Sustainable Built Environment National Research Centre AUSTRALIA

MRWA-Curtin Project

Mapping Main Roads Line Marking Inventory using Machine Learning

CASE STUDY REPORT

Prepared by	Shuyuan Xu and Peng Wu, Curtin University, Nicholas Archer, Lalinda Karunaratne and Qindong Li, MRWA
Status	Final
Date	3 May 2022
Version Number	2

Acknowledgements:

This research has been developed with support provided by Australia’s Sustainable Built Environment National Research Centre (SBEnrc). SBEnrc develops projects informed by industry partner needs, secures national funding, project manages the collaborative research and oversees research into practice initiatives. Core Members of SBEnrc include ATCO Australia, BGC Australia, Government of Western Australia, Queensland Government, Curtin University, Griffith University, RMIT University and Western Sydney University. This research would not have been possible without the valuable support of our core industry, government and research partners.

Project Leader: Peng Wu, Curtin University

Researchers:

Shuyuan Xu and Peng Wu (Curtin University)

Project Partners:

Nicholas Archer, Lalinda Karunaratne and Qindong Li (Main Roads Western Australia)

Citation: SBEnrc (2022) MRWA-Curtin Project, Mapping Main Roads Line marking Inventory using Machine Learning, SBEnrc 2.82 – Digitally-enabled Asset Life-cycle Management. https://sbenrc.com.au/research-programs/2-82/

Executive Summary

The BIM/DE theme has been one of the main, and growing, research themes led by SBEnrc since 2013. This project "Mapping Main Roads Line Marking Inventory using Machine Learning" is a continuation of the BIM/DE theme and builds on findings from previous projects related to the use of artificial intelligence in asset management.

This project aims to make full use of road pavement video data archived in Main Roads Western Australia (MRWA) and keep updated the inventory data of pavement markings for various managerial operations. To achieve this aim, a deep learning-based tool is developed to automatically identify different types of line markings in pavement images and make inference on line types. The developed tool has been validated on the Wheatbelt Region (where ground truth is available). The algorithm used for the detection of individual line marking achieves an accuracy of 0.96 on the validation dataset, and the tool achieves an average accuracy of over 0.80 for line type inference. In addition, the tool has been used to retrieve the inventory data in multiple regions.

TABLE OF CONTENTS

EXECUTIVE SUMMARY..3

1. BACKGROUND 5

2. PROJECT OVERVIEW 5

3. METHOD 5

3.1 INPUT AND OUTPUT OF THE DEVELOPMENT 6

3.2 PROCEDURE 7

4. THE DEVELOPED TOOL..10

5. VALIDATION ON THE WHEATBELT REGION.......................................12

6. PROJECT DELIVERABLES AND CONCLUSIONS...................................14

DISCLAIMER...14

REFERENCES..14

1. Background

Pavement markings and signs constitute the most fundamental way to communicate with road users and they are, in most cases, the most effective way to regulate, warn and guide traffic [1]. Advances in computer vision and deep learning in recent years have proven its capability to replace tedious manual work and extract information out of visual data. Additional support, e.g., GPU and parallel computing, will further boost its ability in industrial applications. In previous projects, image processing techniques and machine learning methods were used for the identification of line markings in pavement images [SBEnrc 2.64[1]]. Also, a deep learning-based tool has been developed to automatically detect individual line markings in road pavement videos [SBEnrc P2.51ARC[2]]. In this project, an improved deep learning-based tool is developed to conduct line marking detection and line type inference automatically and efficiently.

2. Project overview

This project aims to make full use of road pavement video data archived in Main Roads Western Australia (MRWA) and keep updated the inventory data of pavement markings for various managerial operations. To achieve this aim, a deep learning-based tool is developed to automatically identify different types of line markings in pavement images and make inference on line types. The developed tool has been validated on the Wheatbelt Region (where ground truth is available). The algorithm used for the detection of individual line marking achieves an accuracy of 0.96 on the validation dataset, and the tool achieves an average accuracy of over 0.80 for line type inference. In addition, the tool has been used to retrieve the inventory data in multiple regions.

