NACOE
NATIONAL ASSET CENTRE OF EXCELLENCE

ANNUAL SUMMARY REPORT

R54 – Automated collection of AusRAP road attributes using DVR and Pattern Recognition Techniques – Y2 (2018/19)

ARRB Project No.: PRS18072

Author/s: Prof Brijesh Verma, Dr Joseph Affum & Dr Mingyang Zhong

Prepared for: Queensland Department of Transport and Main Roads

9/09/2019

v1

AN INITIATIVE BY:

arrb Queensland Government

SUMMARY

This interim report presents the proposed methods, experiments, results and future directions for the project entitled automated collection of AusRAP attributes using DVR, MLS and pattern recognition. The proposed 3-D segmentation and classification method using MLS data and 2-D segmentation and classification method using DVR data for identifying AusRAP attributes are presented. The distance calculation techniques for both MLS and DVR data are described. The proposed methods are implemented in Python programming language and incorporated in development of software for automatically identifying AuSRAP attributes. The proposed methods are tested on a large training and testing data. The experimental results and future directions are presented in this report.

Although the Report is believed to be correct at the time of publication, the Australian Road Research Board, to the extent lawful, excludes all liability for loss (whether arising under contract, tort, statute or otherwise) arising from the contents of the Report or from its use. Where such liability cannot be excluded, it is reduced to the full extent lawful. Without limiting the foregoing, people should apply their own skill and judgement when using the information contained in the Report.

Queensland Department of Transport and Main Roads Disclaimer
While every care has been taken in preparing this publication, the State of Queensland accepts no responsibility for decisions or actions taken as a result of any data, information, statement or advice, expressed or implied, contained within. To the best of our knowledge, the content was correct at the time of publishing.

CONTENTS

1 INTRODUCTION

Automation of the extraction of road attributes from DVR using advanced image analysis and deep learning, and cross-validation with other data sources such as MLS, has the potential to provide a range of value-added products consistently and inexpensively: road condition (e.g. deflection, cracking, rutting), road safety (e.g. selected AusRAP attributes and iRAP star rating compliance), environmental (e.g. fire risk and vegetation encroachment) and improved obstacle clearance estimates (e.g. overhead wires, roadside hazards, heavy vehicle widths).

The goals of this project are to review MLS and DVR data, create training and testing datasets, develop and evaluate software for automatically identifying AusRAP attributes. The project milestones are presented below in Section 1.1.

1.1 MILESTONES

Milestone 1 – Review and analysis of MLS and DVR data for AusRAP attributes

Review of AusRAP attributes from MLS and DVR data

Analysis of usefulness of new AusRAP attributes from MLS & DVR data

Milestone 2 – Creation of large training and testing datasets from MLS and DVR data

Extraction of attributes from MLS and DVR data

Analysis of MLS attributes that can be used with DVR attributes for improving performance

Creation of large training and testing datasets

Conduction of experiments

Milestone 3 – Identification and proximity measurement techniques

Implementation of techniques

Evaluation of techniques

Milestone 4 – Software for automatically determining attributes

Development of a software by incorporating learning and identification of attributes

Evaluation of software on large number of attributes

2 REVIEW AND ANALYSIS OF DATA FOR AUSRAP ATTRIBUTES

We have reviewed many roads and AusRAP attributes from DVR and MLS data and found that the 26 attributes listed in Table 2-1 are suitable and can be extracted from video data. We have also reviewed sub-attributes which can be extracted from video data.

Table 2-1. List of the Australian Road Assessment Program (AusRAP) attributes and sub-attributes

No	AusRAP Attributes	Sub-attributes
1	Area type	Buildings
2	Speed limit	10, 20, 30, 40, 60, 100, 110
3	Median type	Line, Metal barrier, Concrete barrier, Median concrete, Grass, Flexipost
4	Centreline rumble strips	
5	Roadside severity - driver-side distance	Road
6	Roadside severity - driver-side object	Pole, Metal barrier, Concrete barrier, Median concrete, Tree, Guide post
7	Roadside severity - passenger-side distance	
8	Roadside severity - passenger-side object	Pole, Metal barrier, Concrete barrier, Median concrete, Tree, Guide post
9	Shoulder rumble strips	
10	Paved shoulder - driver-side	Road
11	Paved shoulder - passenger-side	
12	Intersection type	Roundabout sign, Merge lane sign, Signal sign, Railway sign
13	Number of lanes	
14	Lane width	
15	Quality of curve	Curvature sign
16	Road condition	Defect
17	Delineation	Warning sign, Guide post, Line markings
18	Street lighting	
19	Pedestrian crossing facilities - inspected road	Pedestrian crossing, Pedestrian sign
20	Pedestrian fencing	
21	Sidewalk - passenger-side	
22	Speed management / traffic calming	Speed hump, Speed hump sign
23	Vehicle parking	Parking sign
24	Facilities for bicycles	Bicycle lane
25	Roadworks	Roadwork sign
26	School zone warning	School zone sign

2.1 MLS DATA

In order to investigate the detection of AusRAP attributes using MLS data, we used point cloud data provided by Department of Transport and Main Roads. We have obtained point cloud data from recent survey data for 4 roads including 10L, 10A, 14A and 210A. The data provided in the form of large LAS files have been analysed. The visualizations of samples from road 210A are shown below in Figures 1 and 2. Figure 2.1 visualizes a part of a LAS file of 210A containing approximately, 43,776,604 points.

