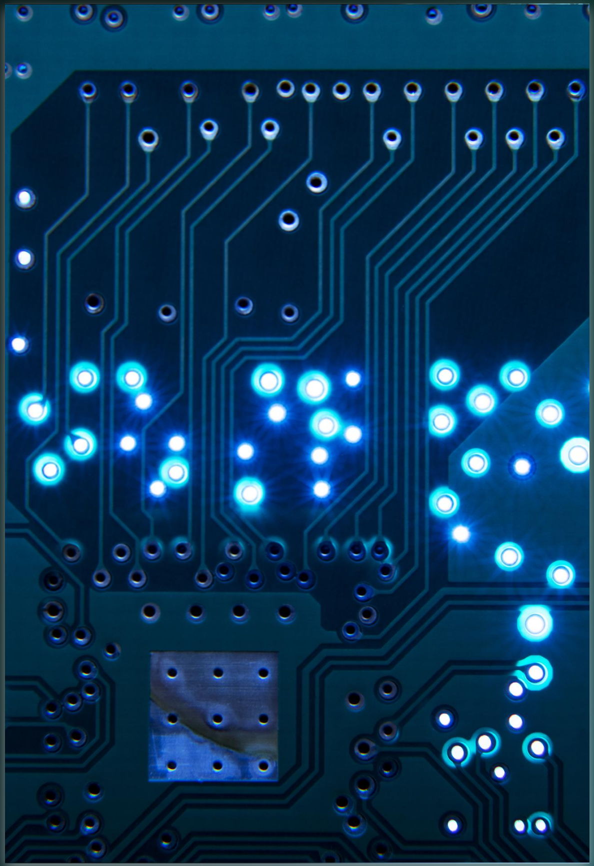
Drawbacks



1. The approach is only good when we have limited computational resources , we can implement this paper of high performance machine. The architecture can be implemented using Big Data approach using Spark ✓ [3]
2. The transformer only deals with classification we can implement semantic segmentation on remote sensing datasets means I2M(Image to model)*[6] ✓
3. The paper only trains some limited datasets we can train remote sensing by using U-Net ✓ [2]

* Madhav: The datasets can be diversified

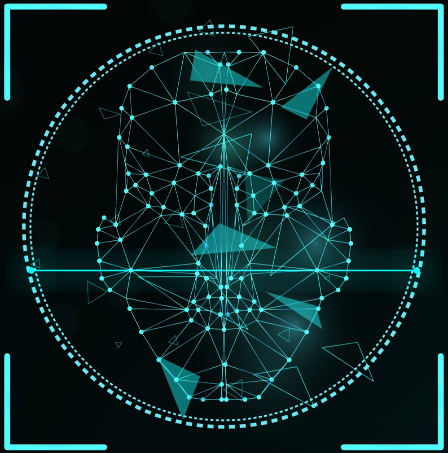
Demonstration



Demonstration

We see the project demonstration by accessing the below attached link.

Click here



Project Implementation

Timeline of Project

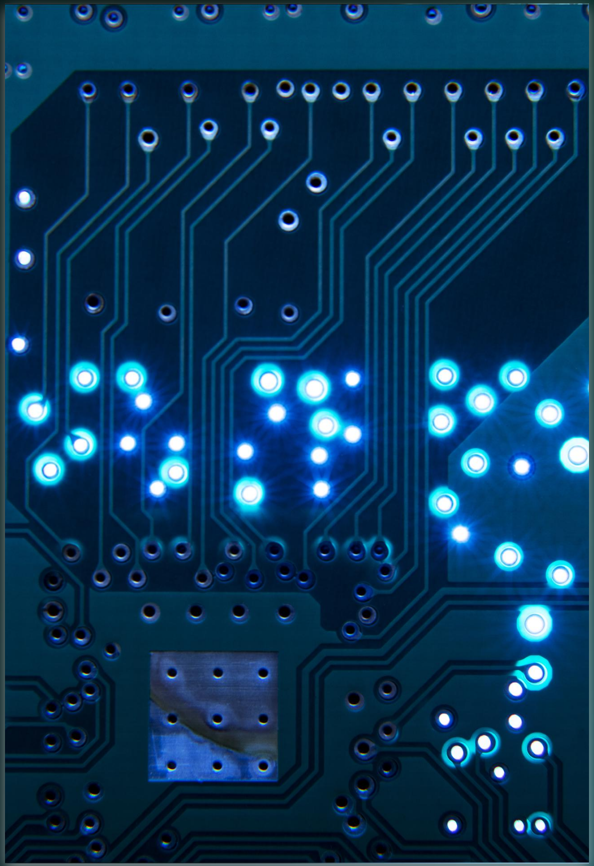
- Literature Survey : 19 March to 22 March
- Approach Brainstorm : 22 March to 24 March
- Selecting Appropriate Dataset : 24 March to 27 March
- Preprocessing of Data : 27 March to 31 March
- Patchification : 1 April
- Transfer Learning Exploration 1 April to 3 April
- Model Design : 3 April to 5 April
- Consolidating and Experimenting : 5 April to 6 April
- Presenting and making reports and documentations : 6 April

