Multi-Source Traffic Scene Understanding Using Deep Learning

Akhilesh Negi, 221AI008
Aishini Bhattacharjee, 221AI007

Introduction

Traffic scene understanding is critical for modern urban planning, intelligent transportation systems, and emergency response. This project explores deep learning techniques to tackle key aspects of traffic analysis, namely:

1. Road extraction
2. Vehicle detection
3. Accident recognition

By processing diverse imaging sources — such as aerial and surveillance imagery — the project aims to build reliable pipelines capable of analyzing and interpreting complex traffic environments.

Motivation

Traditional traffic monitoring systems often fall short in providing real-time, comprehensive insights into road conditions, vehicle movements, and accident occurrences. This project aims to harness deep learning techniques to bridge these gaps, enhancing traffic safety and efficiency through intelligent analysis of diverse visual data sources.

Literature Review

Paper	Methodology	Merits	Demerits
Drone Image Segmentation (2020) (Journal Paper)	Mapped vegetation in Clara Bog using drone images. Compared ML (Random Forest + graph cut) and DL (SegNet, UNet, PSPNet with transfer learning).	DL models (≈90% accuracy) outperformed ML (≈85%). Transfer learning reduced data dependence. High-resolution drone imagery captured detailed vegetation maps.	DL is resource-intensive and time-consuming. Affected by sunlight variation; pre-processing needed. High costs for hardware and training.
Review of DL based Image Segmentation (2023) (Journal Paper)	Systematic review of DL models for polyp segmentation in colorectal cancer detection. Explored CNNs, encoder-decoder models (e.g., UNet, SegNet), attention-based models, and GANs. Analyzed datasets (e.g., Kvasir-SEG) and evaluation metrics (Dice coefficient, IoU).	Improved early detection accuracy using DL models. Comprehensive dataset review enhances reproducibility. Attention and GAN models address data scarcity and variability issues.	High computational requirements for training. Limited generalization due to dataset biases. Performance affected by polyp size and texture variability.

Literature Review

Paper	Methodology	Merits	Demerits
Road Traffic Monitoring using Deep Neural Networks (2021) (Journal Paper)	Utilized UAV videos for vehicle detection, tracking (EfficientDet + SORT), and speed estimation. Lane segmentation (U-Net + Lane-Net) calculated image scale via lane width standards. Tested on four road sections with 97.6% recall, 94.7% precision, and RMSE of 5.27 km/h.	Accurate speed and flow analysis without prior road/vehicle data. Effective in real-time monitoring for both traffic directions. Scalable to wide road areas with UAV flexibility.	Dependent on consistent UAV altitude and video quality. High computational cost for DL models. Limited resolution for distant or crowded roads.
Image Segmentation for 2D Searching (2019) (Journal Paper)	Utilized Mask R-CNN for segmenting and detecting 2D materials (graphene, MoS2, WTe2, hBN) in optical microscope images. Leveraged transfer learning from MS-COCO dataset and trained on 2000 annotated images. Integrated with a motorized optical microscope for automated material searching.	Accurate segmentation for 2D material detecTion. Effective automation with motorized optical microscope.	Limited by dataset variety and annotations. Performance depends on high-quality optical images.

Literature Review

Paper	Methodology	Merits	Demerits
Image Segmentation in the JPEG Compressed Domain (2021) (Conference Paper)	Proposed a modified Eff-UNet model for segmentation directly on JPEG compressed DCT coefficients, avoiding decompression. Removed initial encoder layers to process smaller DCT input size (40x40). Evaluated on Oxford-IIIT Pet and IDD-Lite datasets using semantic segmentation metrics.	Faster training (reduced layers) and efficient segmentation directly from compressed data. Maintains competitive accuracy compared to pixel-based methods. Reduces computational overhead by avoiding full image decompression.	Lower accuracy (mIoU ~68.59%) for small objects in dense images. Limited generalizability to varied compressed formats. Slight loss in detail due to reduced DCT coefficient resolution.
Deep Learning Based Vehicle Detection From Aerial Images (2020) (Conference Paper)	Developed a vehicle detection model using **YOLOv3**, enhanced by Faster R-CNN for feature extraction during training. Trained on Munich Vehicle Dataset, Google Earth, and DJI drone images. Evaluated detection performance for various altitudes (30m–75m+), including complex scenarios.	Improved YOLO detection accuracy by 3.2% with Faster R-CNN aid. Efficient for real-time applications due to YOLO's speed. Handles varying vehicle sizes and complex backgrounds.	Struggles with dark vehicles in shadowy regions. Lower precision in high-altitude images with small vehicles. Limited exploration of newer YOLO versions (e.g., YOLOv12).

Literature Review

Paper	Methodology	Merits	Demerits
Real-time Traffic Surveillance and Detection using Deep Learning and Computer Vision Techniques (2023) (Conference Paper)	YOLOv5 and YOLOv7 for real-time detection via web app. Fine-tuned models with augmented datasets. Speed detection using YOLOv7 and contour calculations.	Accurate detection with customization options. Efficient and scalable real-time monitoring.	Limited to specific vehicles and conditions. Issues in low light and dependent on detection accuracy.
A Systematic Review of Drone Based Road Traffic Monitoring System (2022) (Conference Paper)	Systematically reviewed drone-based traffic monitoring systems using deep learning for detection, tracking, and counting. Analyzed drone datasets, frameworks, and preprocessing methods for urban and highway scenarios. Examined metrics like precision, recall, and MOTA for evaluating effectiveness.	Comprehensive analysis of existing frameworks and datasets. Highlights future research directions in smart city traffic management.	Lacks practical deployment results or real-world implementation insights. Limited focus on addressing challenges like privacy and computational constraints.

Literature Review

Paper	Methodology	Merits	Demerits
Real-Time Traffic Analysis using Deep Learning Techniques and UAV based Video (2019) (Conference Paper)	UAVs collected traffic data, analyzed using Mask RCNN for real-time tracking and segmentation. Vehicle speeds and counts estimated using bounding box movement and overlap calculations. Affine transformation used for pixel-to-space conversion, with preliminary metrics validated against manual analysis.	Accurate instance-based tracking and traffic metrics. Effective for congestion analysis using real-time UAV data.	Limited by fixed UAV and basic metrics. Accuracy issues in complex scenarios like traffic deadlocks.
Real-Time Ground Vehicle Detection in Aerial Infrared Imagery Based on Convolutional Neural Network (2018) (Journal Paper)	Used CNN for real-time ground vehicle detection in aerial infrared images. Built a UAV-based platform and created an aerial infrared vehicle dataset. Applied region proposal and non-maximum suppression for accurate detection.	Detects both stationary and moving vehicles in real urban environments. Achieves real-time performance with high precision and recall.	Limited by low-resolution and noisy infrared images. Performance may degrade in complex lighting and occlusion scenarios.

Outcomes of Literature Survey

1. **Segmentation-driven RoI extraction significantly boosts detection accuracy**, but incurs high computational overhead—especially for drone and high-resolution imagery. (Bisio et al., 2023)

2. **Transfer learning and attention mechanisms reduce annotation demands and improve robustness in varied environments**, yet remain sensitive to lighting changes and occlusions. (Bhatnagar et al., 2020) (Gupta and Mishra, 2024)

Outcomes of Literature Survey

3. **Real-time object detectors (YOLOv3/5/7, EfficientDet) achieve fast vehicle detection and tracking suitable for live monitoring**, but performance degrades for small targets, shadowed regions, and high-altitude views. (Byun et al., 2021) (Singh et al., 2021) (DIKBAYIR and Ïbrahim BÜLBÜL, 2020)

Problem Statement

To develop deep learning-driven solutions for comprehensive traffic scene understanding, focusing on road extraction, vehicle detection, and accident recognition across diverse imaging sources.

Objectives

1. Develop a model for Region of Interest (RoI) extraction from road images, focusing on road segmentation in satellite images and accident region identification in accident images.
2. Implement a vehicle detection model applied specifically to the extracted RoI to enhance detection accuracy and efficiency and use that for further tasks like accident detection.

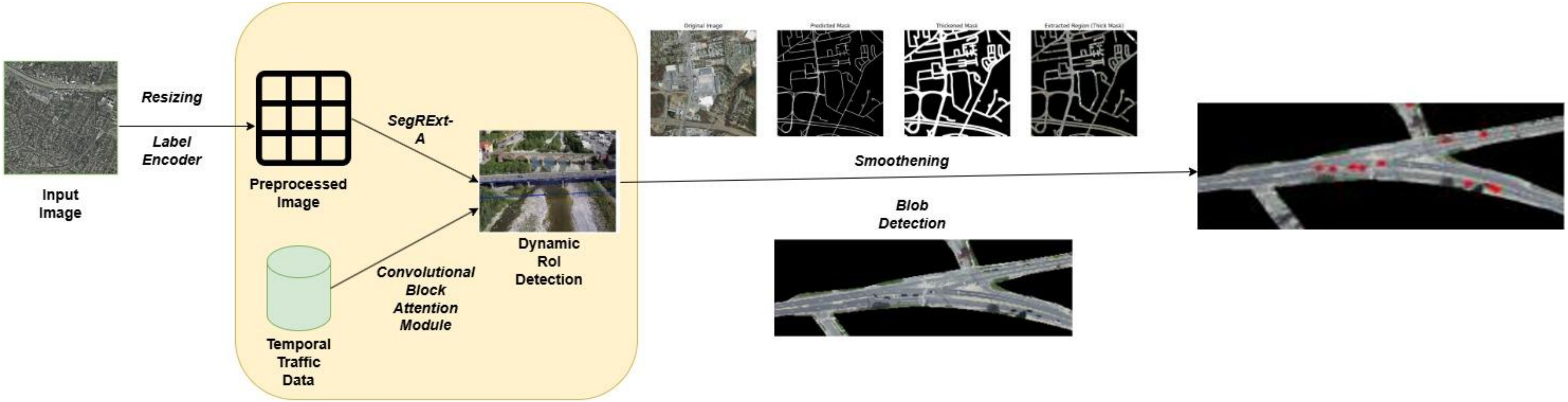
Practical Implications and Significance

1. Focusing on the Region of Interest (RoI) before vehicle detection reduces the processing load, which is crucial for real-time applications. In practice, this can lead to systems that run efficiently on embedded hardware, lowering operational costs compared to fixed camera networks that cover limited areas.
2. With accurate speed estimation and vehicle tracking, authorities could use the system to detect abnormal driving patterns or sudden congestion. This real-time insight can trigger timely interventions, enhancing road safety and supporting emergency services during accidents or natural disasters.