Figure 2-1. Sample MLS data

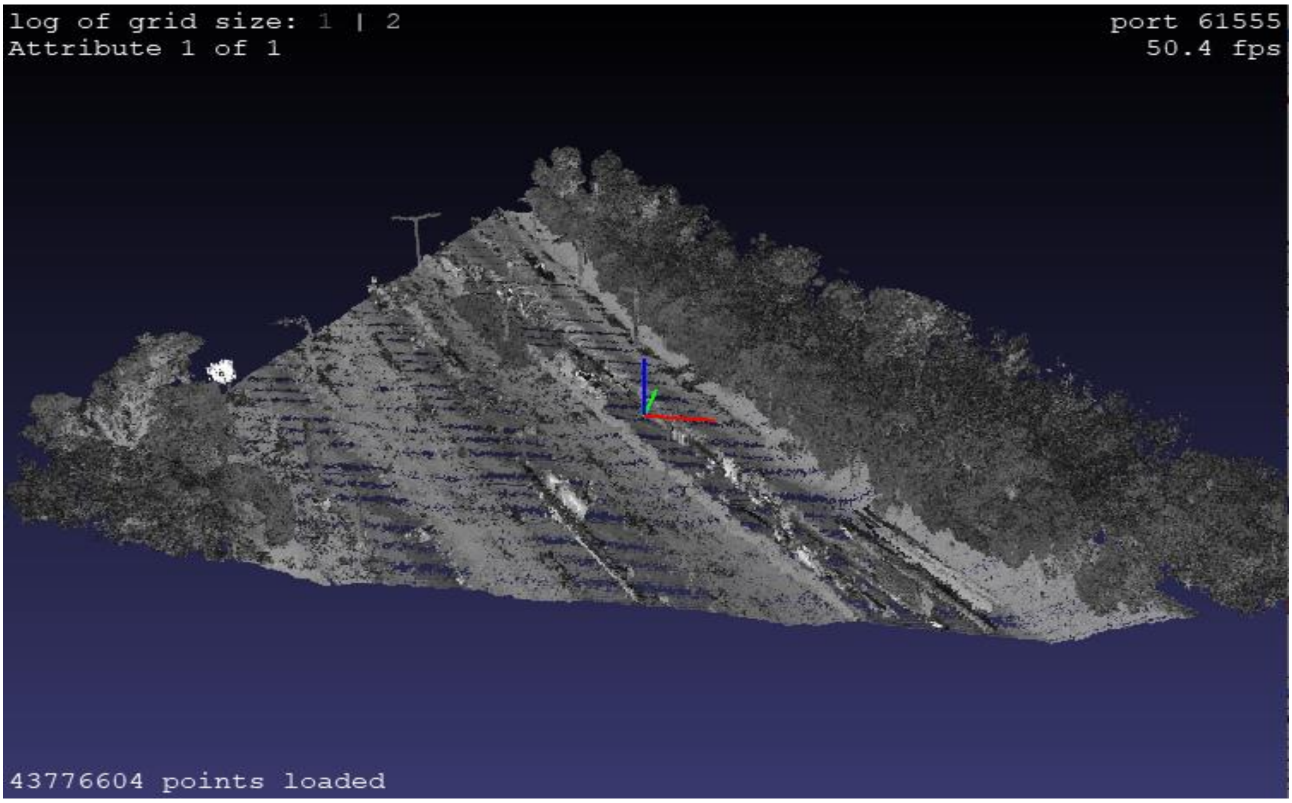
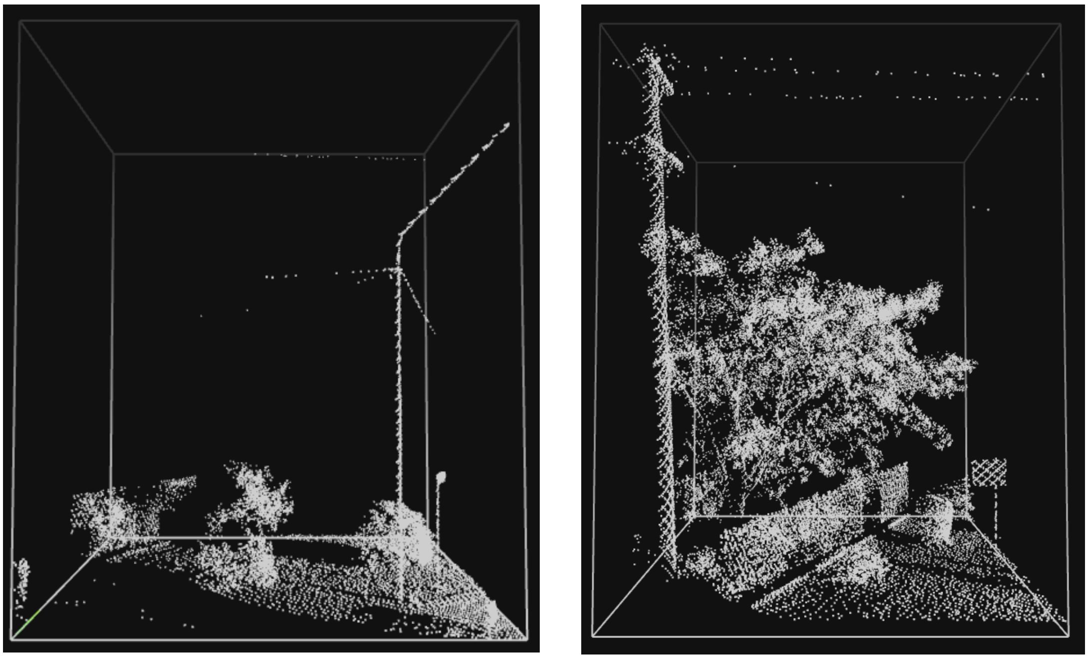


Figure 2-2. Details of sample MLS data including some attributes (pole, tree, road sign and ground)



2.2 DVR DATA

In order to automatically detect the AusRAP attributes using DVR data, we used video data provided by Department of Transport and Main Roads. We have obtained videos from recent survey data that contained four videos (left, right, front, rear) for each road (10L, 10A, 14A and 210A). After analysing the videos, we found that the video data from front camera are appropriate to extract the AusRAP attributes. From the video data, we have extracted frames which are used to extract the sub-attributes. The samples are shown below in Figures 2.3 and 2.4.