3. Method

The developed tool is mainly based on a classic deep learning algorithm for object detection tasks, i.e., You Only Look Once (yolo) [2]. There are four steps included in the development, as illustrated in Figure 1.

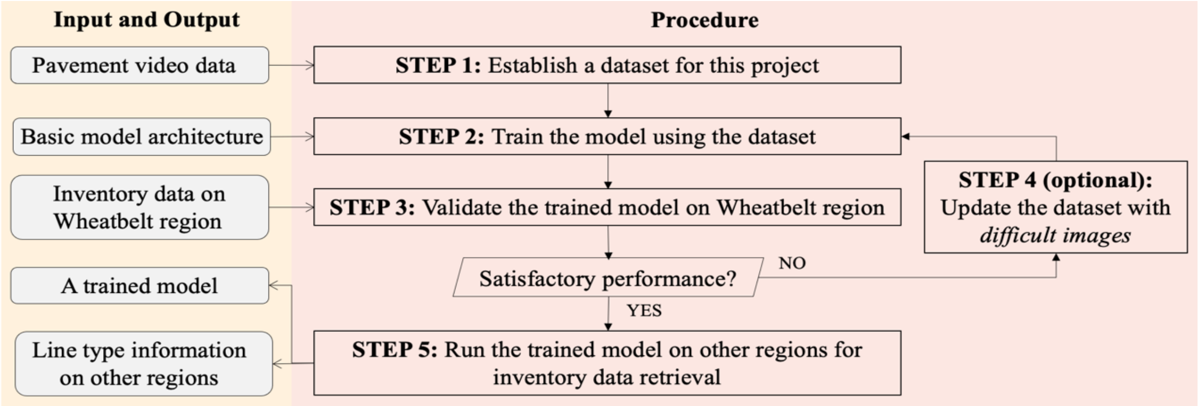

Input and Output
Procedure
Pavement video data
Basic model architecture
Inventory data on Wheatbelt region
A trained model
Line type information on other regions
STEP 1: Establish a dataset for this project
STEP 2: Train the model using the dataset
STEP 3: Validate the trained model on Wheatbelt region
Satisfactory performance?
NO
YES
STEP 4 (optional): Update the dataset with difficult images
STEP 5: Run the trained model on other regions for inventory data retrieval

Figure 1. Input, output, and procedure for the development

3.1 Input and output of the development

Input data

The data to be used for development include: (a) pavement video data on eight regions, (b) the open-source model architecture of yolo, and (c) the inventory data on Wheatbelt region. Specifically, randomly selected pavement images are first retrieved from (a). Images are then labelled for different types of line markings using the labelImg tool [3]. The images and corresponding label files together constitute the dataset for this project, which are further divided into training dataset and validation dataset.

A few notes on the input data that can cause detection errors include, i) there are a few videos (partly) corrupted, e.g., H053. ii) a file precisely mapping the Frame No. to the SLK is lacking for the video data. iii) a few pavement markings, such as painted median sections, islands in the middle of the road and kerbs, are diversified and difficult to include in the dataset for training.

Output data

The outputs of this project include: (a) a trained model for line marking detection; and (b) the processing results on other regions' pavement videos. Road regions included in this project are RA01, RA02, RA04, RA05, RA08, RA11, RA14. The processing results include: a) frames with marked line markings (which can be set to be optional), b) line marking detection results (including coordinates), c) line type identification results (raw), and d) line type identification results (modified).

3.2 Procedure

The development of this project can be best described by following five steps. Each step is illustrated in details and the corresponding input and output data are specified in Table 1.

Step 1 – Establish a dataset for this project

The development of line type identification tool starts with the establishment of dataset. Firstly, pavement images are retrieved from multiple videos, so as to ensure the diversity of the dataset in lighting conditions, occlusions and several environmental factors. Then, pavement images are labelled using the labelImg tool. The images and corresponding label files are stored in 'datasetes/road/labelimg/images' and 'datasets/road/labelimg/labels', respectively. To make the dataset trainable by the yolo, all the above label files will be converted to yolo format. Also, the dataset will be divided into training dataset and validation dataset and the ratio is set to be 8:2.

