Road Segmentation Using DeepLabV3+

Subtitle: AI Capstone Project

Diep – nguyen9657@saskpolytech.ca
Harnil - patel7344@saskpolytech.ca
Do – thin0013@saskpolytech.ca

SASKATCHEWAN POLYTECHNIC

Objectives

Main Goal:

- Implement and evaluate the **DeepLabV3+** model for **road segmentation** in satellite imagery.

Aims:

- Utilize DeepLabV3+'s advanced features (e.g., atrous convolution) to effectively segment roads in various urban and rural environments.
- Measure and analyze performance using metrics like Accuracy, Intersection over Union (IoU), and Dice Coefficient.

Project Timeline

Jan 2025

Project Planning and Data Collection

- Research and finalize topic with suitable datasets
- Define project goals, scope, & success metrics
- Prepare high level proposal & presentation
- Set up the development environment

Feb 2025

Data Preprocessing and Exploration

- Load and explore the dataset (Visualize satellite images & road mask)
- Preprocess the data (Resize, normalize, split data)
- Augment the dataset (Rotation, flipping, cropping)

Model Selection and Baseline Development

- Select model and implement a baseline model
- Train and evaluate the baseline model

Mar 2025

Model Optimization/Improvement

- Analyze baseline performance to optimize & improve model
- Implement, train and evaluate the optimized/improved model
- Compare results among the baseline & optimized/improved model

Validation, Testing

- Perform cross-validation to ensure model robustness
- Test the model on the test set & compute final metrics
- Visualize predictions, analyze failure cases & identify potential improvements