Figure 2-3. Sample image frame

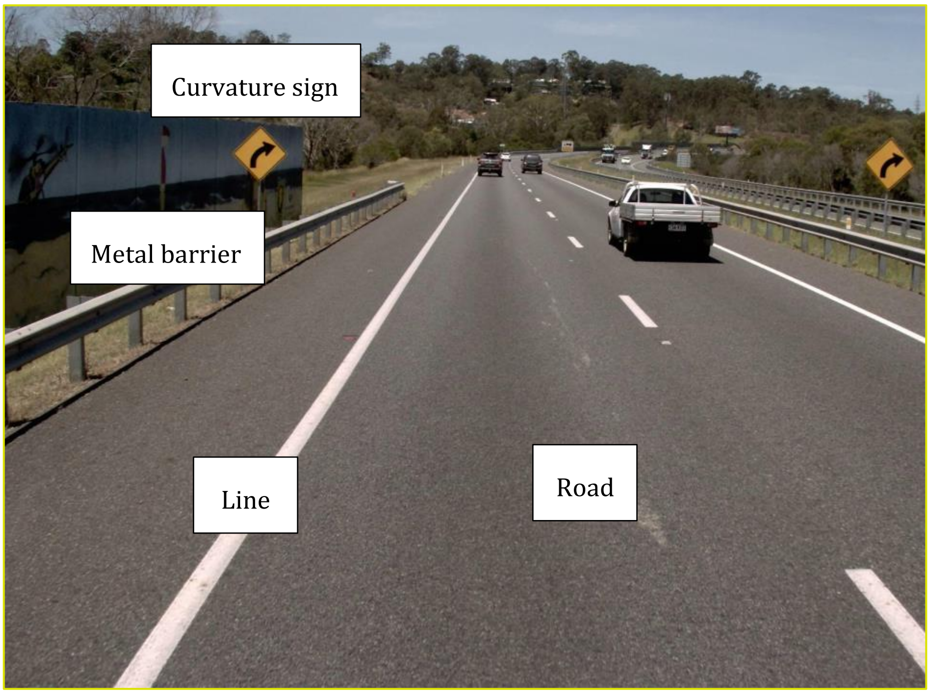


Figure 2-4. Sample image frames collected from different regions with new attributes such as bicycle lane, concrete barrier, speed limit sign and roadwork sign

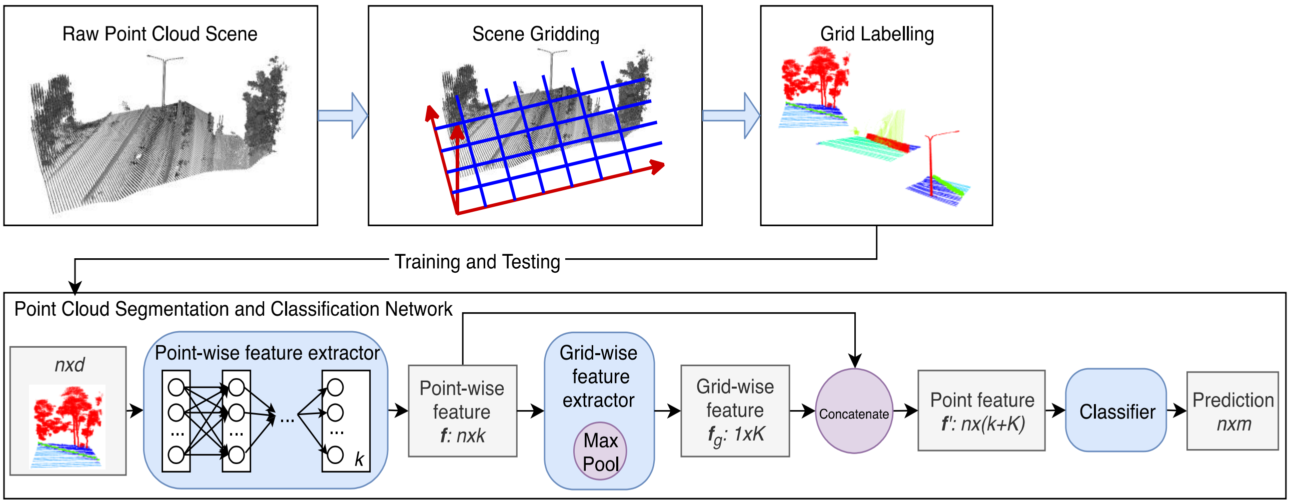
3 RESEARCH FRAMEWORK AND SOFTWARE IMPLEMENTATION

3.1 FRAMEWORK AND SOFTWARE FOR MLS DATA

Research works on object recognition use point cloud data collected by LiDAR sensors. Most of them employ a pipeline method that combines various handcrafted features or descriptors with a machine learning classifier and have problems for identifying poles and trees. Thus, poles and trees are generally classified into a pole-like category. Similarly, for semantic segmentation, structured output classifiers are used instead of the single output classifier. To overcome these limitations our study employed a different approach. Our framework learns to extract features and performs point-wise classification from raw point cloud data, resulted in extracted features that are richer than these handcrafted features.