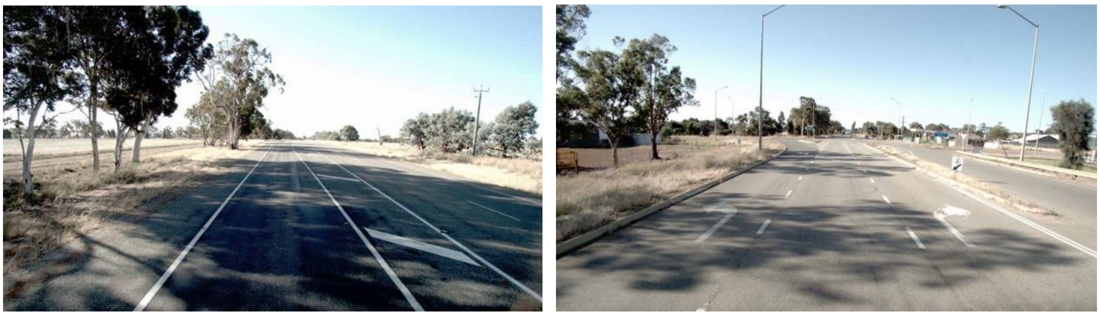
Step 2 – Train the model using the dataset

The model is trained on the training dataset and validated on the validation dataset. It is specified that the dataset path is 'ROOT/datasets/road/yolo/ and the basic model path is 'ROOT/yolov5'. During this step, according to the detection performance on the validation dataset, the model can be sent back for re-training. An updated dataset is required, by adding more diversified pavement images and improving the quality of labelling.

Step 3 – Validate the trained model on Wheatbelt region

Since the inventory on Wheatbelt region is available, the trained model is validated on this region. For each video, line markings are first identified and combined to allow for line type inference on each frame. The line type inference is achieved by first comparing to the line type definition and then referring to the nearest six line types for modification.

As a result, each frame is assigned a line type code as well as a TRUE/FALSE assessment after a comparison to the ground truth. The overall detection accuracy is lastly calculated as the ratio of TRUE among all valid frames. By using 'valid frames', we have excluded the following two types of frames: i) frames with multiple rows of line types according to the ground truth and ii) frames with large markings OR land in the middle of the pavement (which are not assigned with a line marking code). Example frames are shown in Figure 2 for the second case, in particular.



(a) Large markings in the middle (b) Land in the middle

Figure 2. Examples of non-valid frames

Step 4 (Optional) – Update the dataset with difficult images

When the performance of the developed method on Wheatbelt region is not satisfactory, we sent back the model for retraining. To improve the detection accuracy, *difficult images* are identified according to the FALSE detection during the validation and the dataset is updated by adding those difficult images. The update of training dataset aims to increase the diversity, e.g., lighting conditions, occlusions and pavement condition.

Step 5 – Run the trained model on other regions for inventory retrieval

The last step is to apply the trained model for video processing on a wider range of road videos. The detection results for each pavement video include: a) a frames folder containing images with marked bounding boxes on line markings, b) a table of individual line markings and their coordinates, and c) tables of line type detection on each frame (raw and modified).

Table 1. Procedures and their corresponding input and output

Steps	**Input**	**Output**
Step 1	Multiple pavement videos	A dataset for pavement markings
Step 2	The dataset Basic model, i.e., yolo	A trained model (including weights and model performance)
Step 3	The trained model Pavement videos of Wheatbelt region	Validated results on line type detection
Step 4	More pavement videos	An updated dataset
Step 5	The trained model Pavement videos of other regions	Detection results

4. The developed tool

Figure 3 (a) presents the composition of the tool's source code. There are a few key files/folders:

- The *datasets* folder. The *datasets* folder under the main directory contains the video data that need to be processed. The path will be ROOT/datasets/video/.
- The *labelimg2yolo.py* file. The label files generated by the labelImg tool are in the form of .xml. The python executive file *labelimg2yolo* is used to convert the label file into the form that can be used by the yolo model.