Apr 2025

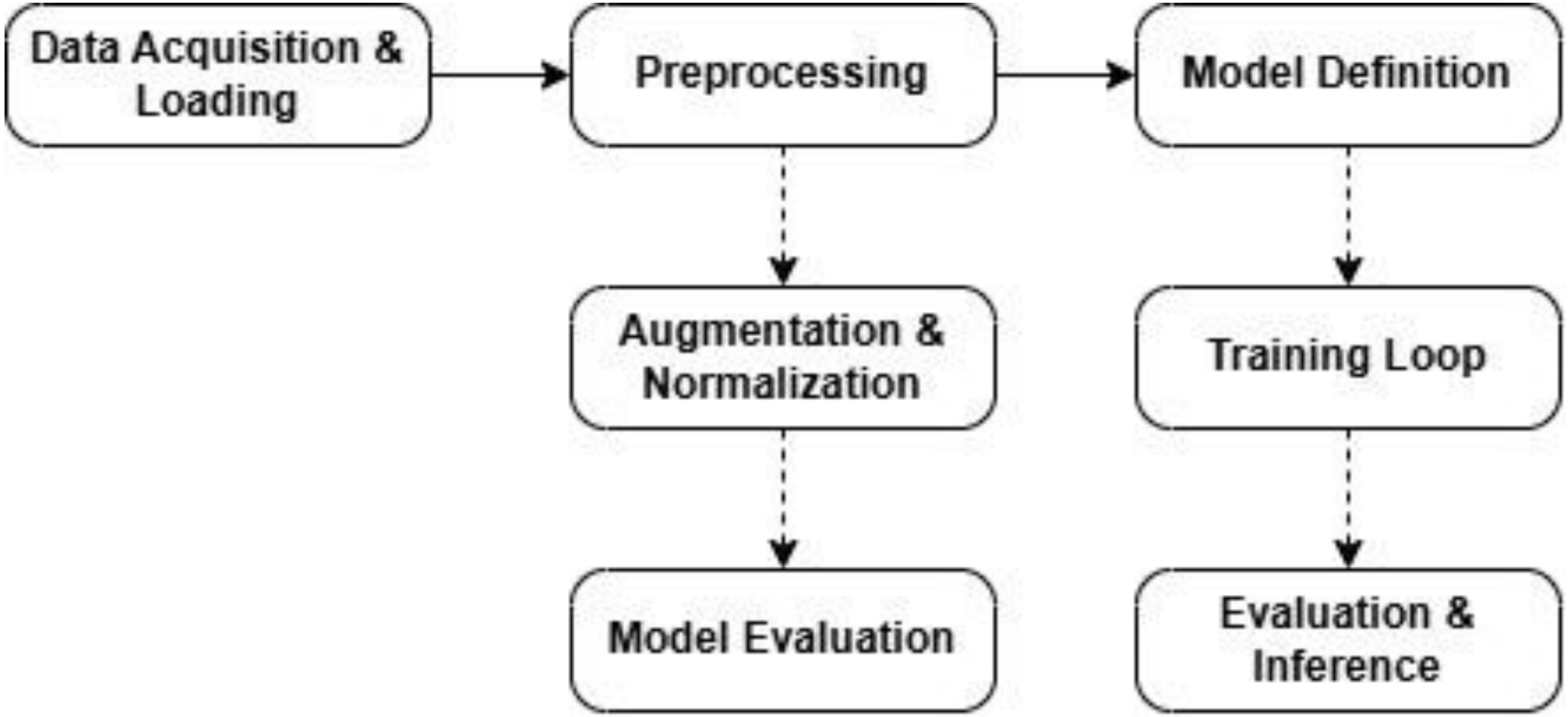
Deployment

- Convert the trained model to a deployable format
- Develop a simple pipeline for inference & test