An overview of the research framework is shown below in Figure 3-1. The proposed framework performs in a pipeline fashion. Taking the raw point cloud data as input, our framework first splits the point cloud of a large scene into small grids. Then these grids are manually annotated and fed to our Point Cloud Segmentation and Classification Network (PCSCN) for training and testing. For testing, with a trained PCSCN, the framework predicts point-wise classifications for each point in a grid, and then the output girds can be aligned back to the original roadside scene.

Figure 3-1. 3-D segmentation and classification



The proposed research framework has been implemented using Python programming language and the Tensorflow package. Algorithm 1 depicts the training process of the proposed framework using TensorFlow pseudocode. As for the testing process shown in Algorithm 2, after gridding the grids are fed to the framework that loads the trained weights, and then girds with point-wise predictions are outputted. Finally, the grids are aligned to generate the complete roadside scene with point-wise predictions.

Algorithm 1. Training process of the proposed framework.

Input: Raw point cloud data P and $p_i = (x_i, y_i, z_i, intensity_i)$, size of grid s, PCSCN training parameters such as *epoch*, *learning rate* and *batch size*

Begin

Gridding:

1. for i_x in *range*((*max*(x)-*min*(x))/s)
2. |--- for i_y in *range*((*max*(y)-*min*(y))/s)
3. |---|--- generate grid g, where g is a $n×4$ matrix of n points with 4 dimensions (*x, y, z, intensity*);

Labelling:

4. annotate the grids and output the annotated grids g's,

   where g' is a $n×5$ matrix of n points with 5 dimensions (*x, y, z, intensity, label*).

PCSCN training:

5. initialize the weights of PCSCN;
6. *optimizer = AdamOptimizer*(*learning rate*);
7. *loss = sparse_softmax_cross_entropy_with_logits*;
8. for e in *range*(*epoch*);
9. |--- for *batch* in *range* (*batch size*);
10. |---|--- feed the $n×4$ point cloud of *batch* to the point-wise feature extractor (shared MLPs). Taking the last layer of the MLPs as an example, the k dimension point-wise feature vector $\boldsymbol{f}_i$ is computed:

    |---|--- $\boldsymbol{f}_i = \varnothing(\boldsymbol{w}\cdot\boldsymbol{x}+\boldsymbol{b})$, where $\varnothing$ is the activation function, x is the feature vector of each point. $\boldsymbol{w}$ and $\boldsymbol{b}$ are the weights and the bias of this layer;
11. |---|--- feed $[\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n]^T$ of the n points to the grid-wise feature extractor and compute the K dimension grid-wise feature $\boldsymbol{f}_g$ by *max pooling*;
12. |---|--- concatenate $[\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n]^T$ of the n points and $\boldsymbol{f}_g$ into a new $n×(k+K)$ point-wise feature matrix $[\boldsymbol{f}_1', \ldots, \boldsymbol{f}_n']^T$;
13. |---|--- feed the new point-wise feature matrix $[\boldsymbol{f}_1', \ldots, \boldsymbol{f}_n']^T$ to the classifier (shared MLPs similar to line 10) to predict the $n×m$ output, where m is the number of semantic categories;
14. |---|--- back propagate to minimize loss;
15. save the trained weights of PCSCN.

End

Output: The weights of PCSCN

Algorithm 2. Testing process of the proposed framework.

Input: Raw point cloud data P and $p_i = (x_i, y_i, z_i, intensity_i)$, size of grid s, PCSCN trained parameters and batch size

Begin

Gridding:

1. for i_x in *range*((*max*(x)-*min*(x))/*s*)
2. |--- for i_y in *range*((*max*(y)-*min*(y))/*s*)
3. |---|--- generate grid *g,* where *g* is a *n×4* matrix of *n* points with 4 dimensions (*x, y, z, intensity*);

PCSCN testing:

4. load the trained weights of PCSCN;
5. for *batch* in *range* (*batch size*);
6. |--- compute the prediction class of each point
7. output the prediction result of the input.

End

Output: The prediction result of the input

3.1.1 DISTANCE MEASUREMENT TECHNIQUE FOR MLS

When calculating distances on MLS data, selecting the points for the distance calculation is essential. Thus, we proposed an object centre approximation method for distance calculation. The key idea is shown in Figure 3-2. We first project 3D space onto a 2D plane. As for poles, the projection is roughly a line shape. Then, we find a line that connects the two far ends of the projection. Finally, we compute the density of points along the line, and the point on the line with highest density is set as the centre of the pole. As for trees, the projection is roughly a circle shape and we assume that trees are symmetric. Then, the centre of the circumscribed square of the circle is set as the centre of the tree. After computing the centre of an object, we can calculate the distance between the centre and any other point.

Figure 3-2. Object centre approximation

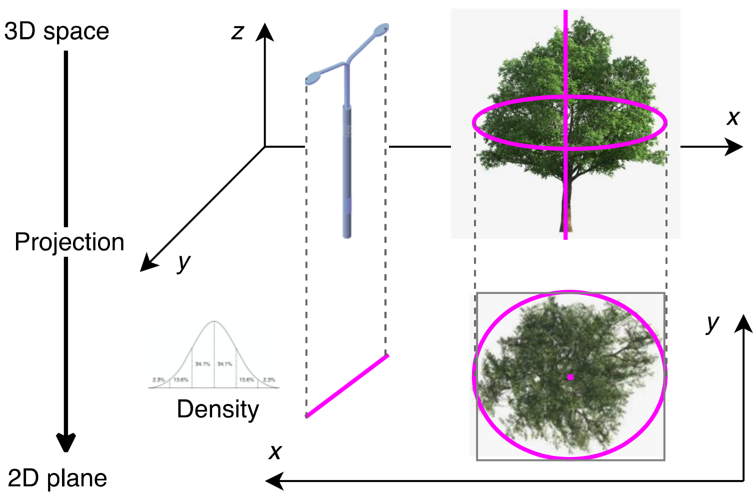


Figure 3-3 is an evaluation example, assuming the trees and lines are identified, the shortest distance between a tree and a road edge line can be approximated as the perpendicular distance from the centre of the tree $p_0 = (x_0, y_0)$