Dataset Description

1. Massachusetts Road Dataset: consists of 1171 aerial images of the state of Massachusetts. Each image is 1500×1500 pixels in size, covering an area of 2.25 square kilometers. The dataset covers a wide variety of urban, suburban, and rural regions and covers an area of over 2600 square kilometers.
2. Accident Detection From CCTV Footage: The dataset contains frames captured from Youtube Videos involving accidents. The images of accidents and non accidents are split into train,test and val folders.
3. Satellite Imagery Multi-vehicles Dataset (SIMD): Contains Satellite images of vehicles along with corresponding bounding boxes for vehicles.

Methodology



Input Image

Resizing

Label Encoder

Preprocessed Image

SegRExt-A

Temporal Traffic Data

Convolutional Block Attention Module

Dynamic RoI Detection

Smoothening

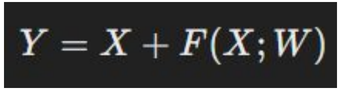
Blob Detection

Work Done: RoI Extraction

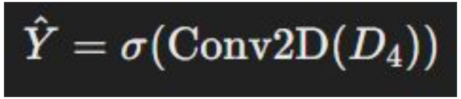
1. Loading and augmenting of dataset
2. Model architectures used: SegRExt-A, SegRExt-F
3. Image smoothening
4. Loss functions: BCE, Dice, EM
5. Metrics: IoU, Dice, Precision, Recall, FPS

SegRExt-A

1. Encoder: ResNet-50, Feature extraction using convolution and pooling layers.

   4 residual layers: 

   $Y = X + F(X; W)$

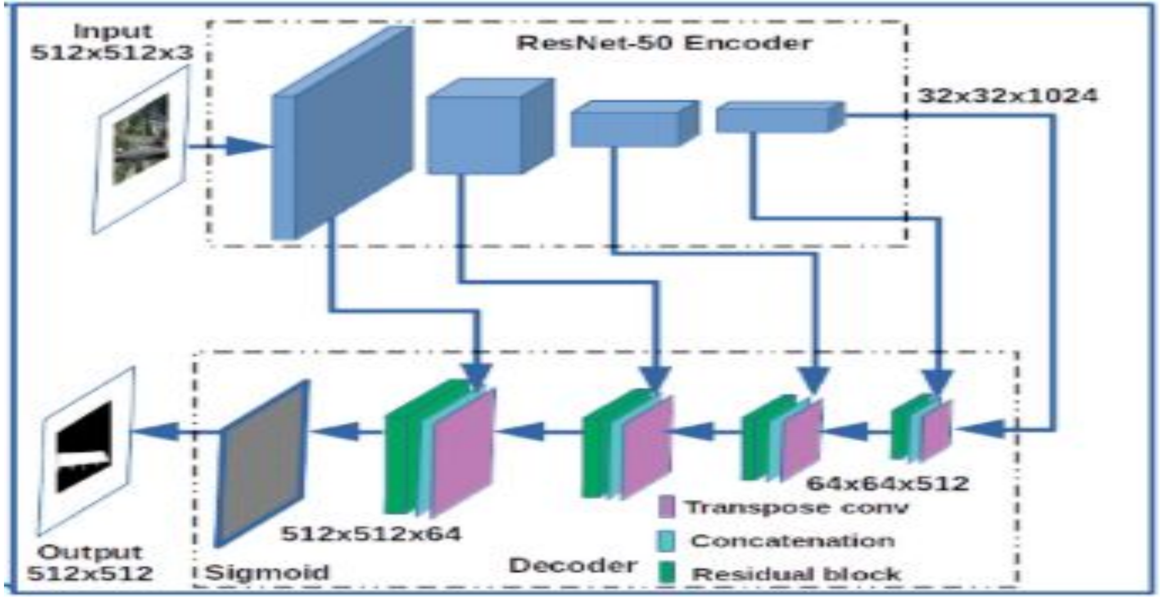
2. The decoder reconstructs the segmentation map by upsampling feature maps using transposed convolutions and residual connections.
3. Finally, the output is passed through a final convolution layer to get a single-channel segmentation mask. 

   $\hat{Y} = \sigma(\mathrm{Conv2D}(D_4))$

4. Final output:

   Binary segmentation mask: 1 if the pixel belongs to the RoI, 0 if not.

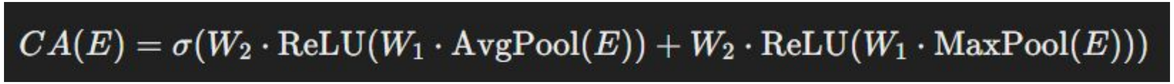
SegRExt A Architecture, Base Paper



SegRExt-F

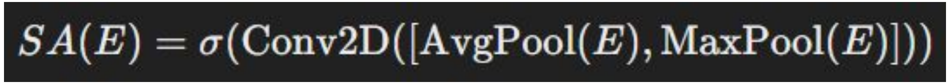
1. Encoder: MobileNet V3 (progressive downsampling to feature maps)
2. Convolutional Block Attention Module (CBAM):

Channel Attention:



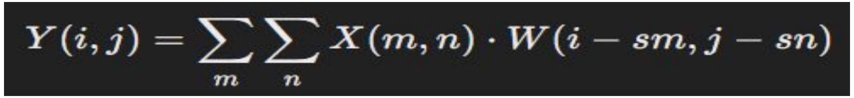
$$CA(E) = \sigma(W_2 \cdot \mathrm{ReLU}(W_1 \cdot \mathrm{AvgPool}(E)) + W_2 \cdot \mathrm{ReLU}(W_1 \cdot \mathrm{MaxPool}(E)))$$

Spatial Attention:

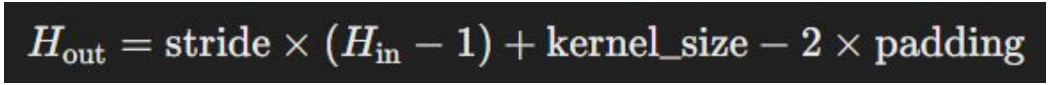


$$SA(E) = \sigma(\mathrm{Conv2D}([\mathrm{AvgPool}(E), \mathrm{MaxPool}(E)]))$$

3. Decoder: Upsampling

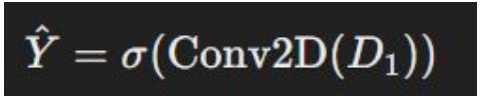


$$Y(i,j) = \sum_m \sum_n X(m,n) \cdot W(i - sm, j - sn)$$



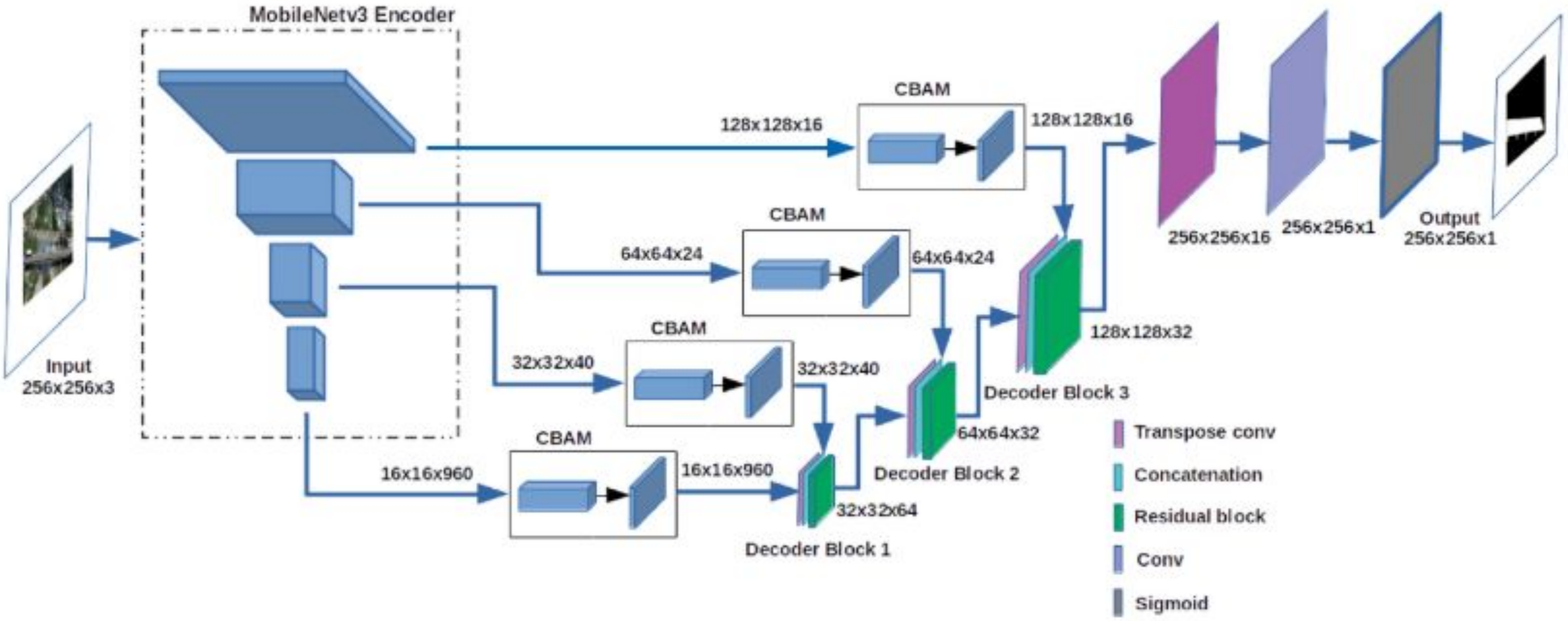
$$H_{\mathrm{out}} = \mathrm{stride} \times (H_{\mathrm{in}} - 1) + \mathrm{kernel\_size} - 2 \times \mathrm{padding}$$

4. Segmentation mask by using convolution and sigmoid activation



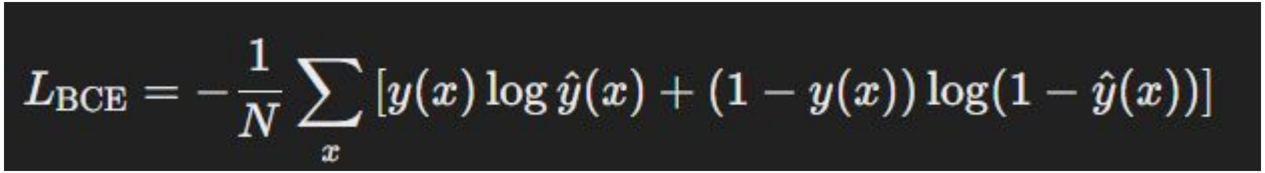
$$\hat{Y} = \sigma(\mathrm{Conv2D}(D_1))$$