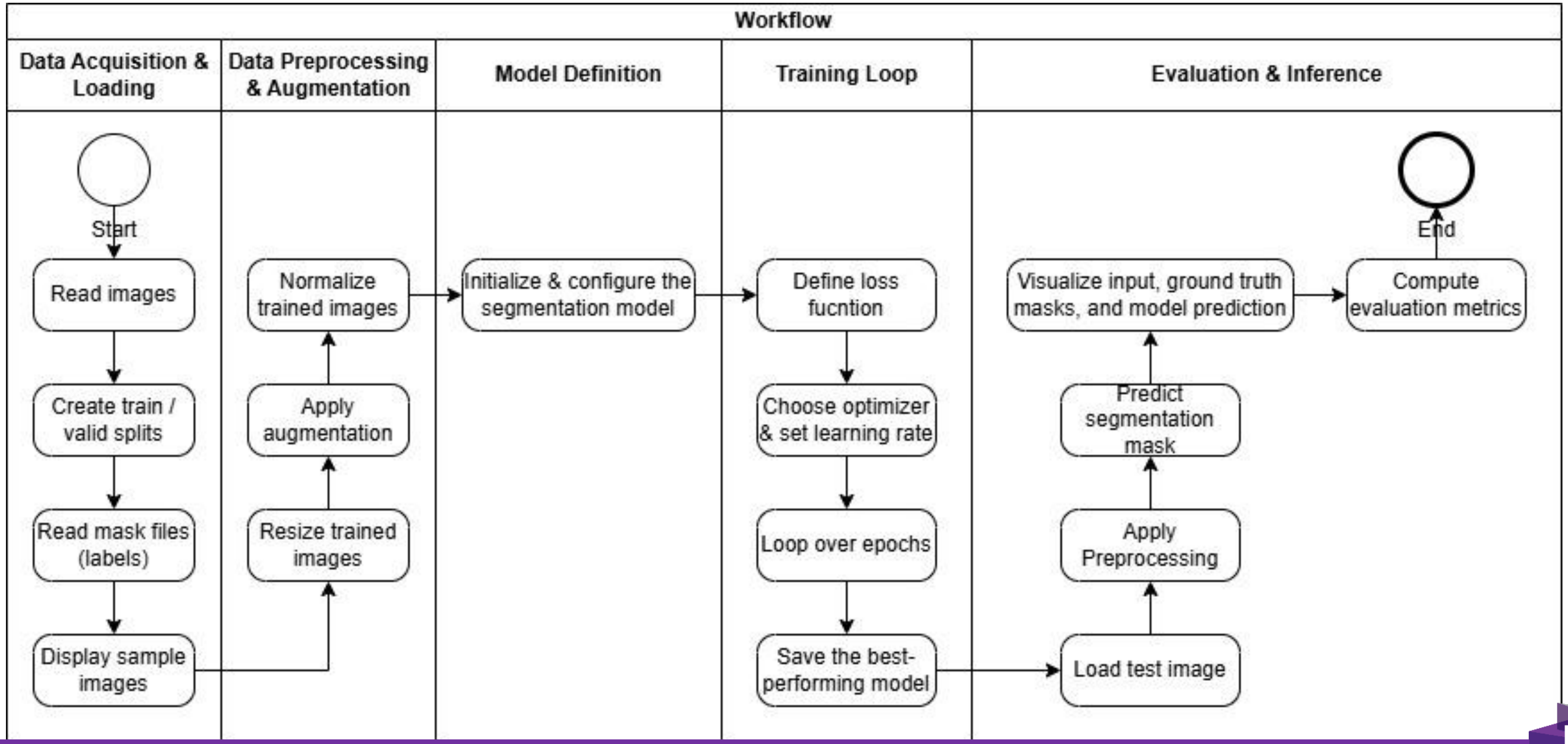
Finalization and Documentation

- Prepare a detailed project report & presentation
- Conclude the project with programming codes & reports

Detail Project Block diagram



Project Flowchart



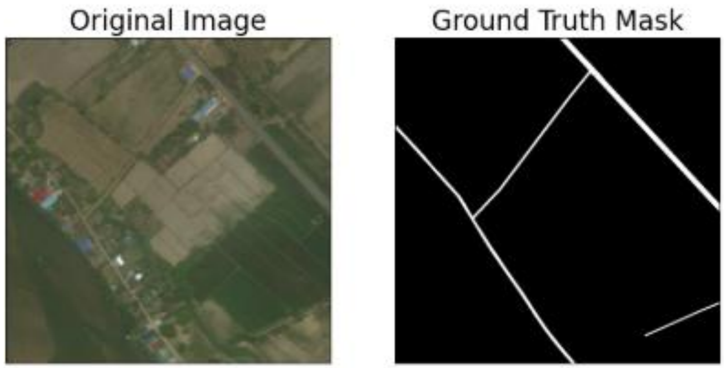
Dataset Used

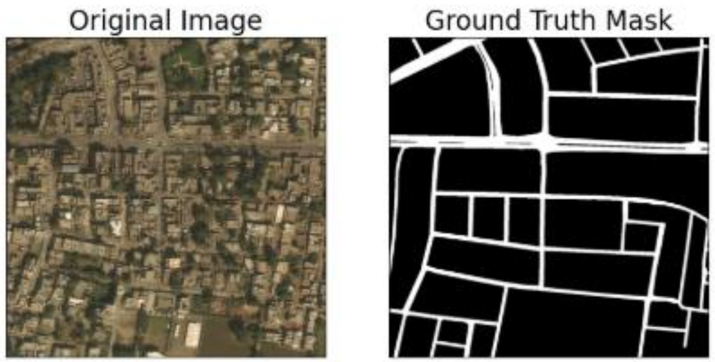
Dataset Source:

Kaggle, containing high-resolution satellite images with road network annotations (https://www.kaggle.com/datasets/balraj98/deepglobe-road-extraction-dataset/data).

- Images: 6,226
- Masks: 6,226

Sample Images & Masks





Detail Data Preprocessing (1/2)

Import Libraries

- Numpy, pandas, torch, cv2, matplotlib
- Albumentations for data augmentation
- Segmentation_models_pytorch for deep learning model

```
import os, cv2
import numpy as np
import pandas as pd
import random, tqdm
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
import warnings
warnings.filterwarnings("ignore")

import torch
import torch.nn as nn
from torch.utils.data import DataLoader

import albumentations as album
```

```
!pip install -q -U segmentation-models-pytorch albumentations > /dev/null
# import segmentation_models_pytorch as smp
import segmentation_models_pytorch as smp
from segmentation_models_pytorch import utils
```

Load the Dataset

- Dataset sourced from
- Metadata file: metadata.csv
- Load using Pandas: pd.read_csv()
- Filter for training data

```
DATA_DIR = "/kaggle/input/dataset-new/Dataset/Satellite Road Extraction Dataset"
print(os.listdir(DATA_DIR))  # Check available files

metadata_df = pd.read_csv(os.path.join(DATA_DIR, 'metadata.csv'))
metadata_df = metadata_df[metadata_df['split']=='train']
metadata_df = metadata_df[['image_id', 'sat_image_path', 'mask_path']]
metadata_df['sat_image_path'] = metadata_df['sat_image_path'].apply(lambda img_pth: os.path.join(DATA_DIR, img_pth))
metadata_df['mask_path'] = metadata_df['mask_path'].apply(lambda img_pth: os.path.join(DATA_DIR, img_pth))
```