computed by the aforementioned object centre approximation to the line defined by two points, $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ that are on the edge line. Finally, the distance d can is computed as:

$$d = \frac{|mx_0 \;-\; y_0 \;+\; b|}{\sqrt{m^2+1}}$$

where $m = \frac{y_2 - y_1}{x_2 - x_1}$ and $b = y_1 - m \cdot x_1$. The actual shortest distance is computed from the centre of the tree p_0 = (492828.609, 6945953.902) to the line defined by two points p_1 = (492816.849, 6945957.679) and p_2 = (492830.518, 6945946.423). The shortest distance is 3.52 meters in this example.

Figure 3-3. Example of distance calculation

3.2 FRAMEWORK AND SOFTWARE FOR DVR DATA

We have implemented the proposed 2-D segmentation and classification framework using a Convolutional Neural Network (CNN). The implementation is done in Python programming language and using the Tensorflow package. The framework is shown in Figure 3-4. Stepwise process of the software is detailed as follows:

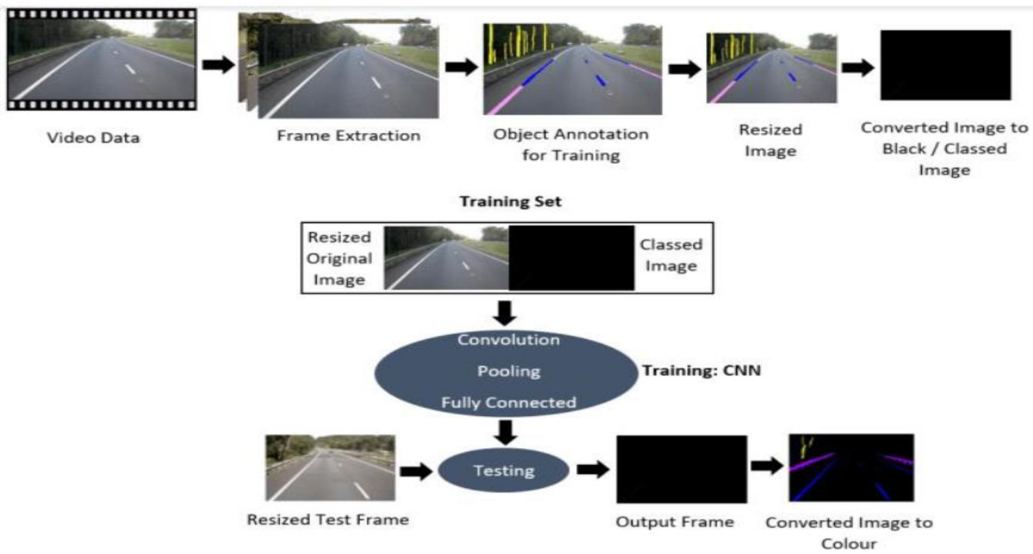
Step 1. Extract frames from video data.

Step 2. Prepare training set (input frames with original attributes and target frames with labelled pixels) by localising AusRAP attributes and labelling frames.

Step 3. Train the segmentation and classification network by optimising CNN parameters.

Step 4. Test the trained network.

Figure 3-4. 2-D segmentation and classification using CNN



3.2.1 DISTANCE MEASUREMENT TECHNIQUE FOR DVR

When calculating distances and widths, the line markings are used as a reference point, due to their reliable width. Figure 3-5 demonstrates how this is used. Stepwise process of the software is detailed as follows:

Step 1. Find all objects in the frame.

Step 2. Measure the known line marking width from a road and make the width as standard for further measurement. An example is shown in Figure 3-5, as shown in figure the line width is 20 cm and the distance between the pole and the line is 280 cm.

Step 3. Calculate the number of actual pixels in the specific region (e.g. line width).