SegRExt F Architecture, Base Paper



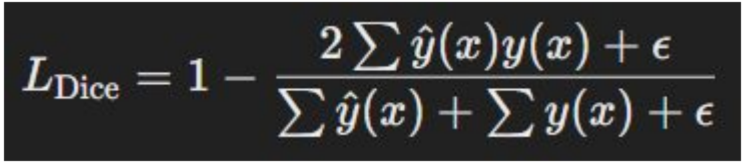
Loss Functions

1. Binary cross entropy: For classifying pixels by assigning probabilities



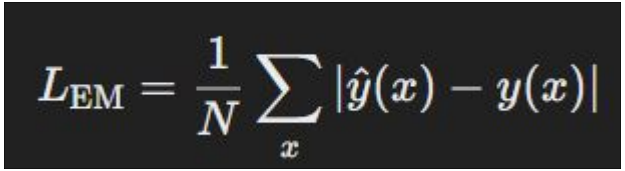
$$L_{\text{BCE}} = -\frac{1}{N}\sum_{x}[y(x)\log\hat{y}(x) + (1 - y(x))\log(1 - \hat{y}(x))]$$

2. Dice loss: Measures overlap between predicted mask and ground truth



$$L_{\text{Dice}} = 1 - \frac{2\sum\hat{y}(x)y(x) + \epsilon}{\sum\hat{y}(x) + \sum y(x) + \epsilon}$$

3. Enhanced matching loss: Tries to align the predicted mask with ground truth in terms of pixel distribution



$$L_{\text{EM}} = \frac{1}{N}\sum_{x}|\hat{y}(x) - y(x)|$$

Metrics

1. IoU: Measures overlap between prediction and ground truth

$$IoU = \frac{TP}{TP + FP + FN}$$

2. Precision and Recall

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

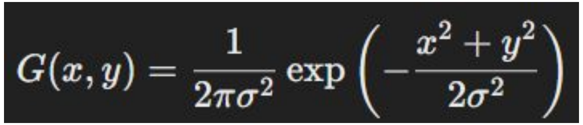
3. Frames per second (FPS): Measures speed in real time

$$\text{FPS} = \frac{\text{Total Images Processed}}{\text{Total Time Taken}}$$

Innovation: RoI Extraction

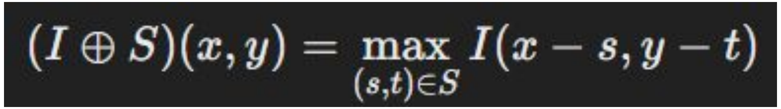
Problem: Even with advanced segmentation networks, the output masks may suffer from imprecise boundaries, which is critical when the RoI (e.g., a bridge) must be sharply delineated.

Solution: Noise reduction with Gaussian blur.

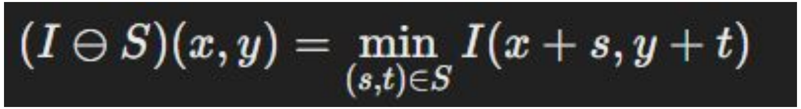


$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Morphological Closing (Dilation Followed by Erosion)

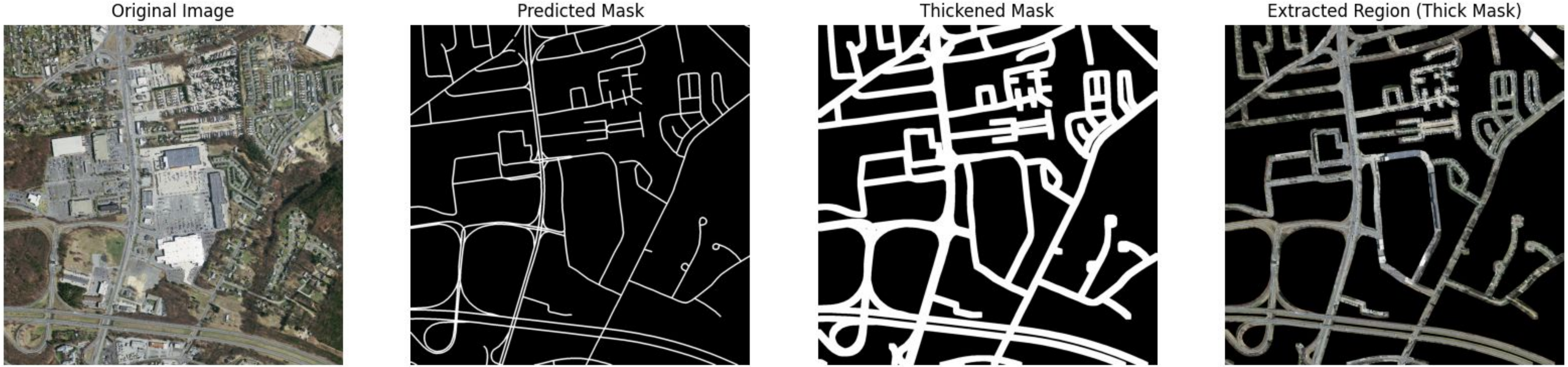


$$(I \oplus S)(x, y) = \max_{(s,t)\in S} I(x - s, y - t)$$



$$(I \ominus S)(x, y) = \min_{(s,t)\in S} I(x + s, y + t)$$

Convert the processed image back to a binary form, ensuring crisp and well-defined edges.