SASKATCHEWAN POLYTECHNIC

Detail Data Preprocessing (2/2)

Path Adjustments

- Convert relative paths to absolute paths

```
DATA_DIR = "/kaggle/input/dataset-new/Dataset/Satellite Road Extraction Dataset"
print(os.listdir(DATA_DIR))  # Check available files

metadata_df = pd.read_csv(os.path.join(DATA_DIR, 'metadata.csv'))
metadata_df = metadata_df[metadata_df['split']=='train']
metadata_df = metadata_df[['image_id', 'sat_image_path', 'mask_path']]
metadata_df['sat_image_path'] = metadata_df['sat_image_path'].apply(lambda img_pth: os.path.join(DATA_DIR, img_pth))
metadata_df['mask_path'] = metadata_df['mask_path'].apply(lambda img_pth: os.path.join(DATA_DIR, img_pth))
```

Data Augmentation

- Use **Albumentations** for transformations
  - HorizontalFlip
  - Normalize trained images
- Ensure model generalization

```
def get_training_augmentation():
    train_transform = [
        album.HorizontalFlip(p=0.5),
        album.VerticalFlip(p=0.5),
    ]
    return album.Compose(train_transform)


def to_tensor(x, **kwargs):
    return x.transpose(2, 0, 1).astype('float32')
```

```
def get_preprocessing(preprocessing_fn=None):
    """Construct preprocessing transform
    Args:
        preprocessing_fn (callable): data normalization function
            (can be specific for each pretrained neural network)
    Return:
        transform: albumentations.Compose
    """
    _transform = []
    if preprocessing_fn:
        _transform.append(album.Lambda(image=preprocessing_fn))
    _transform.append(album.Lambda(image=to_tensor, mask=to_tensor))

    return album.Compose(_transform)
```

Modelling

Model Architecture - DeepLabV3+

- **Base Model:** DeepLabV3+ with ResNet-50 backbone
- **Key Features**
  - Atrous Spatial Pyramid Pooling (ASPP) for multi-scale context
  - Encoder-Decoder structure for high-resolution segmentation

Training Configuration

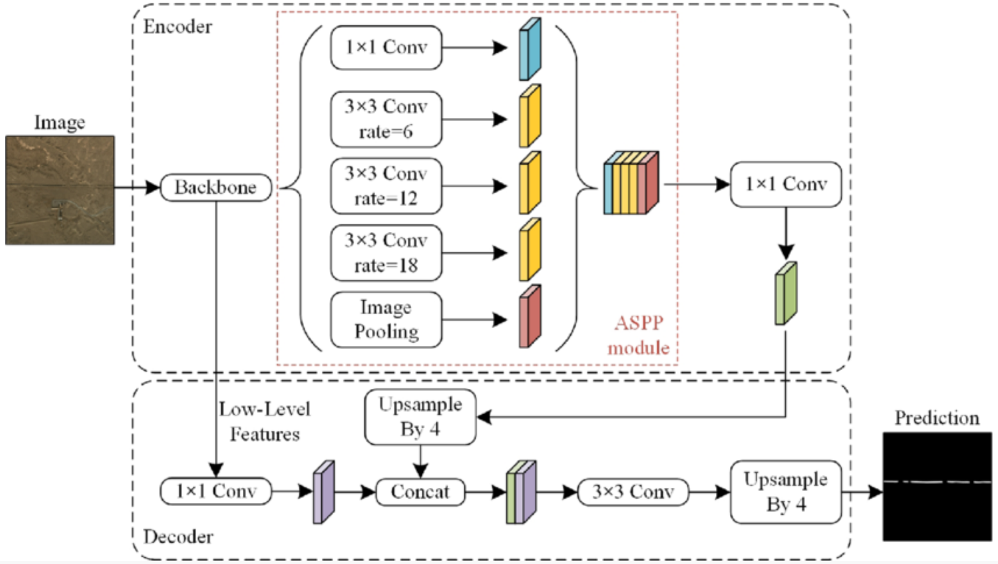
- **Optimizer:** Adam with learning rate scheduling.
- **Batch Size:** 2
- **Epochs:** 3
- **Augmentations:** Applied to improve generalization.
- **Hardware:** Trained on GPU

