Applied Mathematics and Computational Intelligence
Volume 14, No. 4, 2025 [14 - 29]

Development of Road Asset Mapping using Dashcam

Nurul Syahirah Abdul Karim[1], Nor Azuana Ramli[2*], Mohd Radhie Mohd Salleh[3]

[1,2]Centre for Mathematical Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah, Lebuh Persiaran Tun Khalil Yaakob, 26300 Kuantan, Pahang, Malaysia.
[3]Department of Water and Environmental Engineering, Faculty of Civil Engineering, Universiti Teknologi Malaysia.

[*] Corresponding author: azuana@umpsa.edu.my

Received: 6 November 2024
Revised: 21 March 2025
Accepted: 24 March 2025

ABSTRACT

Road asset mapping significantly benefits transportation authorities, infrastructure management, and road users. Recent advancements in Geographic Information Systems (GIS) and digital mapping technologies have substantially improved inventory and asset management. However, technologies such as Light Detection and Ranging (LiDAR) and mobile mapping cameras are costly compared to dashcams. Therefore, this study proposes a cost-effective road asset mapping system by leveraging dashcam video data, object detection using You Only Look Once version 8 (YOLOv8), and GIS for spatial visualization. In this study, images were extracted from dashcam videos using VLC Media Player and processed through an annotation pipeline in Roboflow, where bounding boxes and labels were assigned to road assets such as road signs, streetlights, and traffic lights. The dataset was then divided into training, validation, and test sets for model development. YOLOv8, selected for its high accuracy in object detection and segmentation, was trained to recognize these assets, achieving a precision of 0.895, recall of 0.873, and mean Average Precision (mAP) of 0.876 at 50% Intersection over Union (IoU). To integrate YOLOv8 with GIS, the detected road assets were geotagged based on GPS metadata from the dashcam footage, allowing spatial mapping within a GIS platform. The identified assets were then visualized on a GIS interface, facilitating efficient road asset inventory management. This approach demonstrates that low-cost dashcam-based data collection, combined with AI-powered object detection and GIS mapping, offers a viable alternative to expensive mapping technologies. Future research should focus on enhancing dataset quality and expanding the range of detectable assets to further improve system accuracy and applicability.

Keywords: deep learning, dashcam, geographic information systems, object recognition, road asset mapping.

1 INTRODUCTION

Effective road asset mapping is crucial for transportation authorities, infrastructure management, and general users, as it facilitates informed decision-making regarding road network expansion, safety improvements, and maintenance planning. Traditionally, road asset inventory relies on remote sensing technologies such as high-accuracy laser point clouds, terrestrial photographs, aerial imagery, and Geographic Information Systems (GIS), significantly enhancing asset management by

providing precise spatial information. However, the cost and complexity of existing mobile mapping systems (MMS), including Light Detection and Ranging (LiDAR) and multi-sensor integrations, pose significant challenges to widespread adoption. Many total station surveys are conducted in remote, underdeveloped areas where no local control network is expected to be established and connected to a projected coordinate system for real-world applications. While mapping has long been a well-established engineering discipline with significant influence on urban planning and infrastructure development, there is an increasing need for timely and accurate updates of geospatial data. The shift toward rapid geospatial data capture has been driven by the advancement of multi-platform and multi-sensor integrated mapping technologies, transforming mapping into a dynamic and mobile process [2].

In the 1970s, highway transportation departments utilized photo-logging systems to monitor pavement performance, signage, maintenance effectiveness, and encroachments. However, these systems lacked three-dimensional (3-D) object-measuring capabilities due to the low accuracy of vehicle location and the use of only a single camera configuration. With the development of GPS and video imaging technology, inefficient photo-logging methods were replaced with video-logging systems based on the Global Positioning System (GPS). Mobile mapping systems now offer complete 3-D mapping capabilities through integrated multi-sensor data collection and processing technologies, distinguishing them from video-logging systems [2]. Despite these advancements, the cost and operational complexity of MMS remain significant barriers. The development, hardware cost, and accuracy requirements of MMS are interconnected, and their deployment in urban areas is expensive due to the need for intricate mathematical methodologies and computationally intensive processing steps [3]. The vast amount of data generated by MMS, including LiDAR point clouds, high-resolution images, and GPS coordinates, requires significant storage and computational resources, further adding to operational costs. Moreover, mobile mapping systems typically rely on multiple sensors to collect diverse data types simultaneously, necessitating complex calibration, sensor fusion, and geometric alignment tasks [4].

To address these limitations, this study proposes a cost-effective alternative for road asset mapping by utilizing a dashcam-based system integrated with You Only Look Once version 8 (YOLOv8) for object detection and GIS for spatial mapping. Dashcams offer several advantages, including affordability, ease of deployment, and the ability to capture high-resolution video footage under real-world driving conditions. Unlike expensive LiDAR-based systems, dashcams can be easily installed on various vehicles, enabling continuous and standardized data collection with minimal disruption to traffic. YOLOv8 was chosen for its ability to perform real-time object detection with high accuracy, allowing for automated identification of road assets such as road signs, streetlights, and traffic lights. The detected assets are then geotagged using GPS metadata from the dashcam footage and mapped in a GIS platform, providing a structured and spatially referenced road asset inventory. By integrating YOLOv8 and GIS, this research offers an innovative, low-cost solution for road asset mapping that reduces reliance on expensive mobile mapping systems while maintaining accuracy. This approach demonstrates the feasibility of using dashcam-based road asset mapping and highlights the potential for further improvements through enhanced dataset quality and model optimization.