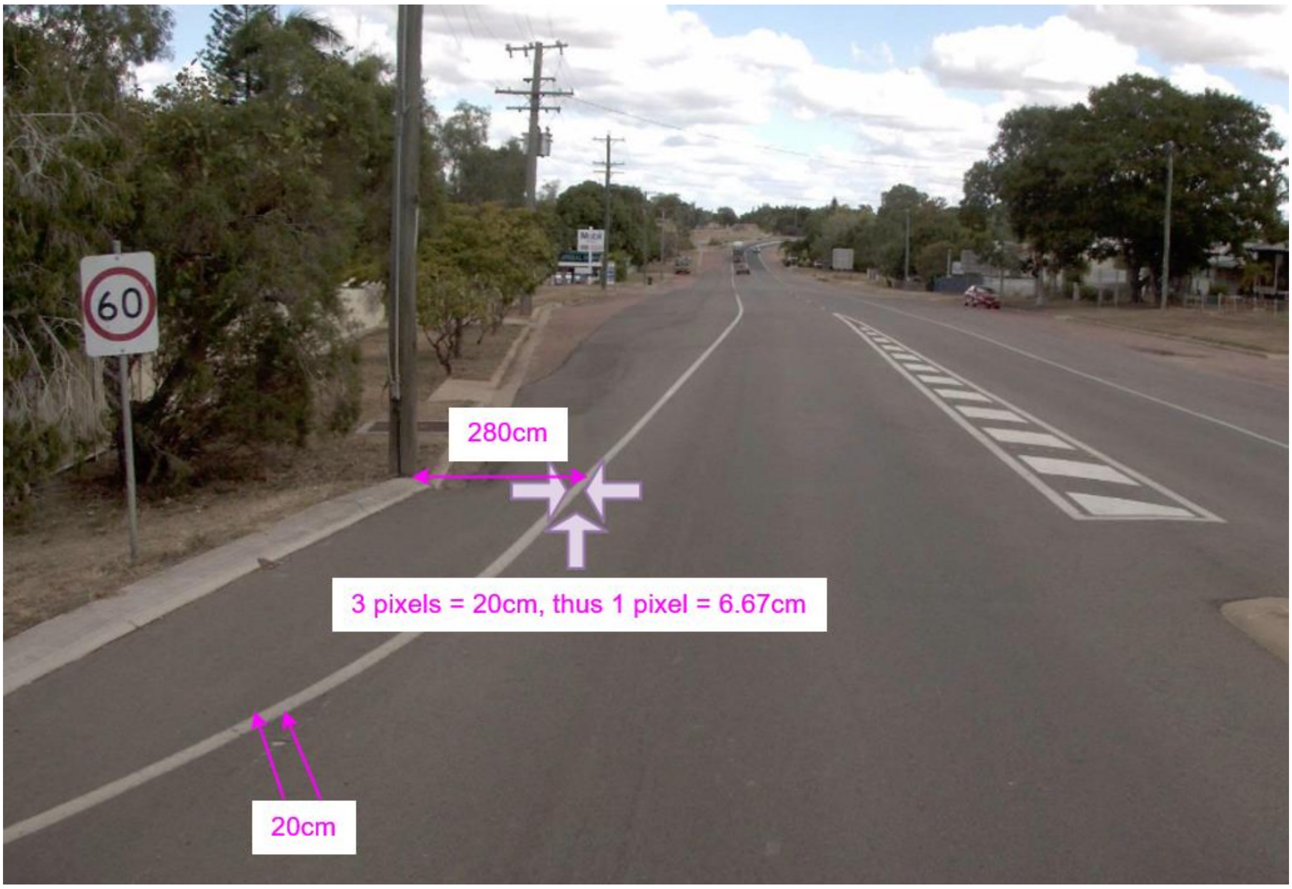
Step 4. Calculate the width in terms of centimetre for each pixel.

Step 5. In each row, pixelwise distance will vary. Thus, calculate a weight for each row.

Step 6. To calculate distance between the line and objects (e.g. tree object), first identify the tree region and figure out the base pixel. As shown in figure, the base pixel is chosen from the connected component of the tree.

Step 7. Finally, start from base pixel and move towards the line pixel column wise. After reaching/touching line, calculate the number of pixels from base pixel to shoulder line and then multiply by pixelwise width to convert it into meters.

Figure 3-5. Example of distance calculation based on DVR data



3.3 IDENTIFIED ATTRIBUTES FOR MLS DATA AND DVR DATA

The 3-D model for MLS data has been trained to identify 5 different attributes in roadside point cloud data. The 2-D model has been trained to identify 29 attributes and sub-attributes in roadside images. The 2-D model using DVR data attribute numbers 1-9, 12-19 and 22-26 listed in Table 2-1. The trained/identified attributes for both MLS and DVR data are listed below in Table 3-1.

Table 3-1. List of the currently identified attributes

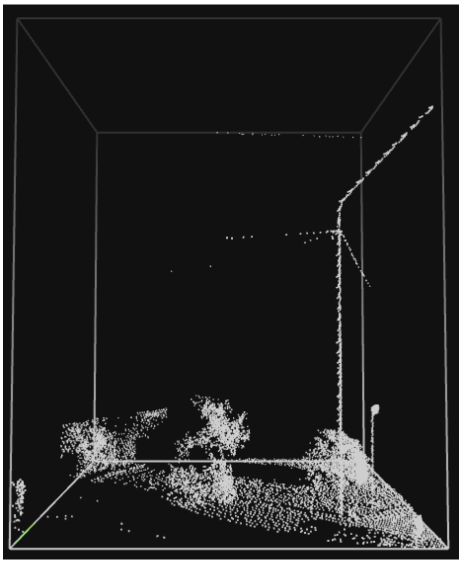
	Identified AusRAP attributes		
No in Table 2-1	**DVR Data**		**MLS Data**
1	Area type	Buildings	Pole
2	Speed limit	60, 100, 110	Tree
3	Median type	Line, Metal barrier, Concrete barrier, Median concrete, Grass, Flexipost	Ground
4	Centreline rumble strips		Metal barrier
5	Roadside severity - driver-side distance	Road	Sign
6	Roadside severity - driver-side object	Pole, Metal barrier, Concrete barrier, Median concrete, Tree, Grass, Guide post	
7	Roadside severity - passenger-side distance		
8	Roadside severity - passenger-side object	Pole, Metal barrier, Concrete barrier, Median concrete, Grass, Guide post	
9	Shoulder rumble strips		
12	Intersection type	Roundabout sign, Merge lane sign, Signal sign, Railway sign	
15	Quality of curve	Curvature sign	
16	Road condition	Defect	
17	Delineation	Warning sign, Guide post, Line	
18	Street lighting		
19	Pedestrian crossing facilities - inspected road	Pedestrian Crossing, Pedestrian sign	
22	Speed management / traffic calming	Speed hump, Speed hump sign	
24	Facilities for bicycles	Bicycle lane	
25	Roadworks	Roadwork sign	
26	School zone warning	School zone sign	