Issues Faced

- The dataset was needed with a optimal size which can run on our Colab or Spark Environment, It took 2 hours on this
- The Spark Environment was not setup on the Servers so we had to implement this on Colab (It will work on the cluster as well), It wasted our majority of time as there was no proper cluster setup
- The patchification was to be done in a parallel manner this created many logical errors initially, It took 2 hours on this
- The data was unstructured and we had to normalize the dataset and resize the dataset so we took codes from my old model in which preprocessing step was performed
- The UNet architecture has to be used and transformer was to be used and we needed to design a robust model, raw model was designed in 5 hours
- The model architecture was the main chaotic area where the majority of the time was spent, the model is very complex which involves many downsampling and upsampling techniques , It took 10 hours on this
- Appropriate Evaluation techniques was to be seen for semantic segmentation advice was taken from our most helpful PHd people who guided us throughout the project, It took us 1 hr to review this

0

Experiments



Preliminary Study

Experiment	Output	Observation
Training model with CPU/GPU	CPU takes 20 hours while GPU takes 20 minutes	GPU supports parallel processing and image data is very large
The model fed with some random input	The model performs just ok in spite of being trained to dog dataset	The transfer learning enabled model to work on other objects advantage of transfer learning
The patchification was done serial order	Time taken for less number of image was less while for large amount of samples it was low	Big Data Approach works well for large datasets
The data was not normalized	The accuracy reduced	The magnitude must not influence model performance
The data augmentation was not done	The accuracy reduced	Model performance reduces due to model being less robust to dirty data
Epochs to 30 / Epochs to 5	Both cases accuracy reduces	The model is overfit/ model is underfit

Our Model

Layers	# Heads	MLP Ratio	Params (M)	Accuracy
2	2	1	0.284	89.22
4	2	1	0.482	89.88
6	4	2	3.327	90.09
7	4	2	3.760	91.80

Previous Model[3]

Layers	# Heads	MLP Ratio	Params (M)	Accuracy
2	2	1	0.284	89.17
4	2	1	0.482	91.45
6	4	2	3.327	94.81
7	4	2	3.760	94.78

0

Conclusion and Future Works

Future Works

- Implementing this code in a better GPU system and using a large .tif file for patchification and then passing them to the model along with labels
- Trying to implement domain adaptation approach to reduce the task of manual labelling
- Training the model on larger remote sensing datasets
- Implementing the frontend of the model for better interface
- To explore more metrics as guided by our PHd friends , like dice loss

Conclusion

KNN approach was used to combine the image with the true mask

To ensure the model is robust data augmentation is done

The data is cleaned normalized and resized

The proposal stands to develop a frontend and give better computational resources to make model better

The Hybrid-Transformer was used for semantic segmentation

Downsampling and Upsampling techniques were used carefully for ensuring the dimensions of mask and the image must remain same

0

Contributions

Contributions

Madhav Sharma

- Data source identification
- Image patchification using rdd
- Model architecture design and mathematics of upsampling and downsampling
- Model test metrics
- Decoupling Functions

Shenbagasujan VS

- Data cleaning
- Data standardization
- Data normalization
- Exploration of computing environments available
- Data visualisation
- Model result visualisation

References

[1]Cross-Parallel Transformer: Parallel ViT for Medical Image Segmentation by Dong Wang
[2]Unsupervised Domain Adaptation for the Semantic Segmentation of Remote Sensing Images via One-Shot Image-to-Image Translation
[3]Compact Transformers by Ali Hassan
[4] Tensorflow Documentation
[5]Kaggle Datasets
[6]Segmenter: Transformer for Semantic Segmentation by Robin Strudel, Ricardo Garcia
[7]Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation Hu, Hu Cao, Yueyue Wang

Thank You!

Shenbagasujan V S 121ad0011@iiitk.ac.in
Madhav Sharma 121ad0013@iiitk.ac.in