2 PREVIOUS STUDIES IN DETECTING ROAD ASSET

Several studies have been conducted investigating the application of machine learning and deep learning techniques for detecting road assets. A support vector machine (SVM), artificial neural

network (ANN), Naïve Bayes (NB), and Random Forest (RF) were used to improve the performance of the colour segmentation task in traffic light detection [5]. Once the classifier model is established, the detection phase can begin. This phase was conducted on a single frame, regardless of whether the video was captured in real time. After preprocessing, the method analysed the image by iterating through each pixel in a single frame. Each instance detected is explicitly categorized. If a pixel is determined to represent a traffic light, the method either advances to the next stage of traffic light detection or marks it in the image frame for display on the output device.

Nandargi et al. [6] employed a practical approach, using semantic segmentation, a deep learning technique, to identify the components in an image and perceive the image's elements [6]. The Light Detection and Ranging (LiDAR) input data was processed using deep learning techniques like semantic segmentation, a deep learning algorithm that associates each pixel in an image with a label or category. The main goal is to classify every pixel in the image as either a road or a non-road, making it a practical and efficient method. Other than that, Network-in-Network (NiN) is an implementation of the standard convolutional neural network (CNN) architectural design, consisting of a series of convolutional and pooling layers, followed by several fully connected layers, the final layer performing the loss function. Stochastic gradient descent was used to prepare, and the chain rule registers the gradients precisely as in a conventional multilayer perceptron (MLP).

Another study utilized deep learning architectures, and the model was trained on a vast picture dataset consisting of over 56,000 photos of traffic signs [7]. 56,111 images were used from six distinct traffic sign classes: stopped, yield, no entrance, obligation, prohibition, and danger. The YOLOv3 architecture was trained using a hierarchical classification technique. The ResNet152 architecture was trained using 27,308 images in the instance of obligation and prohibition, in which a greater quantity of images was accessible. It recognized 11 subclasses of pro-prohibition signals, and 14 obligation sign classifications were made.

A novel approach was introduced to segmenting urban assets using automated data processing workflows and a less expensive Azure Kinect [8]. The method was first validated by detecting road signs outside using the Time of Flight (ToF) camera from Azure Kinect. The Region of Interest (ROI) was swiftly and effectively retrieved using the data produced by the ToF camera. After converting the ROI to an RGB image, a hybrid colour-shape-based technique was used to retrieve the traffic sign area. Furthermore, using the depth image as a guide, they measured the distance between Azure Kinect and the traffic sign, showcasing the innovative use of technology in urban asset segmentation.

Transfer learning was employed to recognize traffic signs on dashcam images [9]. The models utilized in this task were all obtained using Pytorch's torch-vision package and pre-trained on COCO. These models were meticulously trained using PyCocoTools' composite loss function, with the primary classes being background and sign. For testing, ResNet50-FPN, MobileNetV3-Large FPN, and MobileNetV3-Large FPN for mobile platforms were the three Faster R-CNN network backbones used. The TDOT dashcam image collection was utilized to train and validate these models. Each model completed five training epochs with a batch size of four and a learning rate of 0.0001, and the validation performance of each model was compared to ensure the most suitable model for detection was selected, providing a thorough and reliable training process. A summary of previous studies has been listed in Table 1.

Table 1 : Summary of previous studies' methods and findings

Author(s)	Methods	Findings
Binangkit & Widyantoro [5]	SVM, ANN, NB, and RF with color models	Based on all color segmentation algorithms combined, the traffic light precision number is 0.43, NB is 0.71, ANN is 0.73, SVM is 0.84, and RF is 0.64. SVM with an RBF kernel is the best learning method compared to MLP, RF, and NB. Compared to the color segmentation approach, the best SVM performance lowers recall by just 5% while improving precision by 93%.
Sairam et al. [1]	Laser scanner, cameras, Inertial Measurement Unit, and GPS	A cost-effective mobile mapping system was developed to solve the limitations of the previous methods. Despite lacking a single, well-investigated procedure for automatically extracting characteristics from point clouds and images, experiments have demonstrated several effective methods for recognizing assets.
Campbell et al. [10]	Deep learning	The results show that the F-score is 89.75%, the accuracy is 95.63%, the precision is 82.97%, the sensitivity is 82.5%, and the specificity is 96.98%. Although accuracy shows success despite system adversity, this model's sensitivity and precision findings cannot be compared to other systems. A manual check across all images was conducted throughout this investigation to yield a 95.63% detection accuracy and a 97.77% classification accuracy.
Elhashash et al. [11]	MMS, GNSS, LiDAR, IMU, DMI, sensor, Imaging Systems, and cameras	Implementing mobile mapping technology in various applications may save operating costs while increasing productivity.
Domínguez et al. [7]	Camera LiDAR and deep learning	The accuracy, recall, F-Score, and mean IoU values are 0.611, 0.868, 0.717, and 0.685, respectively, indicating a notably elevated frequency of false positives. The main reason is that the traffic sign identification algorithm could recognize traffic signals that the manual labeler had overlooked, as they only covered a small enough area within the bounding box to be significant. The initial stage's traffic sign recognition produced the correct findings. 96 of the 99 genuine positive detections were correctly identified. There are 47 prohibition signs; 34 may be identified, and 28 can be correctly predicted.
Qiu et al. [8]	MMS and deep learning	The study's method has an accuracy of 0.8216, whereas deep learning has an accuracy of 0.7466, suggesting that their approach is more flexible and cost-effective.