4 EXPERIMENTS AND RESULTS

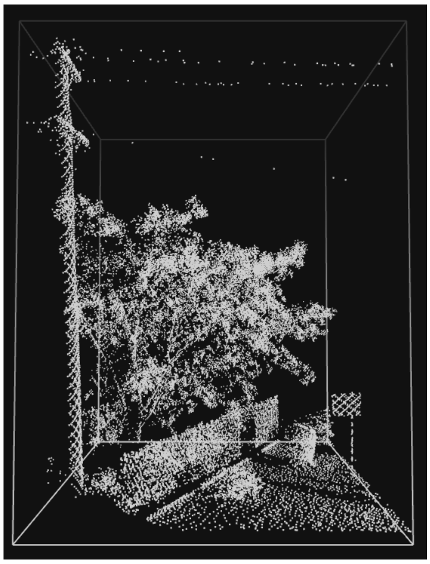
4.1 CREATION OF TRAINING AND TESTING DATASETS FROM MLS DATA

We have created datasets for training and testing purposes. Approximately, 150 grids that were extracted from LAS files in different locations with different appearances were manually annotated/labelled, and then the annotated grids of point cloud data were divided into two groups for training and testing. Some examples of the annotation for preparing training dataset are shown in Figure 4-1.

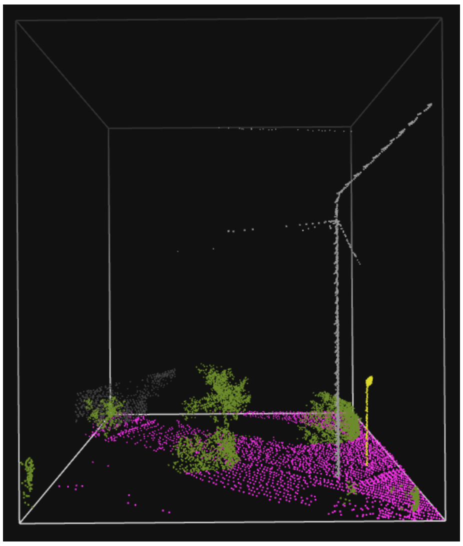
Figure 4-1. Annotation for preparing training dataset



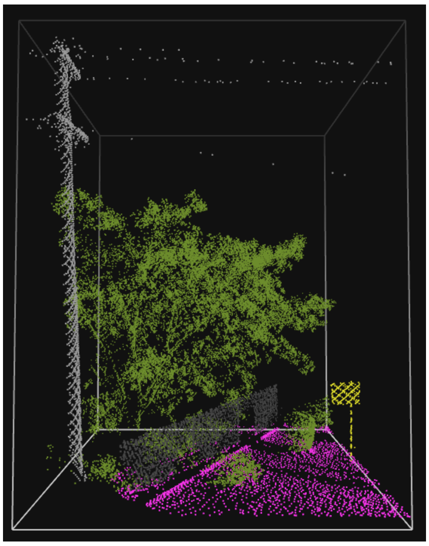
(a) Original point cloud of sample grid 1