Model Evaluation Metrics

- **Metrics Used**
  - Intersection over Union (IoU)
  - Dice Coefficient
  - Pixel Accuracy

Results Visualization

- Display sample predictions vs. ground truth masks




```python
# define optimizer
optimizer = torch.optim.Adam([
    dict(params=model.parameters(), lr=0.00008),
])

# define learning rate scheduler (not used in this NB)
lr_scheduler = torch.optim.lr_scheduler.CosineAnnealingWarmRestarts(
    optimizer, T_0=1, T_mult=2, eta_min=5e-5,
)
```

SASKATCHEWAN POLYTECHNIC

Results and Analysis

	DeepLabV3+	U-Net
Architecture	**Semantic segmentation model** designed for pixel-wise classification	**Fully convolutional network** designed for image segmentation
Loss Function	Cross-Entropy Loss (used during training)	Binary Cross-Entropy Loss.
Metrics	IoU and Dice Coefficient Comparison	Accuracy, IoU and Dice Coefficient Comparison

Results and Analysis

(Accuracy and IoU Comparison)

U-Net: The **accuracy (97.14%)** is high, but accuracy alone can be misleading because it may not capture cases where small road segments are misclassified, especially if the roads are thin or fragmented. It's true when Mean IoU is only 49.57%.

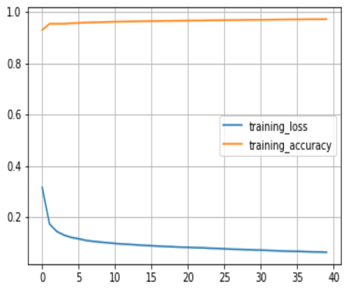
DeepLabV3+: The **Mean IoU (95.18%)** indicates that DeepLabV3+ has a very high overlap between the predicted road segments and the ground truth. This means DeepLabV3+ is more precise in detecting the road areas, especially thin roads and small segments, which is harder for U-Net to capture effectively.

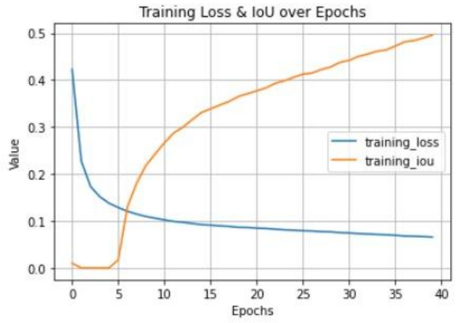
U-Net

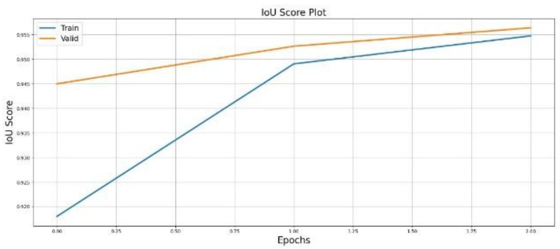
```
Epoch 40/40
195/195 [==============================] - 19s 96ms/step - loss: 0.0627 - accuracy: 0.9714
```

In [9]:

```
plt.plot(retVal.history['loss'], label = 'training_loss')
plt.plot(retVal.history['accuracy'], label = 'training_accuracy')
plt.legend()
plt.grid(True)
```







DeepLabV3+

[23]:

```
train_logs_df = pd.DataFrame(train_logs_list)
valid_logs_df = pd.DataFrame(valid_logs_list)
train_logs_df.T
```

ut[23]:

	0	1	2
dice_loss	0.095968	0.035963	0.028223
iou_score	0.910660	0.938775	0.948517

In [24]:

```
plt.figure(figsize=(20,8))
plt.plot(train_logs_df.index.tolist(), train_logs_df.iou_score.tolist(), lw=3, label = 'Trai
n')
plt.plot(valid_logs_df.index.tolist(), valid_logs_df.iou_score.tolist(), lw=3, label = 'Vali
d')
plt.xlabel('Epochs', fontsize=20)
plt.ylabel('IoU Score', fontsize=20)
plt.title('IoU Score Plot', fontsize=20)
plt.legend(loc='best', fontsize=16)
plt.grid()
plt.savefig('iou_score_plot.png')
plt.show()
```