In summary, previous studies on road asset mapping have primarily relied on expensive and complex methods such as mobile mapping systems (MMS), LiDAR, and deep learning models, which require high computational power, extensive data processing, and specialized hardware. These approaches

face cost, scalability, and flexibility limitations, making them impractical for large-scale implementation. Additionally, some methods report elevated false positive rates and challenges in real-time processing. In contrast, the proposed dashcam-based approach with YOLOv8 offers a cost-effective, easily deployable, and scalable solution. Dashcams provide high-resolution, real-world video footage, while YOLOv8 ensures accurate, real-time object detection with minimal computational requirements. Integrating GIS enables efficient spatial mapping of road assets without the need for expensive LiDAR setups, making road asset management more accessible and practical.

3 MATERIAL AND METHODS

A review of past studies has offered a thorough evaluation of different methodologies for road asset mapping. This research highlighted the systematic approaches for creating effective road asset mapping using low-cost dashcams. Figure 1 presents the flowchart outlining this research process.

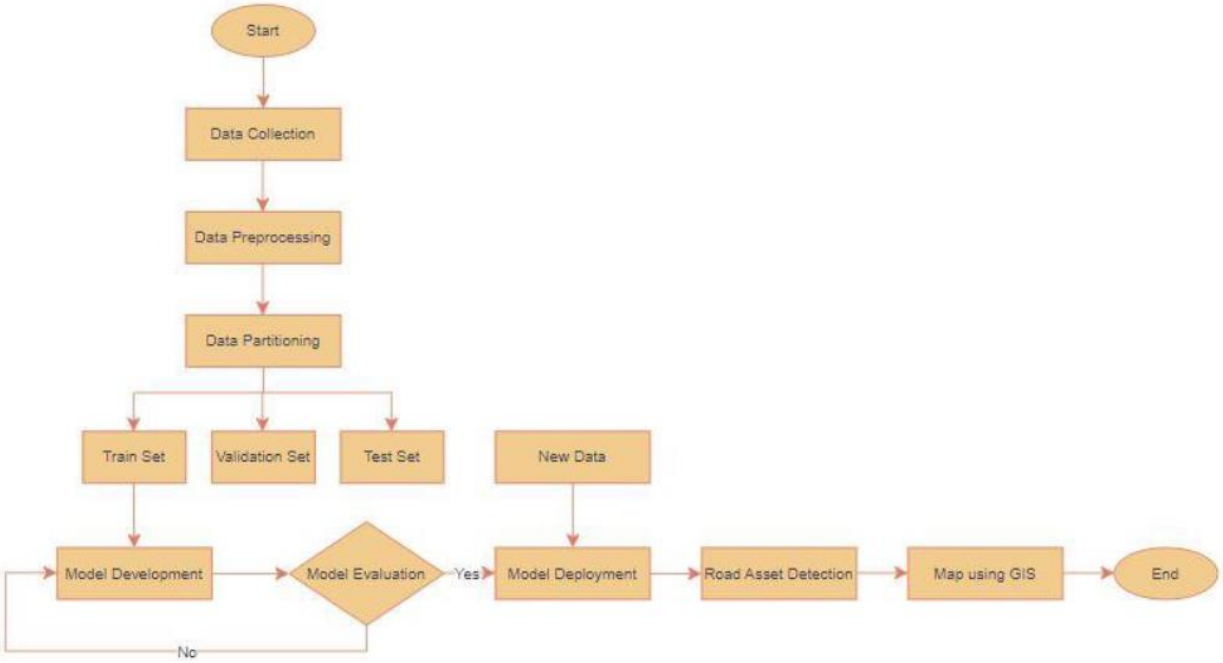

Figure 1 : Research flowchart

3.1 Data Collection

The data used for this research is primary and collected directly from the sources. The dataset was collected using a Garmin-branded dashcam around Skudai, Johor. The videos are approximately 10 minutes long and feature road assets that are valuable for road asset mapping and analysis.

3.2 Image Preprocessing

Maharana et al. [12] stated that the first stage of machine learning is called data preprocessing, where the data is transformed or encoded to put it in a format that allows the machine to analyze or parse

it quickly [12]. Data preprocessing is the most crucial stage in a supervised machine learning algorithm's performance during generalization. The amount of training data increases exponentially with the dimension of the input space. According to estimates, preprocessing is vital to model-building since it can cost as much as 50–80% of the total classification time. Additionally, data quality is necessary to improve the model's performance. Data processing is the set of procedures to be completed before beginning the analysis of the model's actual data. This study used bounding boxes and labeling for data preprocessing.

3.3 Data Partitioning

One of the most critical steps in video indexing is partitioning a video source into meaningful segments. The system uses a set of different metrics to measure how the data changes between movie frames. It is a crucial approach in many fields, such as content analysis, video processing, and computer vision. Data partitioning is important for identifying and detecting road assets in this study. It is simpler to distinguish and follow distinct parts within a video by segmenting the film into separate frames. The dataset was split into 70:20:10, where 70% of the dataset is used for training the model, 20% of the dataset is used for validation of the model, and the remaining 10% is used for testing the model's performance on unseen data.

3.4 Model Development

After applying the data partitioning procedure, the collected data becomes more meaningful, allowing us to proceed with the modeling phase, which utilizes You Only Look Once version 8 (YOLOv8). Figure 2 illustrates the architecture of the YOLOv8 model.

YOLOv8 generically separates its architecture into three main layers: the Backbone, Neck, and Head. The Backbone collects all valuable features from the input image [14]. It is typically a CNN pre-trained on large-scale image classification tasks like ImageNet. The Backbone collects features hierarchically at various scales. In the first layer, the lower-level features are extracted, and in the following levels, high-level features are collected, including object parts and semantic information. According to [15], this makes YOLOv8 able to receive more detailed gradient flow information while remaining lightweight. At the end of the Backbone, the most usual SPPF module is still applied and passed through three serial MaxPools of size 5×5. Afterward, every layer is concatenated to ensure the target precision is maintained at different sizes yet lightweight [14].