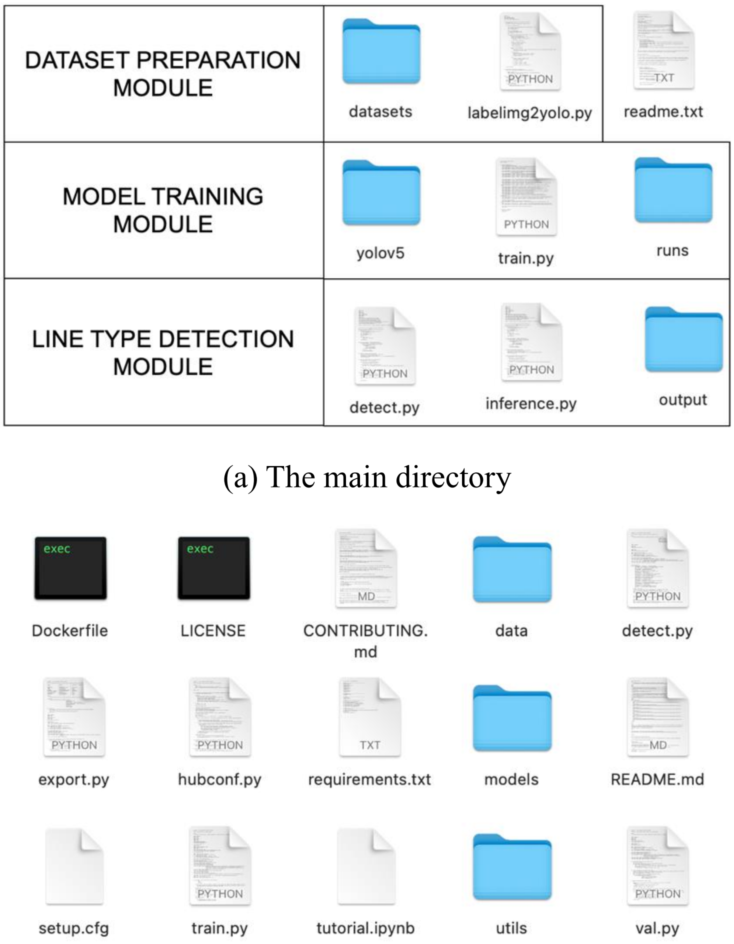
The *datasets* folder and the *labelimg2yolo.py* file constitute the module for DATASET PREPARATION (i.e., Step 1 and Step 4).

- The *yolov5* folder. The *yolov5* folder under the main directory contains the yolo model for line marking detection (as presented in Figure 2(b)) while the weights for this project (i.e., road pavement markings) are stored in the ROOT/runs/train/exp/weights.
- The *train.py* file. The python executive file *train* can be used to train the yolo model on a new dataset.
- The *runs* folder. The *runs* folder mainly contains the weights and parameters of the model as the training output.

The *yolov5* folder, the *train.py* file and the *runs* folder together constitute the module for MODEL TRAINING (i.e., Step 2).

- The *inference.py* file. The python executive file *inference* is mainly used to process new road pavement videos (with inventory data) and evaluate the accuracy for line type identification.
- The *detect.py* file. The python executive file *detect* is mainly used to process new road pavement videos (without inventory data) and retrieve the inventory data of pavement markings.
- The *output* folder. The *output* folder under the main directory contains the processing results for each pavement video.

The *inference.py* file, the *detect.py* file, and the *output* folder constitute the model for LINE TYPE DETECTION (i.e., Step 3 and Step 5).



(a) The main directory

(b) The *yolov5* model folder

Figure 3. Compositions of the source code

The usage of the developed tool can be found in a separate *readme* file.

After training on the road pavement dataset, the proposed method can achieve a mean average precision (mAP) of 0.96 on the validation dataset for the detection of line markings. Table 2 presents the validation results on different types of line markings.