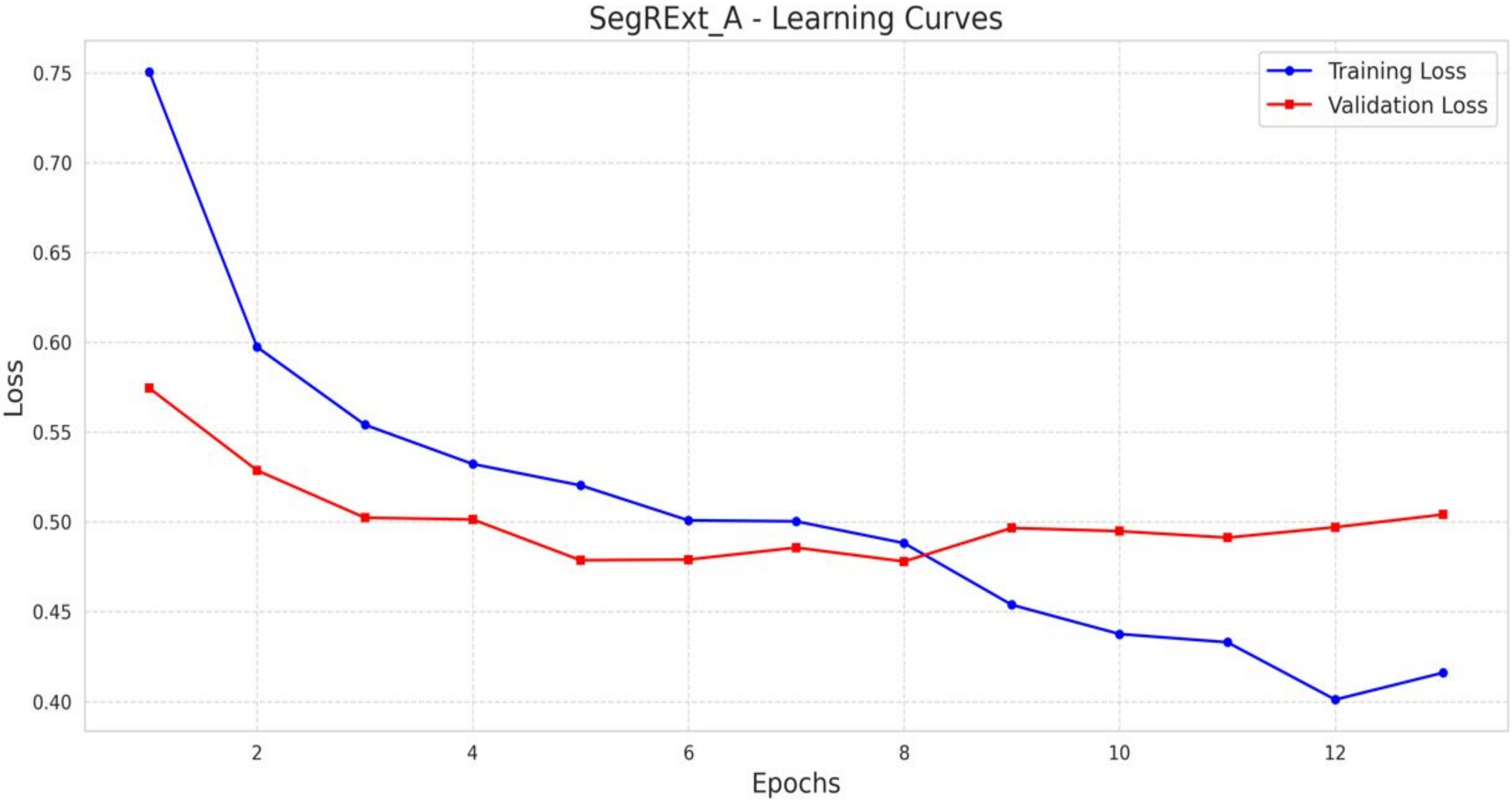


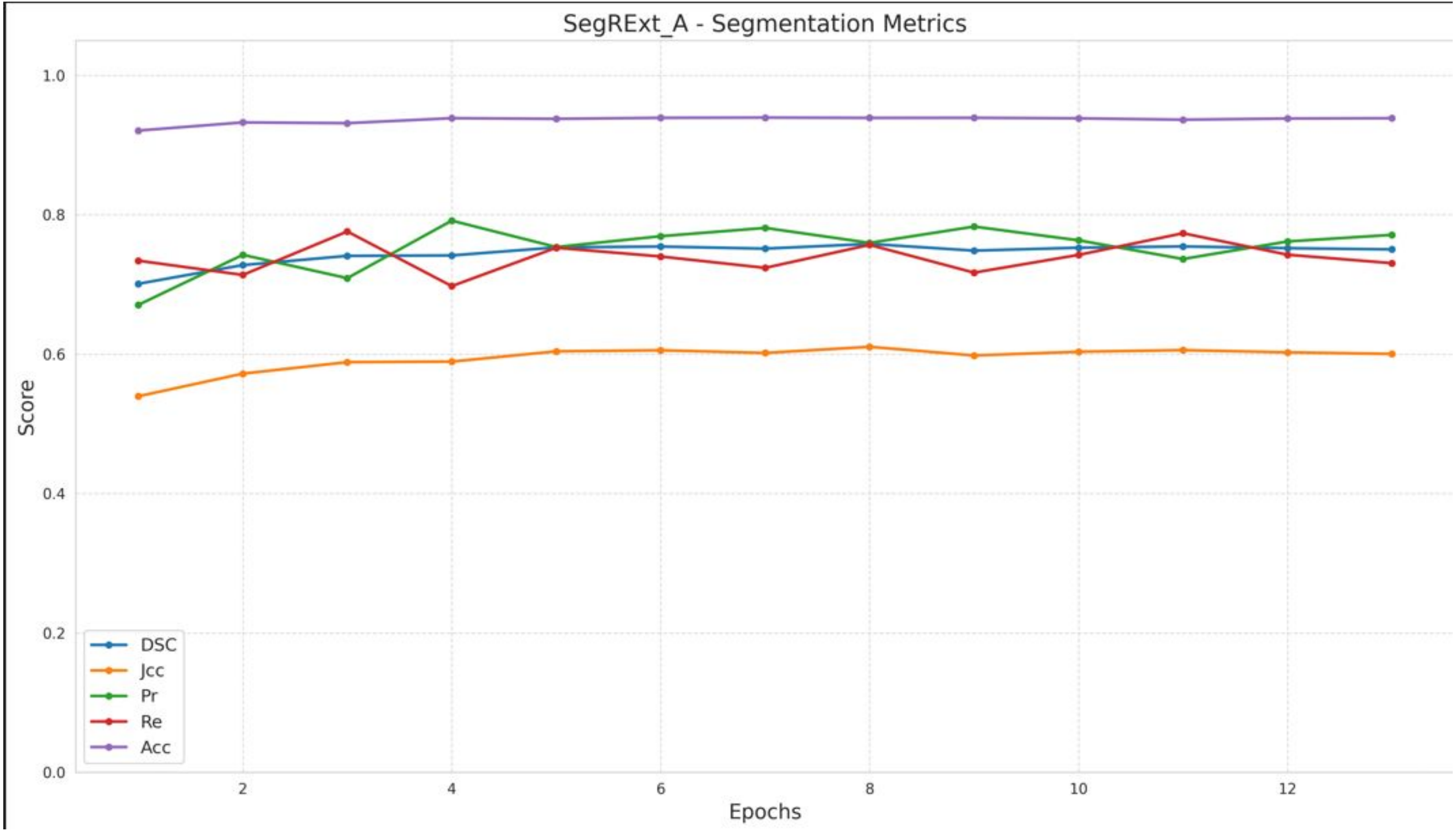
Original Image

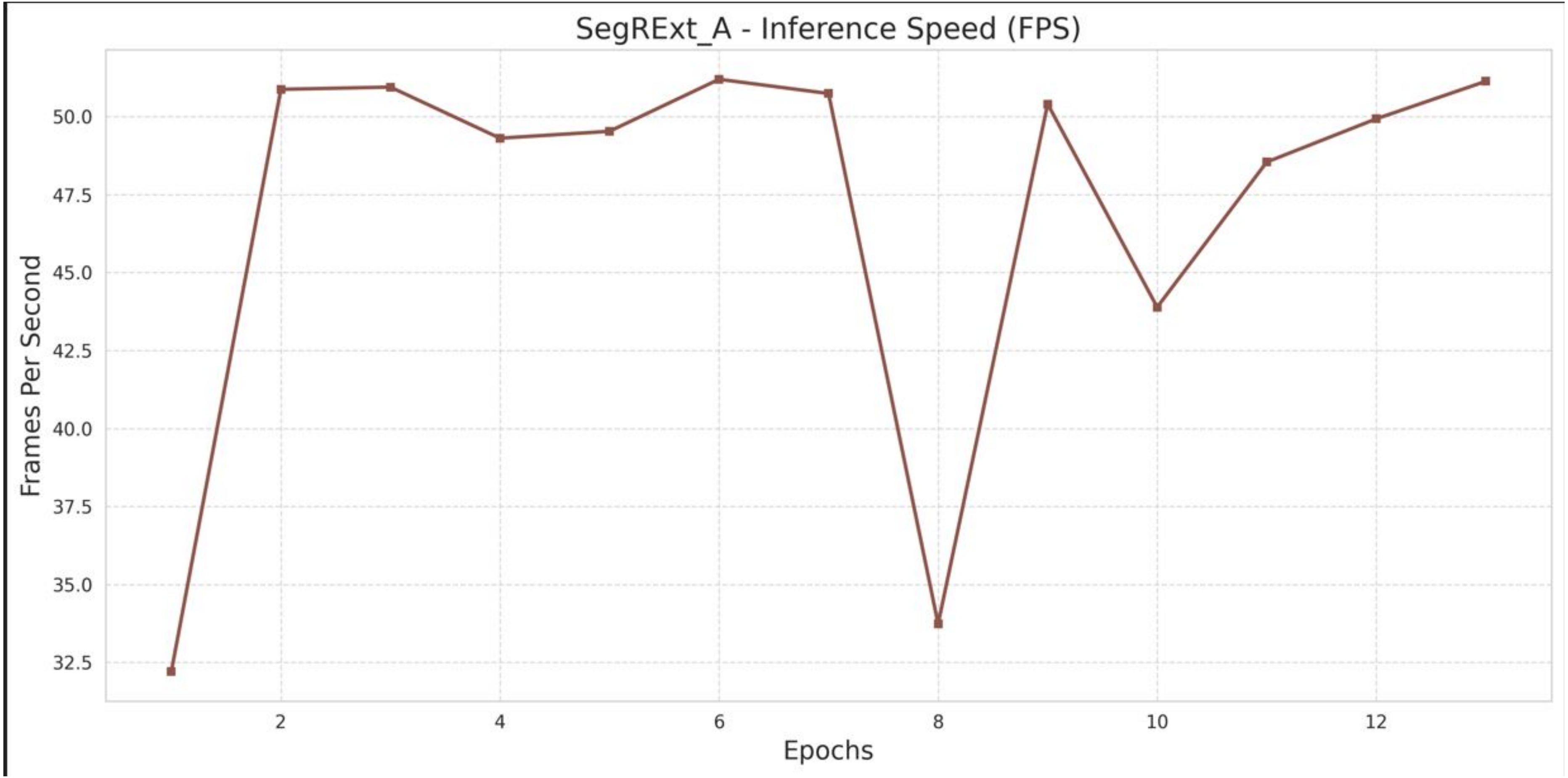
Predicted Mask

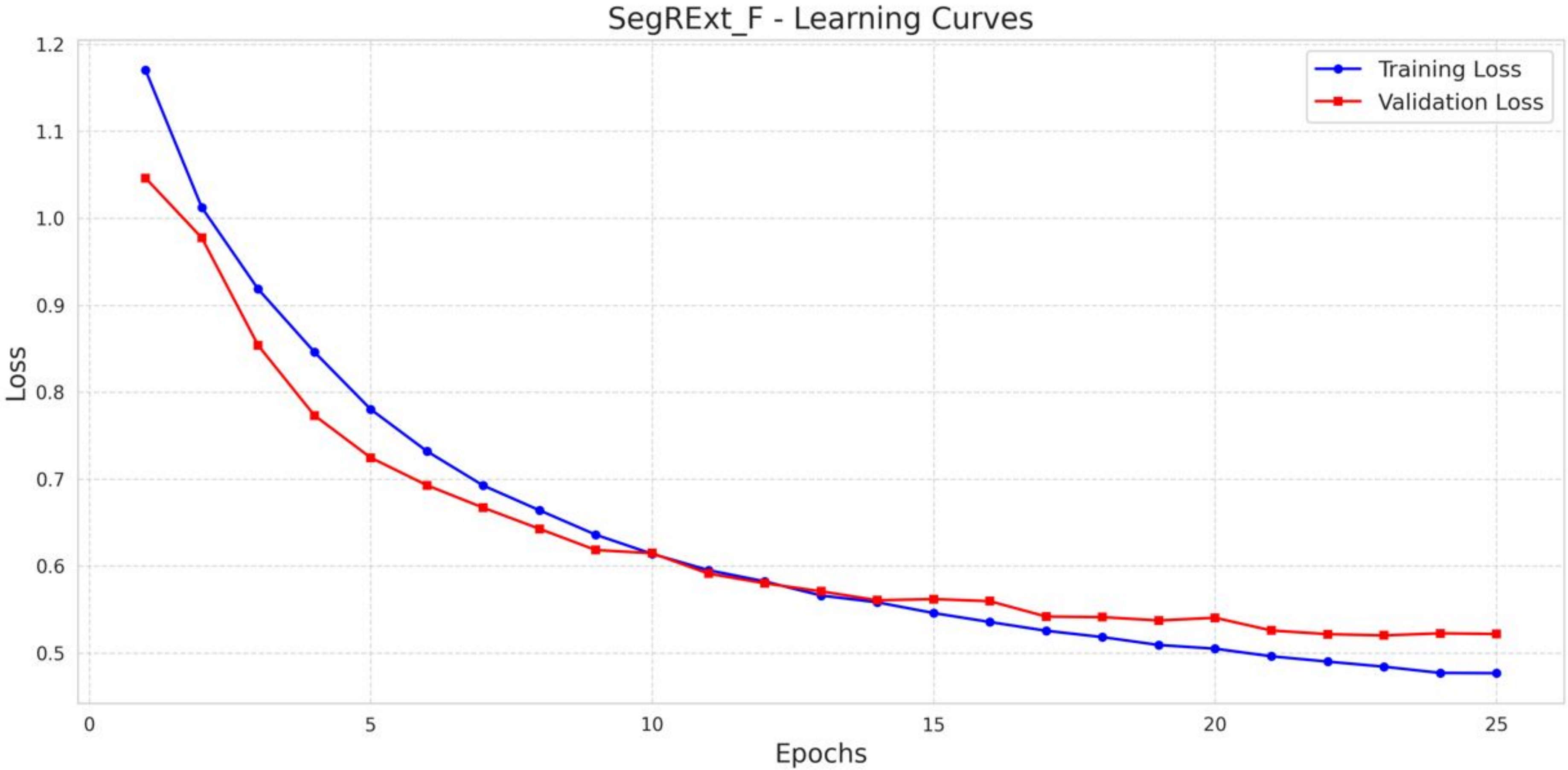
Thickened Mask

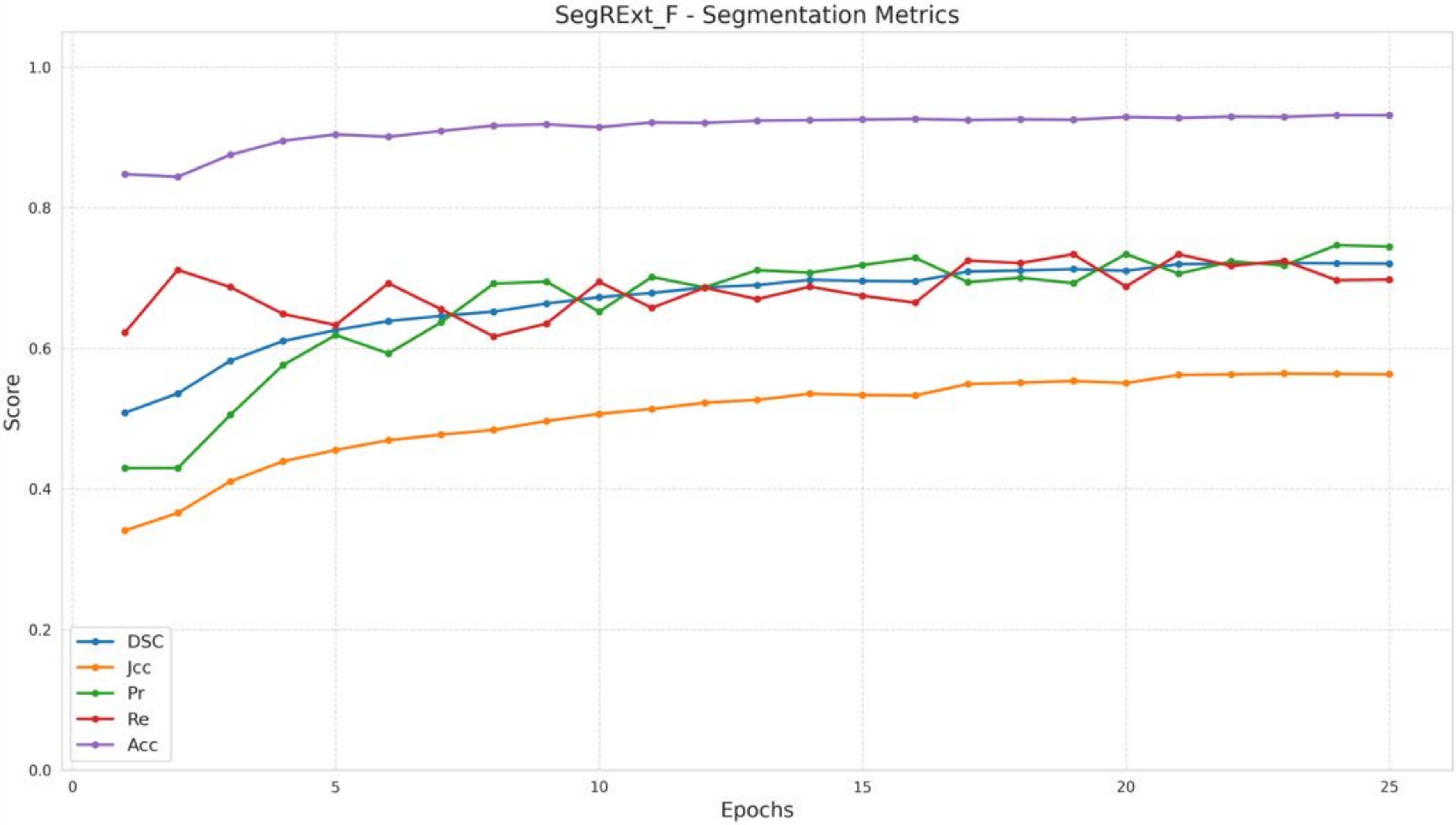
Extracted Region (Thick Mask)

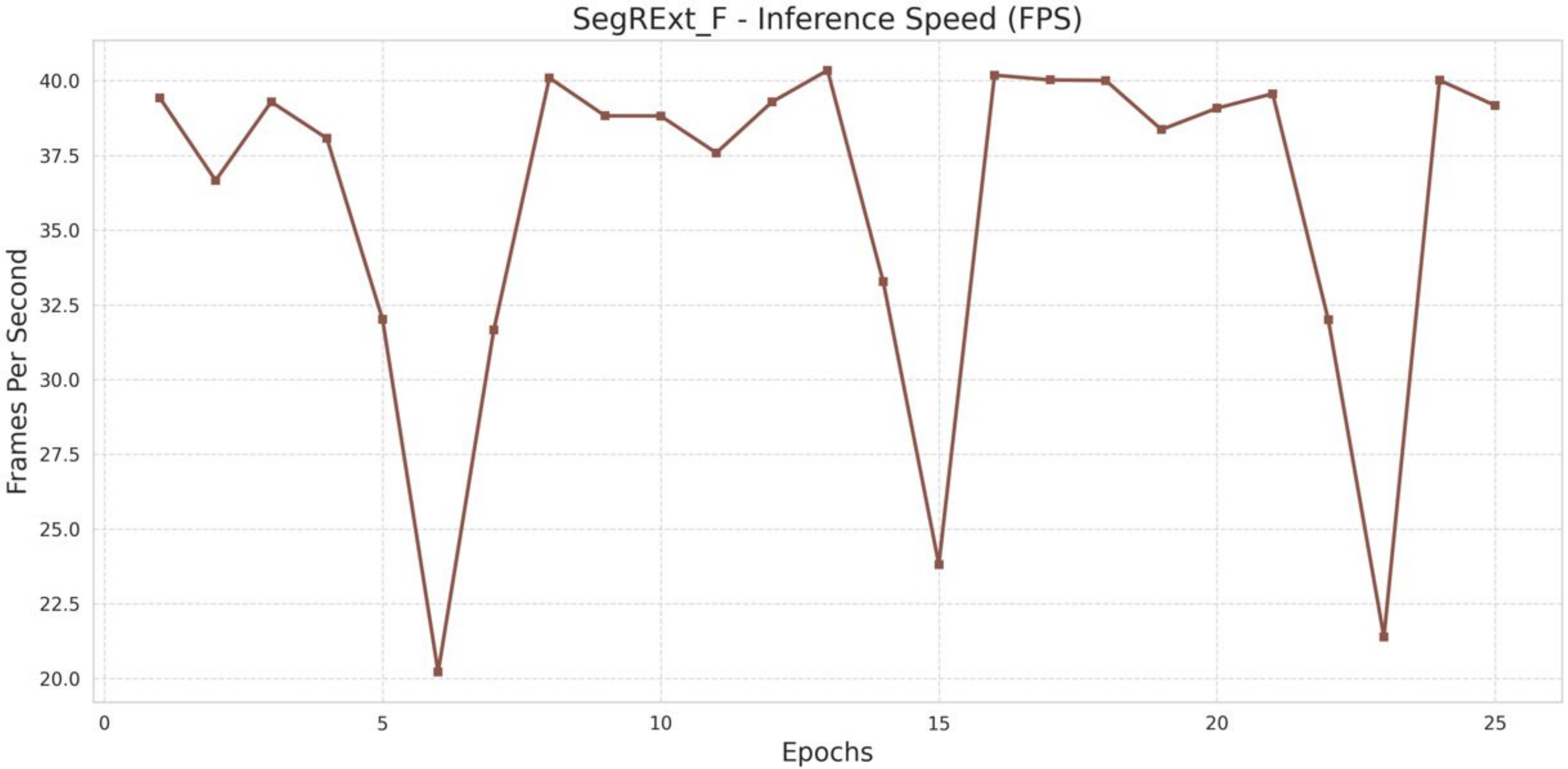

SegRExt_A - Learning Curves
Training Loss
Validation Loss
Loss
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
Epochs
2
4
6
8
10
12


SegRExt_A - Segmentation Metrics
Score
1.0
0.8
0.6
0.4
0.2
0.0
Epochs
2
4
6
8
10
12
DSC
Jcc
Pr
Re
Acc


SegRExt_A - Inference Speed (FPS)
Frames Per Second
50.0
47.5
45.0
42.5
40.0
37.5
35.0
32.5
2
4
6
8
10
12
Epochs


SegRExt_F - Learning Curves
Training Loss
Validation Loss
Loss
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0
5
10
15
20
25