The Neck module acts as a bridge between the Backbone and the Head. It fuses and refines the extracted features from the Backbone, typically oriented to achieve enhancements in spatial and semantic information at various scales. Additional convolutional layers, feature pyramid work, or other methods could be included in the Neck to help refine feature representation. In the neck part, YOLOv8 continues to utilize a feature fusion method called PAN-FPN. This method enhances the original information of the feature layer, allowing for further fusion and utilization across multiple diverse scales [14]. The authors of YOLOv8 designed the Neck module, and the main contents contain two up-sampling and several C2f modules with the final decoupled Head structure. YOLOv8 adopted the concept of separating the Head, similar to YOLOx, in the last part of the Neck. This further development, focused on confidence and regression boxes, achieved a new level of accuracy in detection.

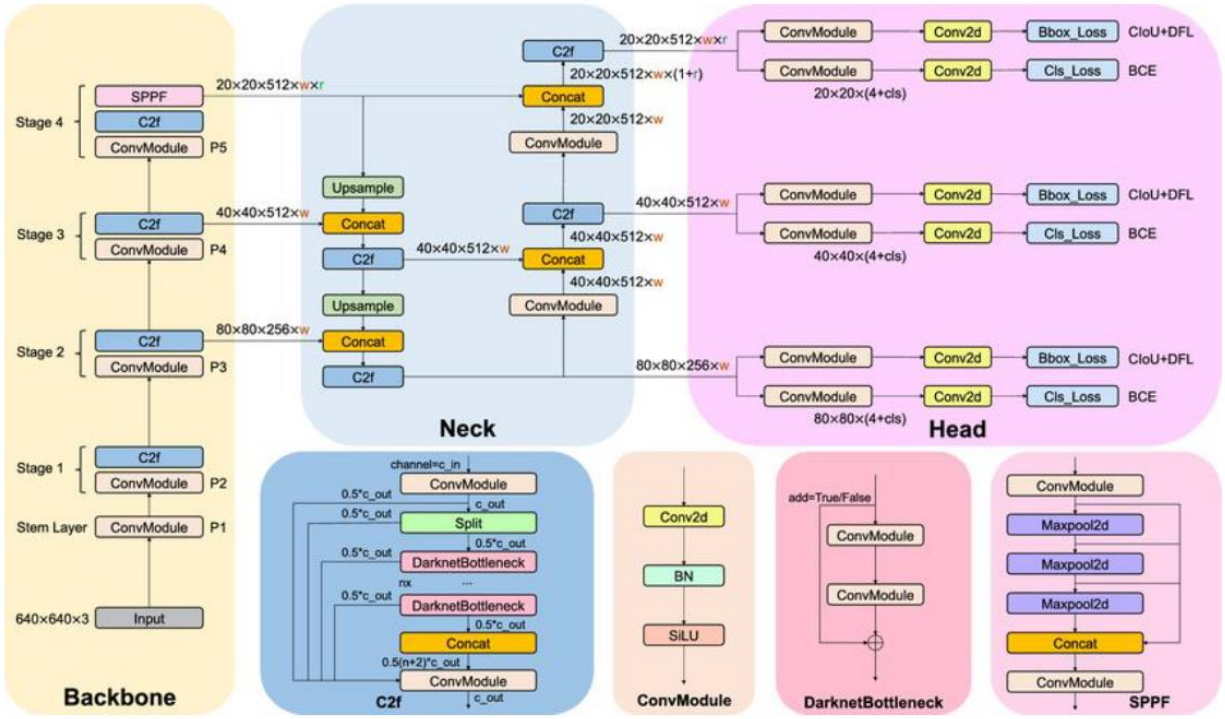

Figure 2: YOLOv8 model architecture [13]

The Head is the last component of any object detector, making predictions based on the features obtained from the Backbone and Neck. It is typically built from one or more task-specific sub-networks performing the functions related to classification, localization, and, progressively, instance segmentation and pose estimation. The Head processes the Neck's characteristics and predicts every feasible object. To measure how well YOLOv8 predicts object locations, the Intersection over Union (IoU) metric is used. IoU quantifies the overlap between the predicted bounding box (B_p) and the ground-truth bounding box (B_g). It is defined as:

$$IoU = \frac{|B_p \cap B_g|}{|B_p \cup B_g|} \tag{1}$$

where $|B_p \cap B_g|$ is the area of intersection and $|B_p \cup B_g|$is the total area covered by both boxes. A higher IoU indicates a better bounding box prediction.

After bounding boxes are predicted, Non-Maximum Suppression (NMS) is applied to filter out redundant detections and retain only the most confident predictions. NMS works by selecting the bounding box with the highest confidence score and suppressing all other boxes with an IoU greater than a predefined threshold. The NMS function is given by:

$$S = \{b_i | IoU(b_i, b_j) < threshold, \forall j > i\} \tag{2}$$

where S is the final set of selected bounding boxes, and b_i and b_j are the predicted bounding boxes ranked by confidence.

To improve localization accuracy, YOLOv8 employs Complete IoU (CIoU) Loss, which enhances standard IoU by incorporating distance and aspect ratio penalties. The CIoU Loss is expressed as:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b_p, b_g)}{c^2} + \alpha v \quad (3)$$

where $\rho^2(b_p, b_g)$ is the Euclidean distance between the centers of the predicted and ground-truth boxes, c is the diagonal length of the smallest enclosing box that covers both predicted and ground-truth boxes, v measures the difference in aspect ratios, and α is a trade-off parameter that balances aspect ratio consistency.