Table 2. Accuracy of line marking detection (on the validation dataset)

Line marking type	barrier-line - one	barrier-line -both	edge-line	Dividing-line	continuity-line
mAP	0.993	0.989	0.993	0.956	0.866

5. Validation on the Wheatbelt region

The validation results on the Wheatbelt region are stored in the ROOT/validation results/.

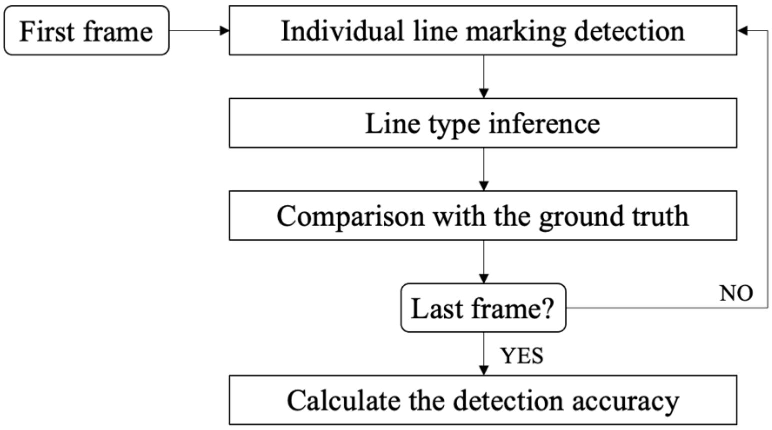
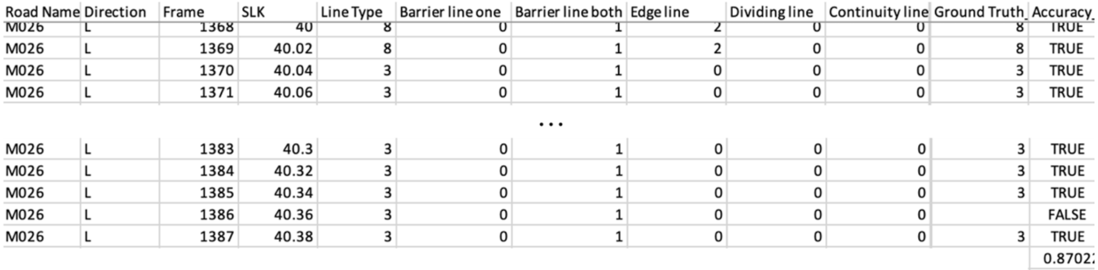

First frame
Individual line marking detection
Line type inference
Comparison with the ground truth
Last frame?
NO
YES
Calculate the detection accuracy

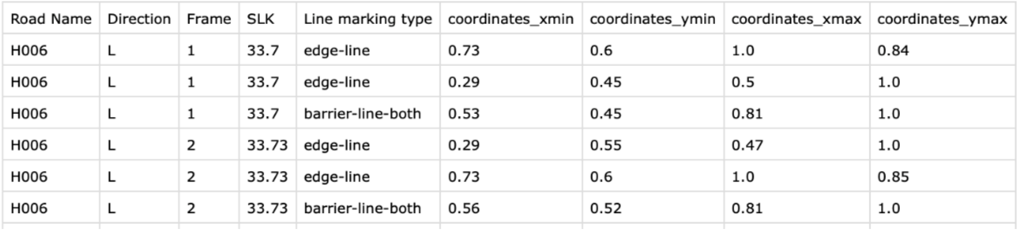
Figure 4. Procedures for validation on the Wheatbelt region

The validation process follows the flowchart illustrated in Figure 4. For each road pavement video, detection of individual line markings on each frame is conducted using the trained yolo algorithm. Two formats of detection results are shown in Figure 5. The line type inference is documented as Figure 5(a) and the accuracy check is performed. The types of line markings and their locations in the image (i.e., the coordinates of the bounding box that highlight the identified marking) are retrieved and documented accordingly, as shown in Figure 5(b).