Epochs

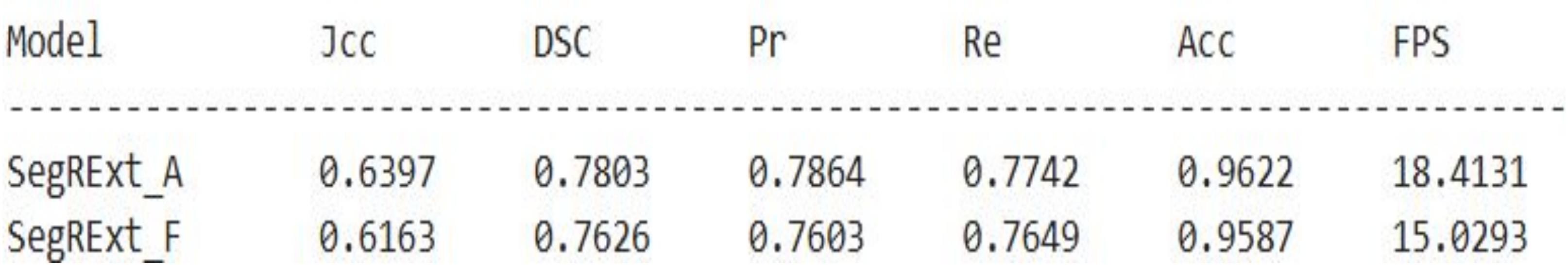

SegRExt_F - Segmentation Metrics
Score
Epochs
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25
DSC
Jcc
Pr
Re
Acc


SegRExt_F - Inference Speed (FPS)
Frames Per Second
Epochs
40.0
37.5
35.0
32.5
30.0
27.5
25.0
22.5
20.0
0
5
10
15
20
25

Testing set metrics after Post Preprocessing

Model	JCC	DSC	Pr	Re	Acc	FPS
SegRExt A	0.648	0.79	0.787	0.798	0.973	18.5
SegRExt F	0.632	0.773	0.761	0.77	0.9608	15.014

Testing Set metrics without Post Processing

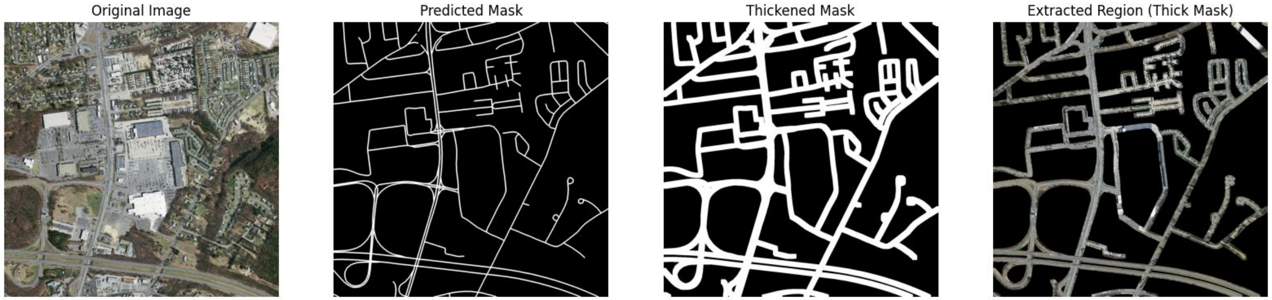


```
Model        Jcc      DSC      Pr       Re       Acc      FPS
------------------------------------------------------------------
SegRExt_A    0.6397   0.7803   0.7864   0.7742   0.9622   18.4131
SegRExt_F    0.6163   0.7626   0.7603   0.7649   0.9587   15.0293
```