After refining bounding box predictions using CIoU Loss, YOLOv8 optimizes its predictions using a combination of classification and regression loss functions:

1. Classification Loss: Binary Cross-Entropy (BCE) Loss is used to evaluate object classification accuracy, ensuring that predicted object categories match ground-truth labels.
2. Regression Loss: YOLOv8 employs CIoU Loss + Distribution Focal Loss (DFL) to improve bounding box localization by focusing on both overlap and fine-grained positional adjustments.
3. Variational Focal Loss (VFL): This loss function incorporates an asymmetric weighting procedure to prioritize difficult-to-classify samples while down-weighting easy predictions.

The generic distribution models the DFL-Box location, allowing the network to emphasize the distribution of the site nearest to the target location. This enhances precision by increasing the probability density at the most relevant positions. The sigmoid output of the network is denoted by s_i, with interval orders y_i and y_i+1, and label y, as formulated in equation (4):

$$\mathrm{DFL}_{(s_i, s_{i+1})} = -((y_{i+1} - y)\log(s_i) + (y - y_i)\log(s_{i+1})) \quad (4)$$

YOLOv8 uses Anchor-Free rather than Anchor-Base. In version 8, a dynamic TaskAlignedAssigner is employed to implement its matching approach. Equation (5) determines the Anchor-level alignment degree for each occurrence, where s is the classification score, u is the IOU value, and α and β are the weight hyperparameters. It picks m anchors with the highest value (t) in each instance as positive samples and the remaining as negative samples before training using the loss function [14].

$$t = s^{\alpha} \times u^{\beta} \quad (5)$$

3.5 Model Evaluation

The evaluation process is essential for determining how well the model performs. One of the significant tools of prediction analysis within machine learning is the confusion matrix. It can be used to estimate a model's efficiency and effects concerning machine learning in classification. Additionally, it is a calculated summary of the number of accurate and inaccurate predictions a classifier produced for tasks involving binary classification. An N x N matrix, where N is the number of target classes, is called a confusion matrix when assessing a classification model's effectiveness. By

visualizing the confusion matrix, one might assess the model's correctness by analysing the diagonal values representing the number of accurate classifications.

From this matrix, metrics like accuracy, precision, recall, specificity, and F1-score can be calculated. Positive observation is known as positive, and negative observation is known as negative. A result when the model accurately predicts the positive class is called a True positive (TP), while the model correctly predicts the negative classes, known as True Negatives (TN). The model incorrectly predicts the positive class when negative, also known as a type 1 error, is a False Positive (FP). Lastly, an outcome where the model incorrectly predicts the negative class when it is positive, also known as a type 2 error, is a False Negative (FN).

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. It can be calculated from the confusion matrix using the equation below:

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

Average Precision (AP) is the average accuracy of the model. It can be calculated from the equation below:

$$AP = \int_0^1 p(r)dr \quad (7)$$

Mean Average Precision (mAP) is the average value of the AP. It can be calculated from the equation below:

$$mAP = \frac{1}{k}\sum_{i=1}^{k} AP_i \quad (8)$$

4 RESULTS AND DISCUSSION

Data was prepared by extracting images from video obtained from the dashcam. Then, data preprocessing, such as bounding box, labelling, and data partitioning, was done using Roboflow. All images were resized to 640 x 640 pixels and auto-oriented before being used for model training. Figure 3 shows the results obtained from the model training using YOLOv8. The precision values for roads (22 instances), streetlights (53 instances), and traffic lights (30 instances) are 1.000, 0.973, and 0.963, respectively, or overall (105 instances) is 0.979. The high precision indicates that the model's detections are mostly accurate. Next, the mean average precision (mAP) value at 50% for roads, streetlights, and traffic lights is 0.988, 0.974, and 0.995, respectively, or overall is 0.974. The high mAP score suggests the model performs well in accurately localizing objects across different classes. Lastly, the mean average precision (mAP) value at 95% for road signs is 0.874; streetlights are 0.876, traffic lights are 0.937, and overall is 0.896. This indicates that the model performs well despite some drops compared to the mAP50 score.

```
Model summary (fused): 168 layers, 11126745 parameters, 0 gradients, 28.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 3/3 [00:02<00:00,  1.44it/s]
                   all         89        105      0.979      0.968      0.988      0.896
             Road-sign         89         22          1      0.979      0.995      0.874
           Streetlight         89         53      0.973      0.925      0.974      0.876
         Traffic-light         89         30      0.963          1      0.994      0.937
Speed: 0.2ms preprocess, 4.6ms inference, 0.0ms loss, 4.3ms postprocess per image
```

Figure 3: Summary of model training by using YOLOv8

Figure 4 shows the training result. It shows that the "metrics/precision(B)" and "metrics/recall(B)" graphs are growing, indicating that the model is becoming more accurate in identifying objects while resulting in fewer false positives and false negatives.