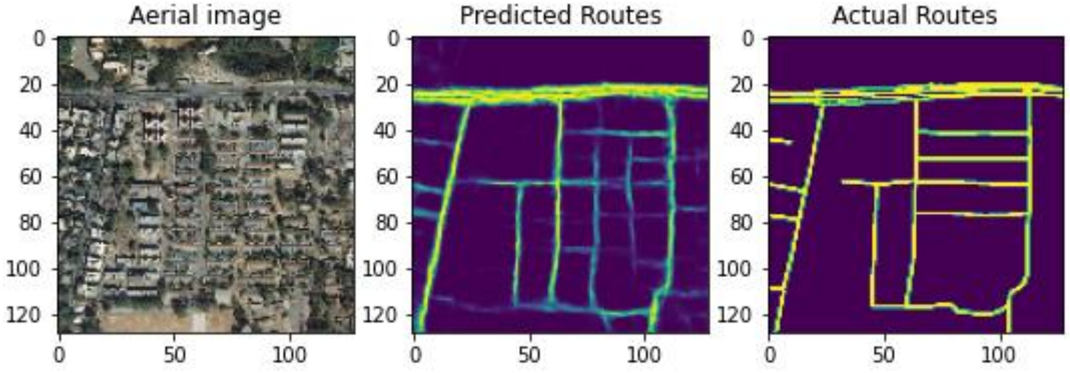
SASKATCHEWAN POLYTECHNIC

Results and Analysis

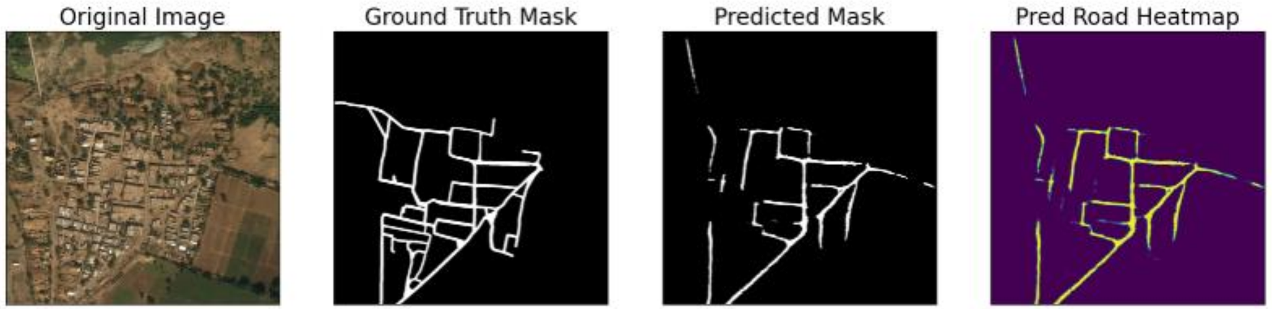
(Sample Segmented Images vs. Ground Truth)

Term	Representation	Purpose
Original Mask	Raw dataset mask	Used for training/testing
Ground Truth Mask	Color-coded segmentation	Help in visualization & comparison
Predicted Mask	Model's output segmentation	Used for evaluation
One-Hot Encoded Mask	Multi-channel binary representation	Help in model training
Predicted Road Heatmap	Probability distribution of "road" pixels	Used for model confidence visualization

U-Net



DeepLabV3+



SASKATCHEWAN POLYTECHNIC

Results and Analysis

(Loss Comparison and Model Preference)

Loss Comparison:

- Both **U-Net** and **DeepLabV3+** has **the low loss** with **0.0658** and **0.0261** respectively but DeepLabV3+ is better because it's able to capture the road regions well and overlap with the ground truth effectively.

Model Preference:

- If your **goal is high precision in road segmentation**, DeepLabV3+ with its **high IoU score** would likely give more accurate road boundaries, even for thin or fragmented roads.
- **U-Net** performs well in terms of **accuracy** and might be preferred if **real-time inference speed** or **less complexity** is needed, especially if you want to deploy the model with fewer computational resources.