Road Name	Direction	Frame	SLK	Line Type	Barrier line one	Barrier line both	Edge line	Dividing line	Continuity line	Ground Truth_	Accuracy_
M026	L	1368	40	8	0	1	2	0	0	8	TRUE
M026	L	1369	40.02	8	0	1	2	0	0	8	TRUE
M026	L	1370	40.04	3	0	1	0	0	0	3	TRUE
M026	L	1371	40.06	3	0	1	0	0	0	3	TRUE
					…						
M026	L	1383	40.3	3	0	1	0	0	0	3	TRUE
M026	L	1384	40.32	3	0	1	0	0	0	3	TRUE
M026	L	1385	40.34	3	0	1	0	0	0	3	TRUE
M026	L	1386	40.36	3	0	1	0	0	0		FALSE
M026	L	1387	40.38	3	0	1	0	0	0	3	TRUE
											0.8702

(a)



Road Name	Direction	Frame	SLK	Line marking type	coordinates_xmin	coordinates_ymin	coordinates_xmax	coordinates_ymax
H006	L	1	33.7	edge-line	0.73	0.6	1.0	0.84
H006	L	1	33.7	edge-line	0.29	0.45	0.5	1.0
H006	L	1	33.7	barrier-line-both	0.53	0.45	0.81	1.0
H006	L	2	33.73	edge-line	0.29	0.55	0.47	1.0
H006	L	2	33.73	edge-line	0.73	0.6	1.0	0.85
H006	L	2	33.73	barrier-line-both	0.56	0.52	0.81	1.0

(b)

Figure 5. Two formats for detection results

In both formats, there are columns sharing the background information of the processed video, i.e., the road name and direction (Left or Right). There are columns documenting the frame number (starting from 1) and the corresponding SLK, as well. Especially in Format (a), columns [‘Barrier line one’], [‘Barrier line both’], [‘Edge line’], [‘Dividing line’] and [‘Continuity line’] document the number of each line marking type identified in a given frame. And based on the line marking detection results, a line type is inferred according to the *line type definition* [4] and documented in the column [‘Line Type’]. To facilitate the accuracy check, the ground truth (i.e., the line type archived in the inventory database for each SLK location) is put in column [‘Ground Truth’] and the last column [‘Accuracy’] represents whether the detection results match the ground truth (TRUE) or not (FALSE). For example, in Frame no.1369, one barrier line both-way and two edge lines are identified, leading to an inference of line type ‘eight’. The inference matches the ground truth (i.e., line type ‘eight’) and therefore, the detection is ‘TRUE’.

After a comparison to the ground truth, the accuracy of line type detection is calculated for each video and presented in Table 3.

Table 3. Validation results on the Wheatbelt region

Road Name	**Accuracy**	**Road Name**	**Accuracy**
H005	0.79	M020	0.91
H006	0.91	M026	0.87
H053	0.94	M028	0.94
H923	0.84	M031	0.93

M001	0.98	M033	0.95
M016	0.96	M038	0.77
M017	0.93	M040*	N/A

*The M040 road segment has no ground truth documented.

6. Project deliverables and conclusions

The deliverables of this project will include the following:

1. The source code of the developed tool for line type detection and its guideline.
2. Detection results (in two formats of excel sheets) for eight regions of road pavement.
3. A formal report on this project (*This file*).
4. Presentation slides for monthly updates and the final meeting.

Based on an early machine learning proof-of-concept for line marking detection, developed by the research team, we have modified and applied the tool for the whole road network of West Australia. The developed line marking detection method has satisfactory capability to save tedious manual work, automate image processing and generate reliable detection results.

Disclaimer

Please note the views expressed in the report do not reflect the view of Main Roads WA.

References

1. *Guide to Traffic Management Part 10: Traffic Control and Communication Devices*. 2019, Austroads Ltd. p. 229.
2. Jocher;, G., et al., *yolov5 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference.* 2022.
3. Tzutalin. *LabelImg*. Git code 2015; Available from: https://github.com/tzutalin/labelImg.

4. Australia, M.R.W., *WBR Longitudinal Line markings inventory database (Original at D16#165926)*. 2021.