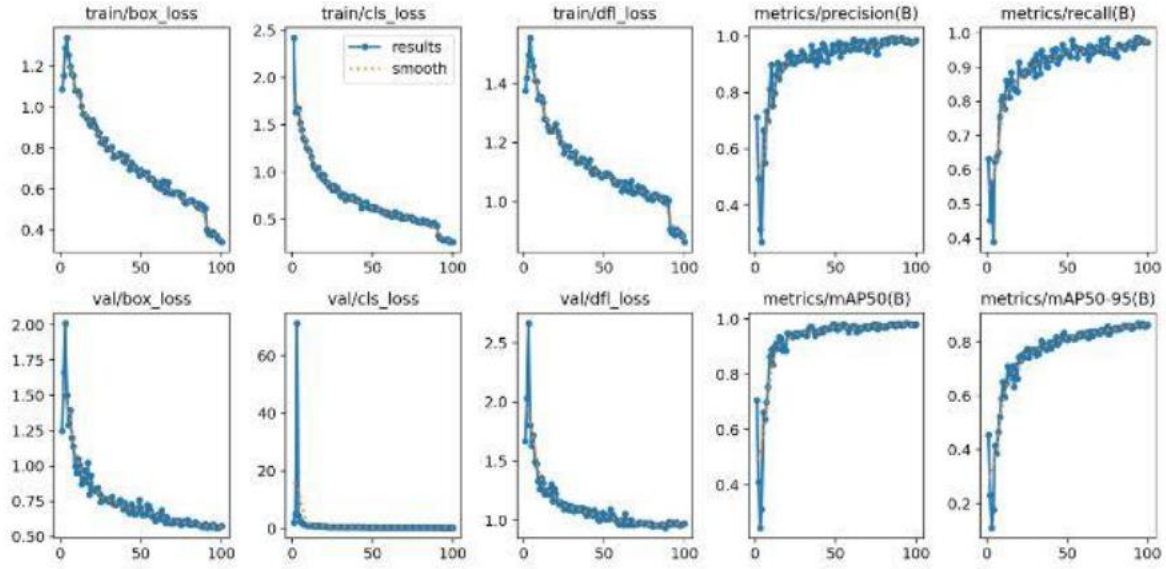

Figure 4: Training results

The F1-Confidence curve of the training model is shown in Figure 5. All individual classes, including road signs, streetlights, and traffic lights, have good F1 scores across most of the confidence thresholds, indicating good performance in class detection. Additionally, the performance across all classes is strong, and the thick blue line shows a close-to-1 total F1 score for most of the confidence range. In short, the F1 score of 0.727 shows that the system balances precision and recall well.

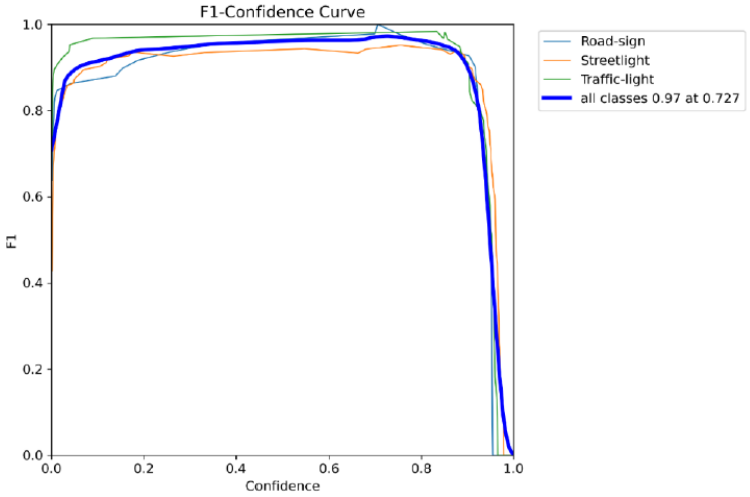

Figure 5: F1-Confidence curve

Figure 6 shows some images of the model's predictions on the first batch of the validation dataset, while Figure 7 shows the model validation result. In summary, 105 total instances give precision of the bounding boxes around detected objects, recall, mAP at 50% Intersection over Union (IoU), and mAP averaged over multiple IoU thresholds from 50% to 95% value of 0.895, 0.873, 0.876, and 0.936, respectively. This indicates the model detected road signs, streetlights, and traffic lights well. Meanwhile, the mAP50 is high across all classes, while the mAP50-95 is lower, indicating that the model's performance changes with the IoU threshold.

Figure 6: Predicted images

```
Model summary (fused): 168 layers, 11126745 parameters, 0 gradients, 28.4 GFLOPs
val: Scanning /content/datasets/road-asset-detection-2/valid/labels.cache... 89 images, 0 backgrounds, 0 corrupt: 100% 89/89 [00:00<?, ?it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:03<00:00,  1.52it/s]
                   all         89        105      0.979      0.968      0.988      0.895
             Road-sign         89         22          1      0.981      0.995      0.873
           Streetlight         89         53      0.973      0.925      0.974      0.876
         Traffic-light         89         30      0.963          1      0.994      0.936
Speed: 3.6ms preprocess, 12.4ms inference, 0.0ms loss, 10.0ms postprocess per image
```

Figure 7: Validation results

Based on the results obtained, the model has been deployed for road asset detection. Figure 8 illustrates the detection of a road sign with a confidence score of 0.93. Such a high confidence score suggests that the detection is likely correct. Therefore, identifying the road sign with a 93% confidence score is very probable. This implies that the object identified by the model matches the actual object in the image, aligning with the ground truth.

Figure 8: Road sign detection

Besides road signs, other assets such as traffic lights and streetlights have also been tested on the model. Table 2 below summarizes the road asset detection results based on its cases.

Table 2: Summary of road asset detection results

No	Detection	Object	Confidence Score	Description
1	Successful detection	Road sign	0.93	The model's detected object is present in the image and has been correctly identified with a high confidence score.
2		Streetlight	0.73	
3	False detection	Road sign	0.33	The road sign false detection on road markings had a 0.33 confidence score.
4		Streetlight	0.68	The detected object is present in the image, but the confidence score for this detection is moderate, allowing it to be accepted or discarded.
5		Traffic light	0.80	The model incorrectly predicted an object's presence in the image, exhibiting excessive confidence with a high confidence score.
6	Missed detection	Traffic light	-	The model might not have seen enough examples of a particular class during training.