SASKATCHEWAN POLYTECHNIC

Results and Analysis

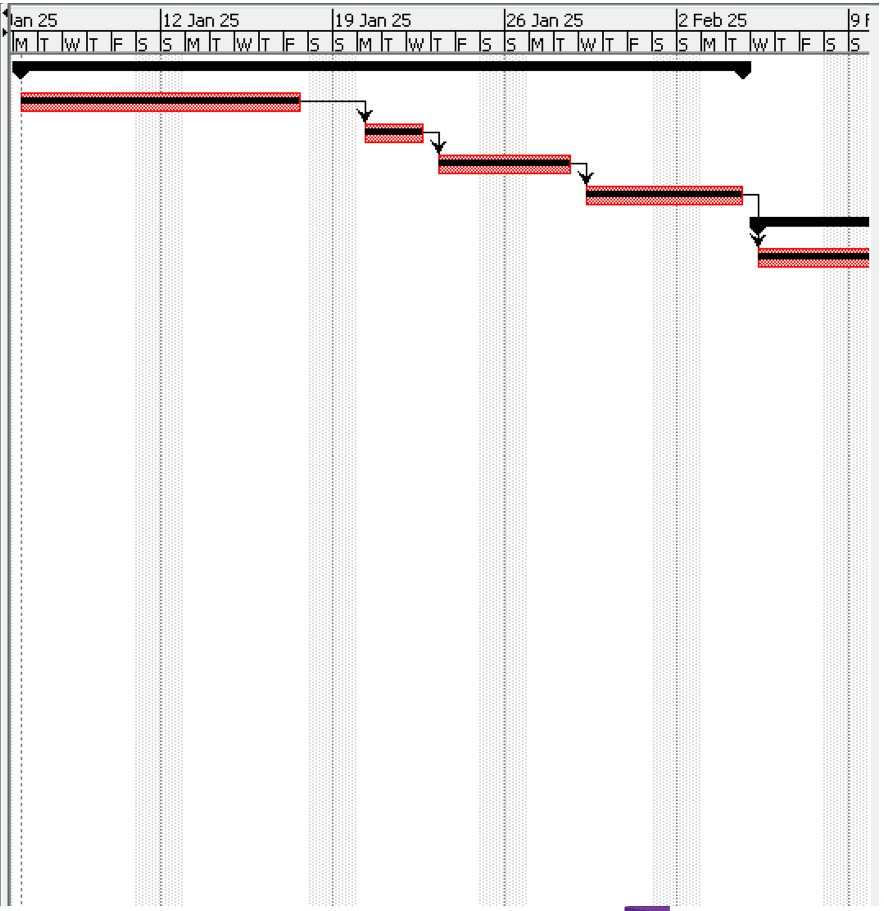
(Conclusion)

Considerations:

- **U-Net** might be performing well on pixel classification but might struggle with finer details or boundaries. The high accuracy is a positive indicator, but **IoU** and **Dice Loss** suggest DeepLabV3+ captures the road segmentation better, especially for smaller or thin road areas.
- **DeepLabV3+** seems to have the edge when it comes to fine-tuned segmentation (judging by its **high Mean IoU** and **low Dice Loss**), which is critical for more complex road segmentation tasks.