Work Done

Vehicle Detection

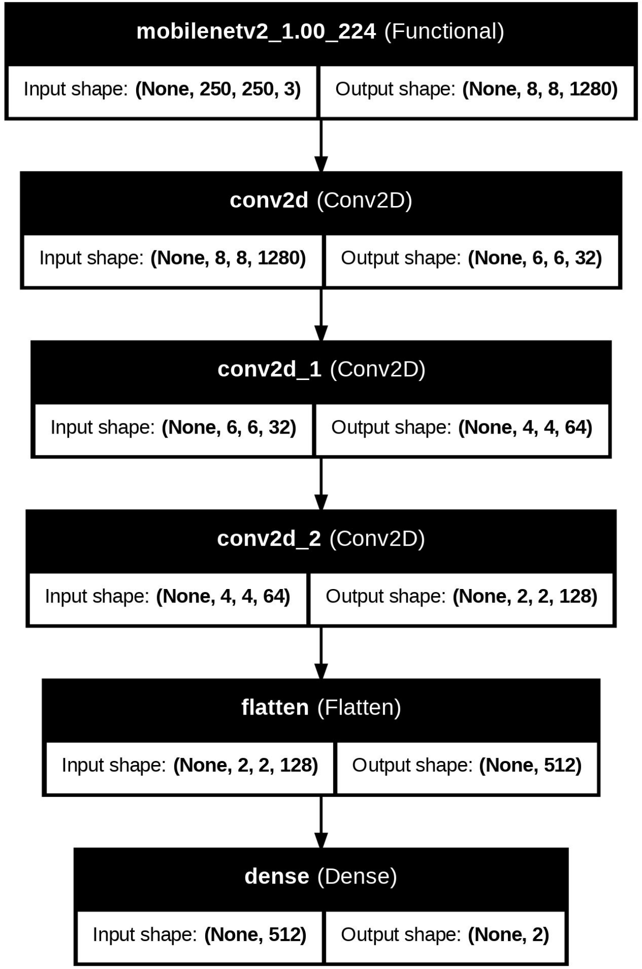
1. Thickened the mask for extracting roads from the image using **Morphological Dilation**. This involves convoluting the 5X5 matrix of ones over the entire image.
2. Approach 1: Used blob detection after applying Gaussian Blur to identify the vehicles.
3. Approach 2: Trained Yolov11 on SIMD dataset for vehicle detection using aerial images.



Work Done

Accident Detection

1. Used MobileNetV2 on a CCTV accident dataset for transfer learning to classify accident vs. non-accident frames.
2. Model Architecture
    a. Base: MobileNetV2 (ImageNet weights, top removed, frozen)
    b. Added three convolutional blocks (32, 64, 128 filters) for feature refinement
    c. Flattened output → Dense softmax layer (2 classes: Accident / Non-Accident)



mobilenetv2_1.00_224 (Functional)
Input shape: **(None, 250, 250, 3)** | Output shape: **(None, 8, 8, 1280)**

conv2d (Conv2D)
Input shape: **(None, 8, 8, 1280)** | Output shape: **(None, 6, 6, 32)**

conv2d_1 (Conv2D)
Input shape: **(None, 6, 6, 32)** | Output shape: **(None, 4, 4, 64)**

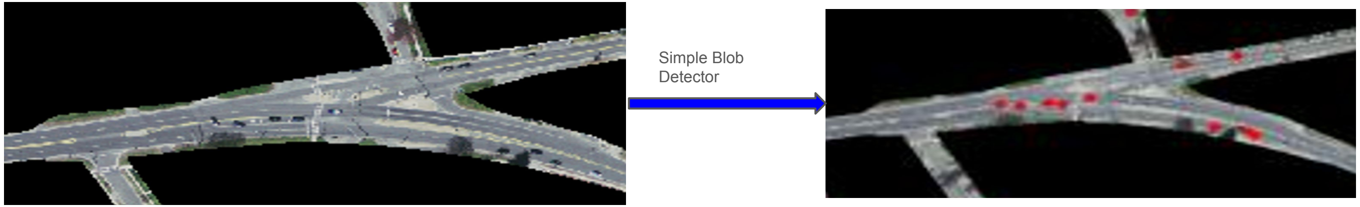
conv2d_2 (Conv2D)
Input shape: **(None, 4, 4, 64)** | Output shape: **(None, 2, 2, 128)**

flatten (Flatten)
Input shape: **(None, 2, 2, 128)** | Output shape: **(None, 512)**

dense (Dense)
Input shape: **(None, 512)** | Output shape: **(None, 2)**

Results

Vehicle Detection



Approach 1: Blob Detection

- Advantage:
  - More suitable in this situation as the vehicles are almost point objects.
- Disadvantage:
  - Gives a lot of false positives and false negatives. Mitigated by adjusting the maximum and minimum blob size allowed to suitable value.

Results

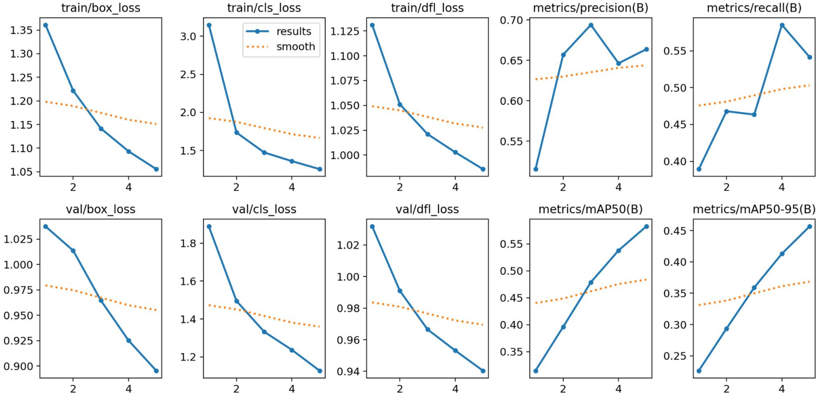
Vehicle Detection

Model	Architechture	mAP@50-95 (%)
Yolov11 (Our Work)	Custom YOLOv11	45.66
SCM-YOLO (Qiang *et al.* 2024)	YOLOv5s (lightweight SCM modules)	27.28

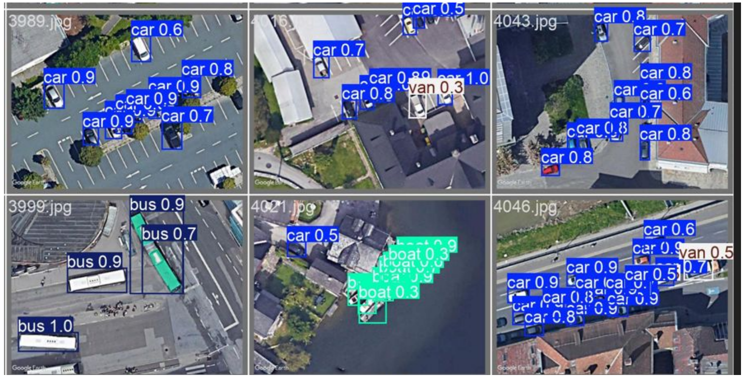
Approach 2: Finetuning Yolov11 over satellite images.

- The model is finetuned on SIMD (Satellite Imagery Multi-vehicle Detection) dataset to make it suitable for detecting vehicles in satellite images.
- Advantages:
  - Yolov11 has been over 1 million images for object detection. Hence, it is often considered suitable for vehicle detection tasks.
- Disadvantages:
  - Massachusettes dataset involves vehicles tiny enough to be called as point objects. Predicting bounding box in this situation is very difficult.
  - It's more suitable to consider the vehicle detection problem on this dataset as landmark detection one rather than object detection. However, due to the lack of availability of dataset in this arena, it is not feasible to train U-Net based models for this task here.

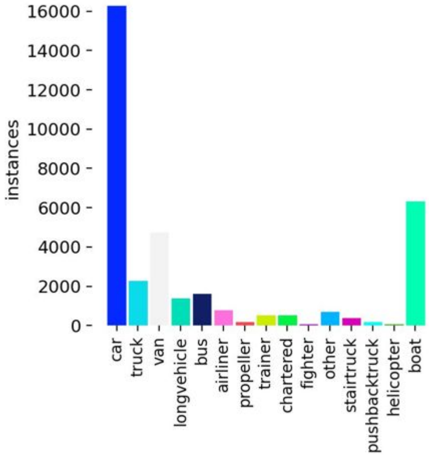
Results



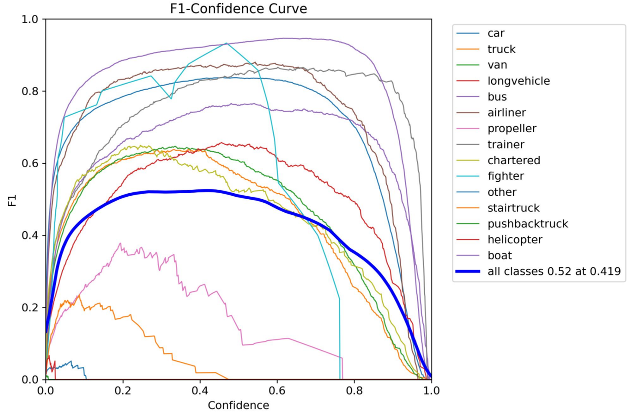
train/box_loss
1.35
1.30
1.25
1.20
1.15
1.10
1.05
train/cls_loss
3.0
2.5
2.0
1.5
results
smooth
train/dfl_loss
1.125
1.100
1.075
1.050
1.025
1.000
metrics/precision(B)
0.70
0.65
0.60
0.55
metrics/recall(B)
0.55
0.50
0.45
0.40
val/box_loss
1.025
1.000
0.975
0.950
0.925
0.900
val/cls_loss
1.8
1.6
1.4
1.2
val/dfl_loss
1.02
1.00
0.98
0.96
0.94
metrics/mAP50(B)
0.55
0.50
0.45
0.40
0.35
metrics/mAP50-95(B)
0.45
0.40
0.35
0.30
0.25
2
4



3989.jpg
car 0.6
car 0.9
car 0.8
car 0.9
car 0.7
4043.jpg
car 0.8
car 0.7
car 0.6
van 0.3
car 1.0
3999.jpg
bus 0.9
bus 0.7
bus 1.0
4021.jpg
car 0.5
boat 0.3
4046.jpg
van 0.5