After the road asset detection process, frames of road assets detected were combined in a folder. The coordinates were then manually extracted into an Excel file as data for GIS purposes. The road asset mapping process starts with changing the projection for the coordinate system to 'WGS 1984' as the standard coordinate in Malaysia. Then, add 'Google Maps Imagery' from 'Portable Basemap Server' to get the world map and drag it as a layer. The longitude and latitude data were added from the Excel file. Afterward, it was exported as a shapefile to ensure the data point would show on the map, and that file was uploaded. Then, the symbology function was used in ArcGIS to represent the road assets, and suitable symbols and colours were chosen to differentiate them.

The Johor shapefile was added to the layer to show 'Jalan' and 'Daerah_Johor' as the location is in Johor. The map as layout was viewed, and details such as title, legend, scale, and North arrow were added to ensure the map layout is clear and visually appealing, as shown in Figure 9 with the title "Road Asset Mapping in Skudai, Johor" and the map can be exported into JPG file. As a result, the yellow dot represents a road sign, and the red dot represents a streetlight. From Jalan Pendidikan and Jalan Pontian Lama to Jalan Pulai, Skudai, Johor, about 1.5 km, the model can detect 36 road assets, including nine road signs and 27 streetlights.

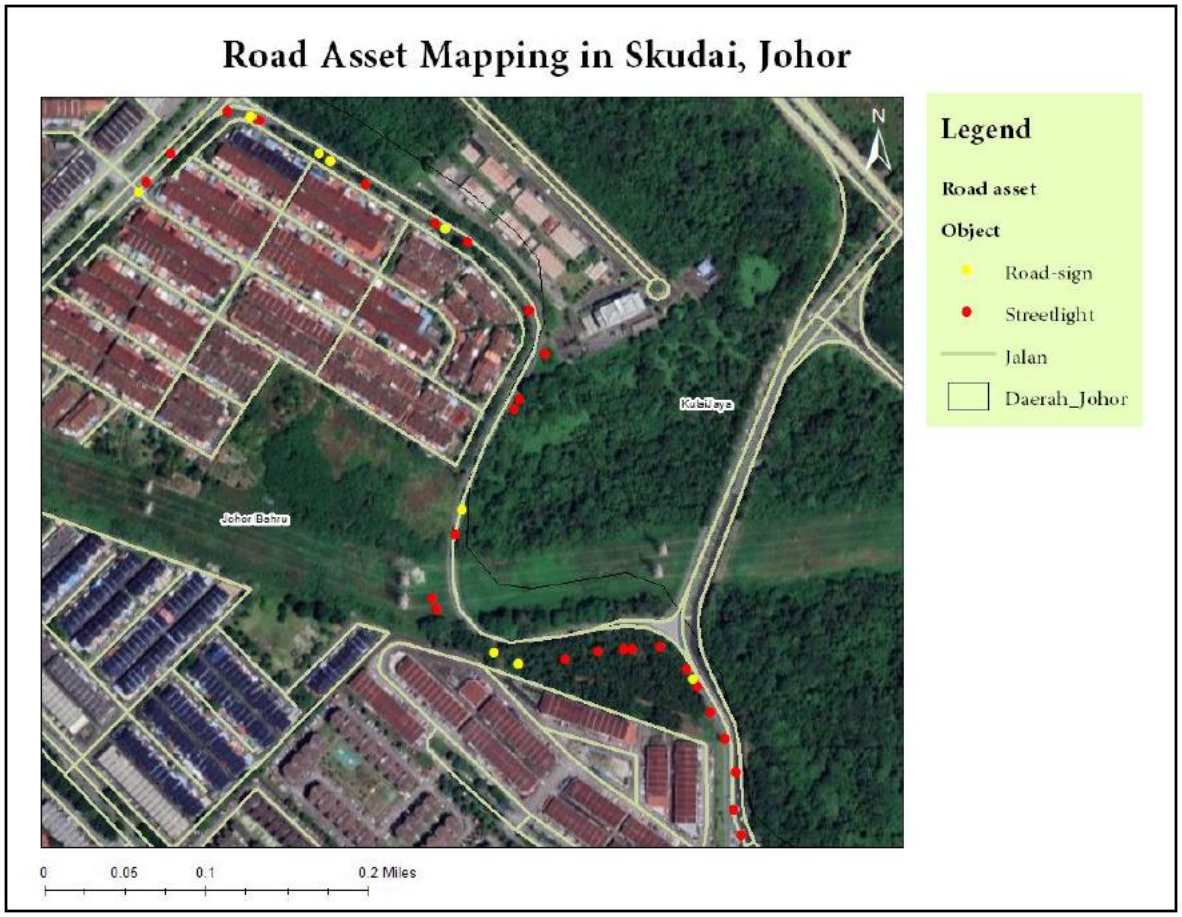

Figure 9: Road assets mapping in Skudai, Johor.