Project Progress Self-Assessment

	Indicators	Name	Duration	Start	Finish	Predecessors
1	✓	**⊟ Project Planning and Data Collection**	22 days	**1/6/25 8:00 AM**	**2/4/25 5:00 PM**	
2	✓	Research and finalize topic with suitable datasets	10 days	1/6/25 8:00 AM	1/17/25 5:00 PM	
3	✓	Define project goals, scope, & success metrics	3 days	1/20/25 8:00 AM	1/22/25 5:00 PM	2
4	✓	Prepare high level proposal & presentation	4 days	1/23/25 8:00 AM	1/28/25 5:00 PM	3
5	✓	Set up the development environment	5 days	1/29/25 8:00 AM	2/4/25 5:00 PM	4
6	✓	**⊟ Data Preprocessing and Exploration**	13 days	**2/5/25 8:00 AM**	**2/21/25 5:00 PM**	
7	✓	Load and explore the dataset (Visualize satellite images & road mask)	5 days	2/5/25 8:00 AM	2/11/25 5:00 PM	5
8	✓	Preprocess the data (Resize, normalize, split data)	5 days	2/12/25 8:00 AM	2/18/25 5:00 PM	7
9	✓	Augment the dataset (Rotation, flipping, cropping)	3 days	2/19/25 8:00 AM	2/21/25 5:00 PM	8
10	✓	**⊟ Model Selection and Baseline Development**	10 days	**2/24/25 8:00 AM**	**3/7/25 5:00 PM**	
11	✓	Select model and implement a baseline model	5 days	2/24/25 8:00 AM	2/28/25 5:00 PM	9
12	✓	Train and evaluate the baseline model	3 days	3/3/25 8:00 AM	3/5/25 5:00 PM	11
13	✓	Prepare Presentation and Report for Mid-Term	2 days	3/6/25 8:00 AM	3/7/25 5:00 PM	12
14		**⊟ Model Optimization/Improvement**	11 days	**3/10/25 8:00 AM**	**3/24/25 5:00 PM**	
15		Analyze baseline performance to optimize & improve model	5 days	3/10/25 8:00 AM	3/14/25 5:00 PM	13
16		Implement, train and evaluate the optimized/improved model	5 days	3/17/25 8:00 AM	3/21/25 5:00 PM	15
17		Compare results among the baseline & optimized/improved model	1 day	3/24/25 8:00 AM	3/24/25 5:00 PM	16
18		**⊟ Validation, Testing**	6 days	**3/25/25 8:00 AM**	**4/1/25 5:00 PM**	
19		Perform cross-validation to ensure model robustness	2 days	3/25/25 8:00 AM	3/26/25 5:00 PM	17
20		Test the model on the test set & compute final metrics	2 days	3/27/25 8:00 AM	3/28/25 5:00 PM	19
21		Visualize predictions, analyze failure cases & identify potential improvements	2 days	3/31/25 8:00 AM	4/1/25 5:00 PM	20
22		**⊟ Deployment**	4 days	**4/2/25 8:00 AM**	**4/7/25 5:00 PM**	
23		Convert the trained model to a deployable format	2 days	4/2/25 8:00 AM	4/3/25 5:00 PM	21
24		Develop a simple pipeline for inference & test	2 days	4/4/25 8:00 AM	4/7/25 5:00 PM	23
25		**⊟ Finalization and Documentation**	4 days	**4/8/25 8:00 AM**	**4/11/25 5:00 PM**	
26		Prepare a detailed project report & presentation	2 days	4/8/25 8:00 AM	4/9/25 5:00 PM	24
27		Conclude the project with programming codes & reports	2 days	4/10/25 8:00 AM	4/11/25 5:00 PM	26



References (1/2)

- **Liu, Y., Guo, Y., Zhang, F., & Wang, X. (2024).** *A Novel Network Framework on Simultaneous Road Segmentation and Vehicle Detection for UAV Aerial Traffic Images.***Sensors, 24(11), 3606.**
- **Mnih, V., & Hinton, G. (2018).** "Learning to Detect Roads in High-Resolution Aerial Images." *Neural Information Processing Systems (NeurIPS)*.
- **Sun, H., Wu, Y., & Zhang, J. (2021).** "Improved Satellite Road Extraction Using DeepLabV3+ with Attention Mechanisms." *IEEE Geoscience and Remote Sensing Letters*.

SASKATCHEWAN POLYTECHNIC

References (2/2)

- **Bahrampour, S., Ramakrishnan, N., Schott, L., & Shah, M. (2015).** *Comparative Study of Deep Learning Software Frameworks.* arXiv preprint arXiv:1511.06435.
- **Dubovikov, K. (2018).** *PyTorch vs TensorFlow — Spotting the Difference.* Towards Data Science.
- **Al-Bdour, G., Al-Qurran, R., Al-Ayyoub, M., & Shatnawi, A. (2019).** *A Detailed Comparative Study of Open Source Deep Learning Frameworks.* arXiv preprint arXiv:1903.00102.