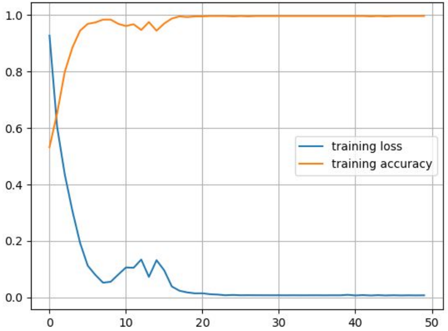
instances
16000
14000
12000
10000
8000
6000
4000
2000
0
car
truck
van
longvehicle
bus
airliner
propeller
trainer
chartered
fighter
other
stairtruck
pushbacktruck
helicopter
boat

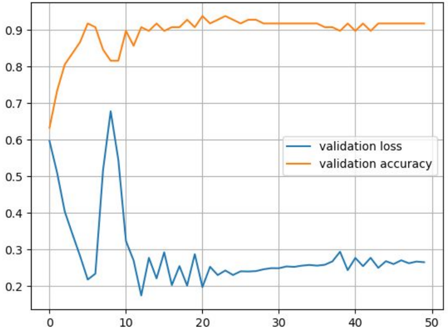


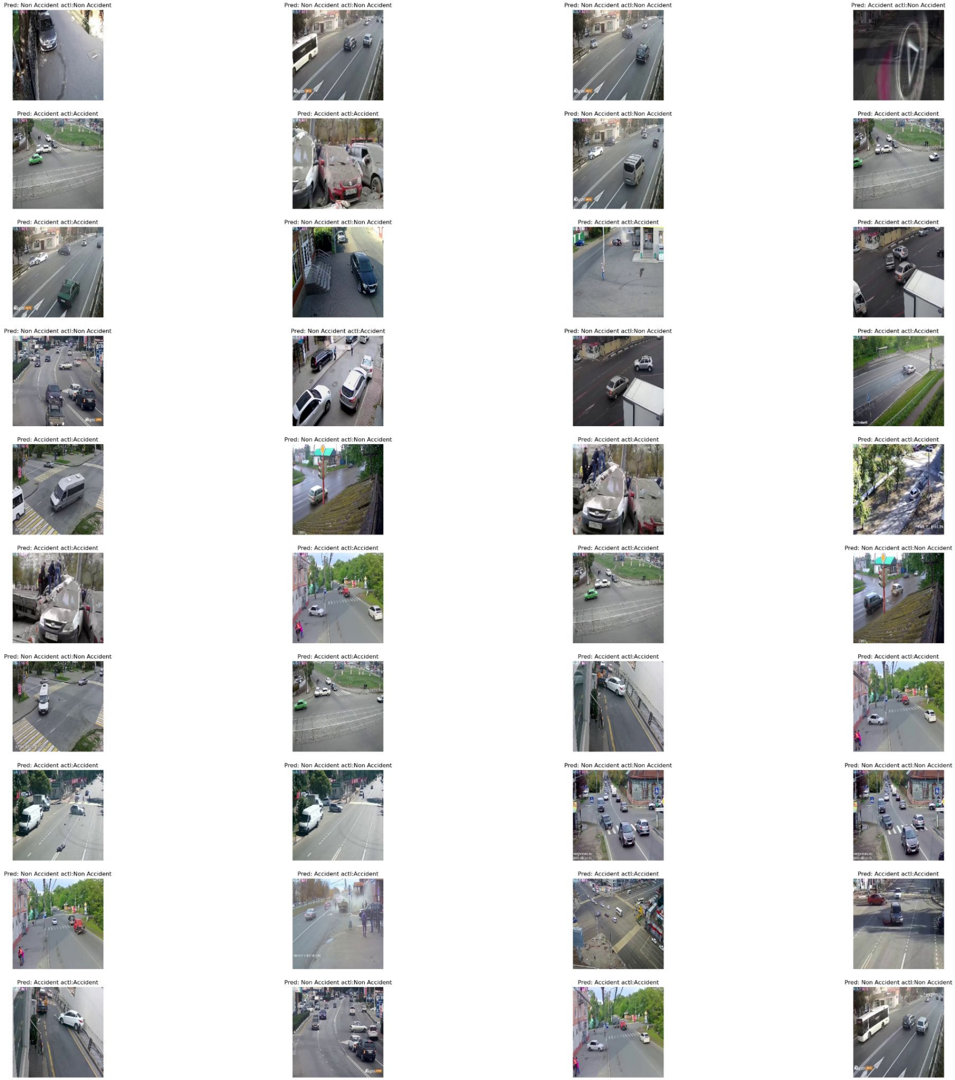
F1-Confidence Curve
F1
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
car
truck
van
longvehicle
bus
airliner
propeller
trainer
chartered
fighter
other
stairtruck
pushbacktruck
helicopter
boat
all classes 0.52 at 0.419

Results

Accident Detection (92% Validation Acc.)



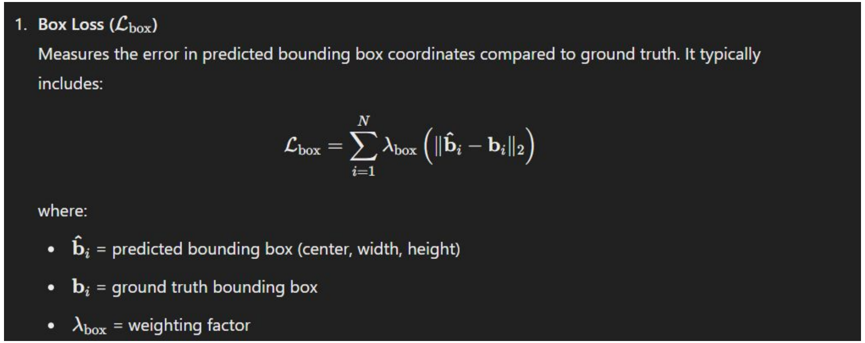



Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident

Comparison with SOTA models (Accident Detection)

Paper Title (Year)	Model(s) Used	Validation/Test Accuracy	Comments
Efficient Deep Learning Methods for Detecting Road Accidents… (Sherimon et al., 2024)	Transfer-learning (VGG16, InceptionV3, etc.) + ML ensemble; Custom SpinalNet CNN	97% (SpinalNet)	SpinalNet (modified CNN) outperformed all baselines on the Kaggle CCTV dataset. Uses frames labeled Accident vs Non-Accident
Road Accident Detection using SVM and Learning: A Comparative Study (Mahesh et al., 2024)	CNN; Recurrent-CNN; SVM; Ensemble (averaging)	94% (ensemble); SVM 93%; CNN 92%; R-CNN 82%	Ensemble (SVM+CNN) gave highest accuracy.
Our Model	MobileNetV7	≈92%	Our model is more light weight and gives comparative accuracy.

Mathematical Formulation

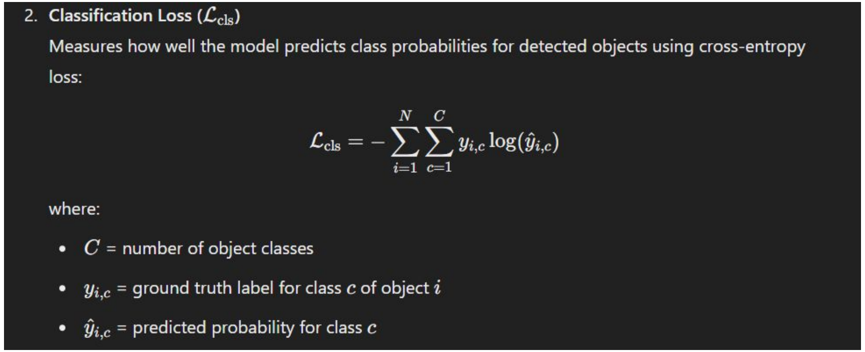


1. **Box Loss ($\mathcal{L}_{\text{box}}$)**
   Measures the error in predicted bounding box coordinates compared to ground truth. It typically includes:

$$\mathcal{L}_{\text{box}} = \sum_{i=1}^{N} \lambda_{\text{box}} \left(\|\hat{\mathbf{b}}_i - \mathbf{b}_i\|_2 \right)$$

where:

- $\hat{\mathbf{b}}_i$ = predicted bounding box (center, width, height)
- $\mathbf{b}_i$ = ground truth bounding box
- λ_{box} = weighting factor

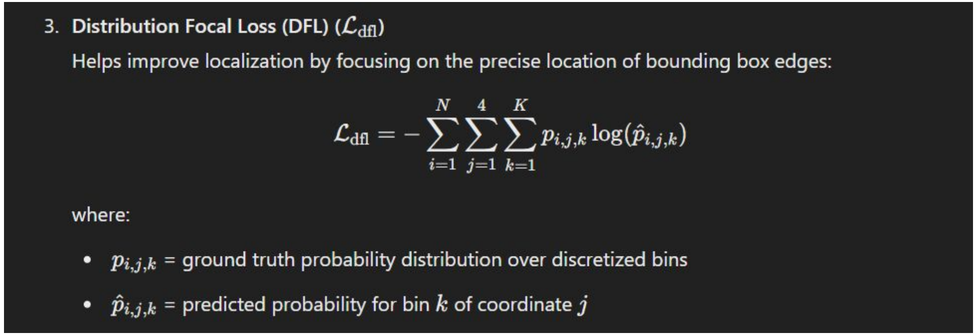


2. **Classification Loss ($\mathcal{L}_{\text{cls}}$)**
   Measures how well the model predicts class probabilities for detected objects using cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c})$$

where:

- C = number of object classes
- $y_{i,c}$ = ground truth label for class c of object i
- $\hat{y}_{i,c}$ = predicted probability for class c

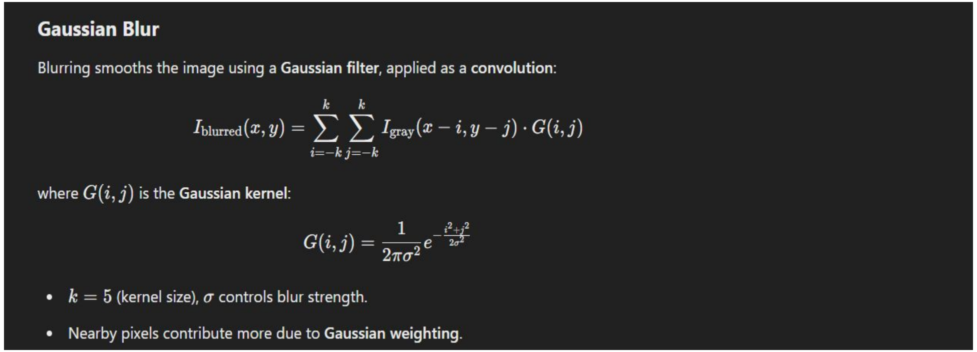


3. **Distribution Focal Loss (DFL) ($\mathcal{L}_{\text{dfl}}$)**
   Helps improve localization by focusing on the precise location of bounding box edges:

$$\mathcal{L}_{\text{dfl}} = -\sum_{i=1}^{N}\sum_{j=1}^{4}\sum_{k=1}^{K} p_{i,j,k} \log(\hat{p}_{i,j,k})$$

where:

- $p_{i,j,k}$ = ground truth probability distribution over discretized bins
- $\hat{p}_{i,j,k}$ = predicted probability for bin k of coordinate j



Gaussian Blur

Blurring smooths the image using a **Gaussian filter**, applied as a **convolution**:

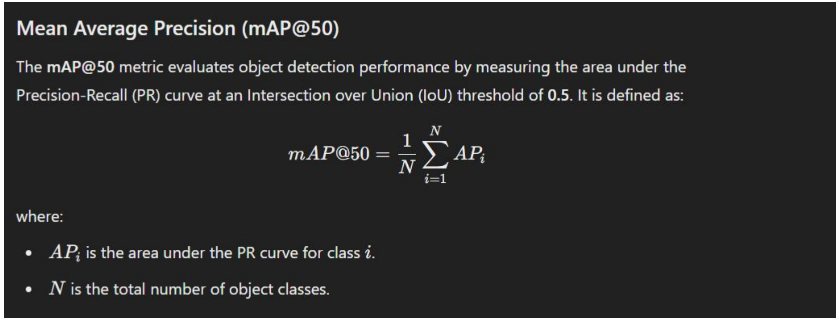
$$I_{\text{blurred}}(x, y) = \sum_{i=-k}^{k}\sum_{j=-k}^{k} I_{\text{gray}}(x-i, y-j) \cdot G(i, j)$$

where $G(i, j)$ is the **Gaussian kernel**:

$$G(i, j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}$$

- $k = 5$ (kernel size), σ controls blur strength.
- Nearby pixels contribute more due to **Gaussian weighting**.

Mathematical Formulation



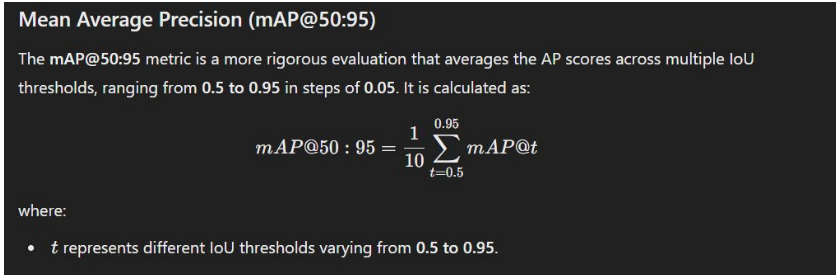
Mean Average Precision (mAP@50)

The **mAP@50** metric evaluates object detection performance by measuring the area under the Precision-Recall (PR) curve at an Intersection over Union (IoU) threshold of **0.5**. It is defined as:

$$mAP@50 = \frac{1}{N}\sum_{i=1}^{N} AP_i$$

where:

- AP_i is the area under the PR curve for class i.
- N is the total number of object classes.



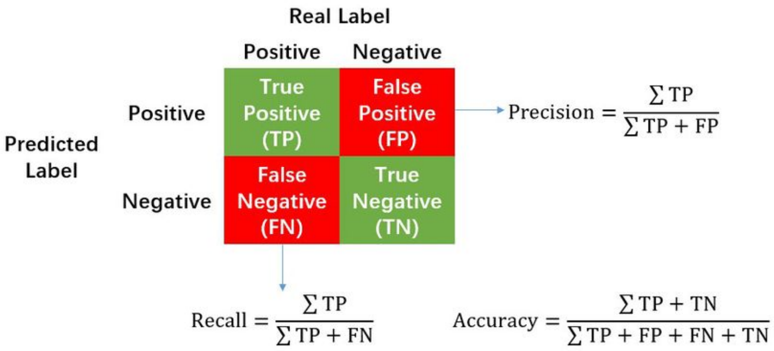
Mean Average Precision (mAP@50:95)

The **mAP@50:95** metric is a more rigorous evaluation that averages the AP scores across multiple IoU thresholds, ranging from **0.5 to 0.95** in steps of **0.05**. It is calculated as:

$$mAP@50:95 = \frac{1}{10}\sum_{t=0.5}^{0.95} mAP@t$$

where:

- t represents different IoU thresholds varying from **0.5 to 0.95**.



		Real Label	
		Positive	Negative
Predicted Label	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

$$\text{Precision} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FN}}$$

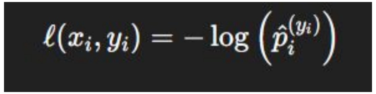
$$\text{Accuracy} = \frac{\sum \text{TP} + \text{TN}}{\sum \text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

Mathematical Formulation

Accident Detection

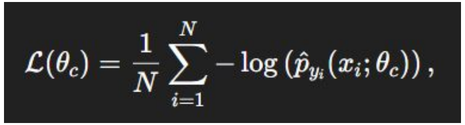
Since labels are integer-coded (0 or 1), we use **Sparse Categorical Cross-Entropy** as loss function:

For one image:

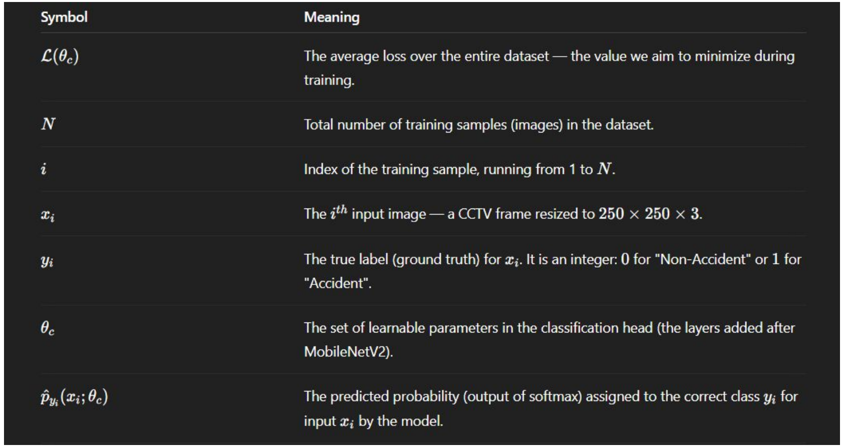


$$\ell(x_i, y_i) = -\log\left(\hat{p}_i^{(y_i)}\right)$$

For the full batch of N samples:



$$\mathcal{L}(\theta_c) = \frac{1}{N}\sum_{i=1}^{N} -\log\left(\hat{p}_{y_i}(x_i;\theta_c)\right),$$



Symbol	Meaning
$\mathcal{L}(\theta_c)$	The average loss over the entire dataset — the value we aim to minimize during training.
N	Total number of training samples (images) in the dataset.
i	Index of the training sample, running from 1 to N.
x_i	The i^{th} input image — a CCTV frame resized to $250 \times 250 \times 3$.
y_i	The true label (ground truth) for x_i. It is an integer: 0 for "Non-Accident" or 1 for "Accident".
θ_c	The set of learnable parameters in the classification head (the layers added after MobileNetV2).
$\hat{p}_{y_i}(x_i;\theta_c)$	The predicted probability (output of softmax) assigned to the correct class y_i for input x_i by the model.

Novelty

1. Independently developed three specialized deep learning pipelines — road extraction, vehicle detection, and accident detection — tailored to different imaging sources (satellite and CCTV).
2. Trained YOLOv11 on the SIMD dataset for improved vehicle detection in satellite imagery, going beyond pre-trained models to adapt to domain-specific challenges.
3. Utilized MobileNetV2 and efficient segmentation networks to ensure deployability even in resource-constrained environments.

Conclusion and future scope

In this project, deep learning pipelines were developed to address key aspects of traffic scene understanding — road extraction, vehicle detection, and accident recognition — across diverse visual environments.

The models demonstrated strong performance on satellite and CCTV data, highlighting the potential of deep learning to enhance traffic monitoring and safety management systems.

Future improvements could involve integrating these pipelines for real-time, end-to-end intelligent traffic analysis.

References

Bhatnagar, S., Gill, L., & Ghosh, B. (2020). Drone image segmentation using machine and deep learning for mapping raised bog vegetation communities. *Remote Sensing, 12*, 2602.

Bisio, I., Garibotto, C., Haleem, H., Lavagetto, F., & Sciarrone, A. (2022). A systematic review of drone-based road traffic monitoring system. *IEEE Access, 10*, 101537–101555.

Bisio, I., Garibotto, C., Haleem, H., Lavagetto, F., & Sciarrone, A. (2023). Traffic analysis through deep-learning-based image segmentation from UAV streaming. *IEEE Internet of Things Journal, 10*(7), 6059–6073.

Byun, S., Shin, I.-K., Moon, J., Kang, J., & Choi, S.-I. (2021). Road traffic monitoring from UAV images using deep learning networks. *Remote Sensing, 13*, 4027.

Dikbayir, H. S., & Bülbül, H. İ. (2020). Deep learning-based vehicle detection from aerial images. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 956–960).

Gupta, M., & Mishra, A. (2024). A systematic review of deep learning-based image segmentation to detect polyp. *Artificial Intelligence Review, 57*(1), 7.

Liu, X., Yang, T., & Li, J. (2018). Real-time ground vehicle detection in aerial infrared imagery based on convolutional neural network. *Electronics, 7*(6).

Masubuchi, S., Watanabe, E., Seo, Y., Okazaki, S., Sasagawa, T., Watanabe, K., Taniguchi, T., & Machida, T. (2020). Deep-learning-based image segmentation integrated with optical microscopy for automatically searching for two-dimensional materials. *npj 2D Materials and Applications, 4*(1).

References

Qiang, H., Hao, W., Xie, M., Tang, Q., Shi, H., Zhao, Y., & Han, X. (2025). SCM-YOLO for lightweight small object detection in remote sensing images. *Remote Sensing, 17*(2).

Singh, A., Rajesh, B., & Javed, M. (2021). Deep learning-based image segmentation directly in the JPEG compressed domain. In *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)* (pp. 1–6).

Tiwari, R., Rumaney, A. H., & Saravanan, M. (2023). Real-time traffic surveillance and detection using deep learning and computer vision techniques. In *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)* (pp. 1–6).

Zhang, H., Liptrott, M., Bessis, N., & Cheng, J. (2019). Real-time traffic analysis using deep learning techniques and UAV-based video. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1–5).