5 CONCLUSION

This study aims to develop a mapping system for road assets using YOLOv8 and GIS. The goal is to create a model capable of detecting road assets based on the available data, which can then be successfully mapped using GIS. However, the study has some limitations. The model fails to detect traffic lights and exhibits false detections (false positives) and missed detections (false negatives) for roads and streetlights. In a video that lasted 2 minutes and 5 seconds, comprising 430 frames used for testing the model, there were 29 false detections, 30 missed detections, and 36 successful detections. Among the false detections, 11 were related to road signs, 17 to streetlights, and 1 to traffic lights. For missed detections, there were 5 road signs, 20 streetlights, and 5 traffic lights that were not detected. Successful detections included 9 road signs and 27 streetlights. Overall, the model achieved an accuracy of only 37.89%. The accuracy for road signs was 36%, while streetlights' accuracy was 42.19%. These results indicate a significant need for improvement.

In comparison, a study in [16] utilized Trident-3D technology, incorporating advanced positioning sensors (GPS, INS, DMI), high-quality digital cameras, laser scanners, and photogrammetry. This approach demonstrated high efficiency in large-scale data collection and asset management, achieving 100% accuracy for manual road sign detection and 99% for automated data extraction. Consequently, the low-cost method of detecting road assets using a dashcam is less accurate or effective than the high-cost mobile mapping camera systems.

This study primarily recommends providing sharper, more precise, and higher quality videos. This improvement will enable the model to learn from accurate and representative examples, directly enhancing its ability to identify and classify objects under specific conditions. As a result, the model's performance can be improved. Additionally, a large dataset is essential for training effective object detection models. Large datasets typically include objects, backgrounds, lighting conditions, perspectives, and variations. This variety helps the model learn to recognize objects in different scenarios and improves its ability to generalize to new, unseen data.

REFERENCES

[1] N. Sairam, S. Nagarajan, and S. Ornitz, "Development of mobile mapping system for 3D road asset inventory," *Sensors*, vol. 16, no. 3, p. 367, Mar. 2016, doi: 10.3390/s16030367.

[2] C. Tao and J. Li, "Advances in Mobile Mapping Technology," ISPRS Book Series in Photogrammetry, Remote Sensing and Spatial Information Sciences, Taylor & Francis, London, ISBN 978-0-415-42723-4, 2007.

[3] K. Chiang, G. Tsai, and J. C. Zeng, "Mobile mapping technologies," in *Urban Informatics*, W. Shi, M.F. Goodchild, M. Batty, M.P Kwan and A. Zhang, Eds. Singapore: Springer Nature, 2021, pp. 439-465.

[4] B. Yang, "Developing a mobile mapping system for 3D GIS and smart city planning," *Sustainability*, vol. 11, no. 13, p. 3713, Jul. 2019, doi: 10.3390/su11133713.

[5] J. L. Binangkit and D. H. Widyantoro, "Increasing accuracy of traffic light color detection and recognition using machine learning," *in Int. Conf. Telecommun. Syst. Serv. Appl. (TSSA)*, 2016, pp. 1–5, doi: 10.1109/TSSA.2016.7871074.

[6] A. A. Nandargi, A. Bagewadi, A. Joshi, A. Jadhav, and A. S. Tigadi, "Understanding the perception of road segmentation and traffic light detection using machine learning techniques," *Int. J. Recent Technol. Eng*., vol. 9, no. 1, pp. 2698–2704, 2020, doi: 10.35940/ijrte.A3103.059120.

[7] H. Domínguez, A. C. Morcillo, M. Soilán, and D. G. Aguilera, "Automatic recognition and geolocation of vertical traffic signs based on artificial intelligence using a low-cost mapping mobile system," *Infrastructures*, vol. 7, no. 10, p. 133, Oct. 2022, doi:10.3390/infrastructures7100133.

[8] Z. Qiu, J. Martínez-Sánchez, V. Brea, P. López, and P. Arias, "Low-cost mobile mapping system solution for traffic sign segmentation using Azure Kinect," *Int. J. Appl. Earth Obs. Geoinf*., vol. 112, p. 102895, 2022, doi: 10.1016/j.jag.2022.102895.

[9] M. Sieverts, Y. Obata, M. Farhadmanesh, D. Sacharny, and T. C. Henderson, "Automated road asset data collection and classification using consumer dashcams," in *Int. Conf. Multisens. Fusion Integr. Intell. Syst. (MFI)*, 2022, doi: 10.1109/MFI.2022.9913859.

[10] A. Campbell, A. Both, and Q. Sun, "Detecting and mapping traffic signs from Google Street View images using deep learning and GIS," *Comput. Environ. Urban Syst.*, vol. 77, p. 101350, Sep. 2019, doi: 10.1016/j.compenvurbsys.2019.101350.

[11] M. Elhashash, H. Albanwan, and R. Qin, "A review of mobile mapping systems: From sensors to applications," *Sensors*, vol. 22, no. 11, p. 4262, Jun. 2022, doi: 10.3390/s22114262.

[12] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Glob. Transit. Proc.*, vol. 3, no. 1, pp. 91–99, 2022, doi: 10.1016/j.gltp.2022.04.020.

[13] R. Ju and W. Cai, "Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm," *Sci. Rep.*, vol. 13, no. 1, 2023, doi: 10.1038/s41598-023-47460-7.

[14] H. Lou et al., "DC-YOLOV8: Small-size object detection algorithm based on camera sensor," *Electronics*, vol. 12, no. 10, p. 2323, May 2023, doi: 10.3390/electronics12102323.

[15] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOV1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680–1716, Oct. 2023, doi: 10.3390/make5040083.

[16] T. Kingston, V. Gikas, C. B. Laflamme, and C. Larouche, "An integrated mobile mapping system for data acquisition and automated asset extraction," Presented at the Int. Conf. Mobile Mapping Technol., 2007. Available: https://www.isprs.org/proceedings/xxxvi/5-c55/papers/kingston_tara.pdf