(b) Original point cloud of sample grid 2



(a) Ground Truth of sample grid 1

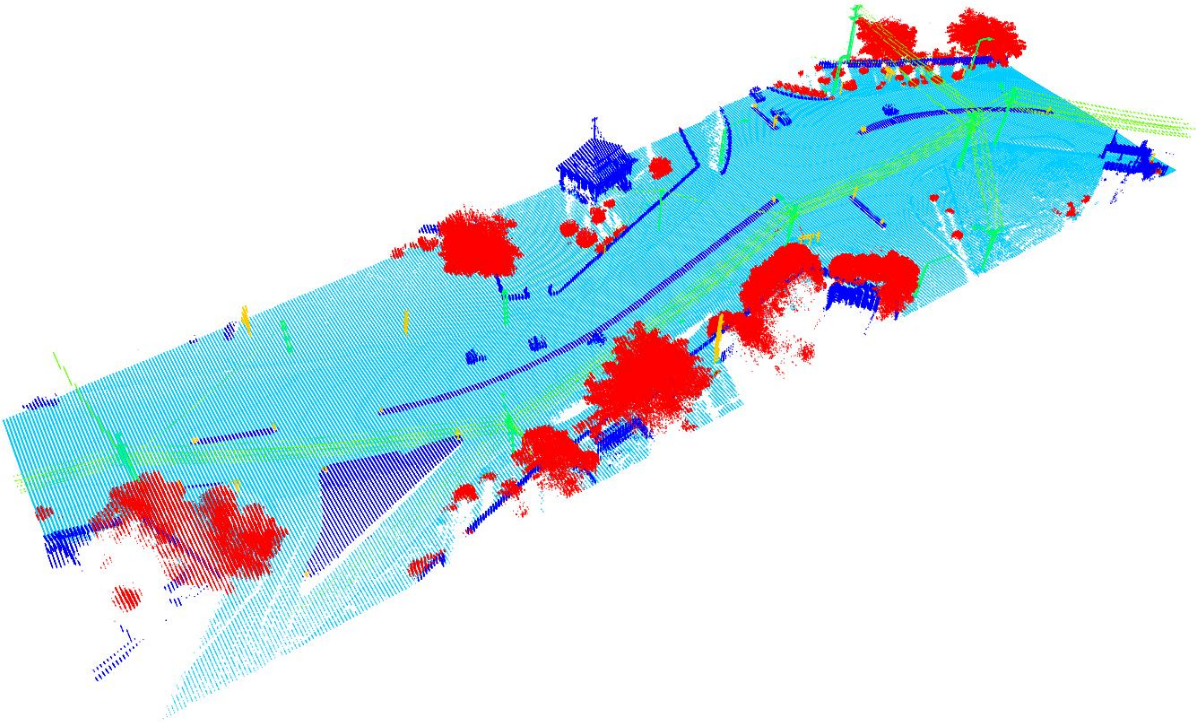


(b) Ground Truth of sample grid 2

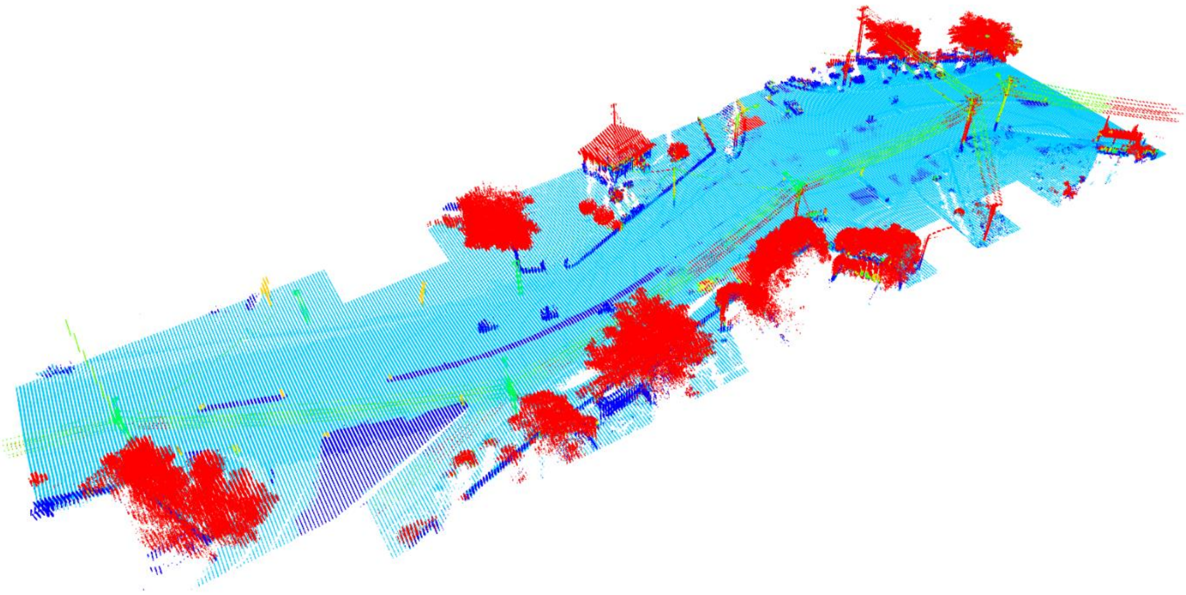
4.2 EXPERIMENTS AND RESULTS ON MLS DATA

We have implemented the framework shown in Figure 3-1 and conducted experiments with the attributes listed in Table 3-1. Figure 4-2 shows the test results of an example road section of road 10L.

Figure 4-2. Test results of an example road section – MLS data



(a) Ground Truth of a sample road section



(b) Recognition results of a sample road section

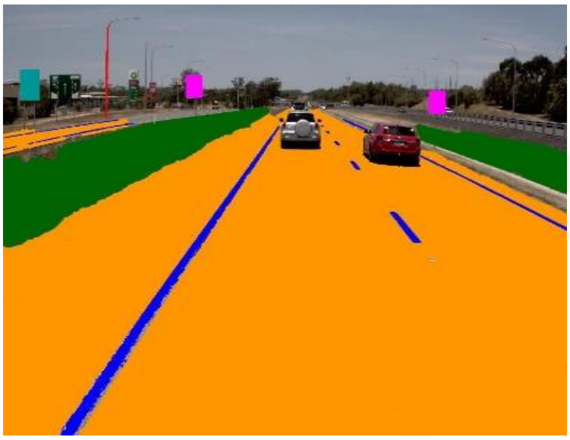
4.3 CREATION OF TRAINING AND TESTING DATASETS FROM DVR DATA

We have created large datasets containing many attributes and sub-attributes for training and testing purposes. Approximately, 400 images from different locations with different appearances were chosen and manually annotated/labelled, and then the annotated images were divided for training and testing. Some examples of the training and testing datasets are shown in Figure 4-3.

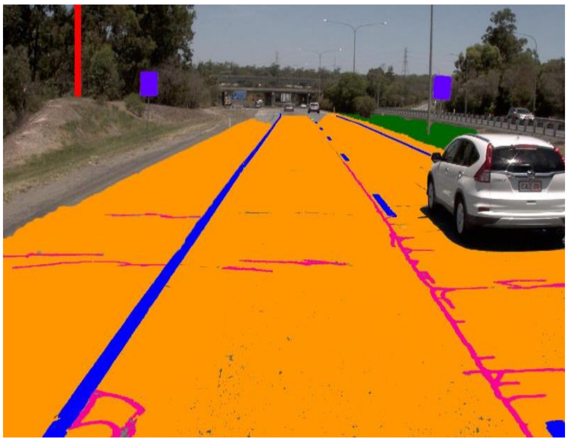
Figure 4-3. Annotation for preparing training dataset

(a) Original image of sample 1

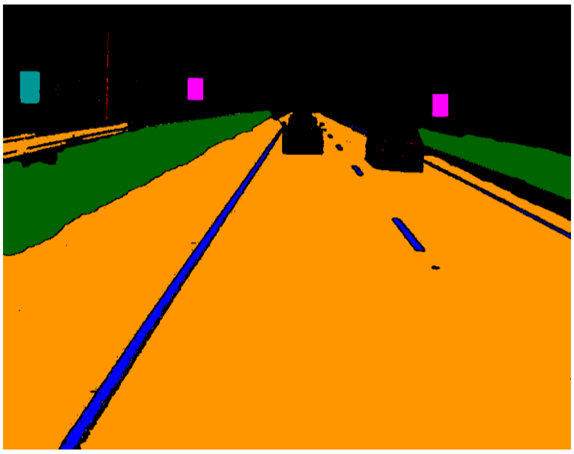
(b) Original image of sample 2



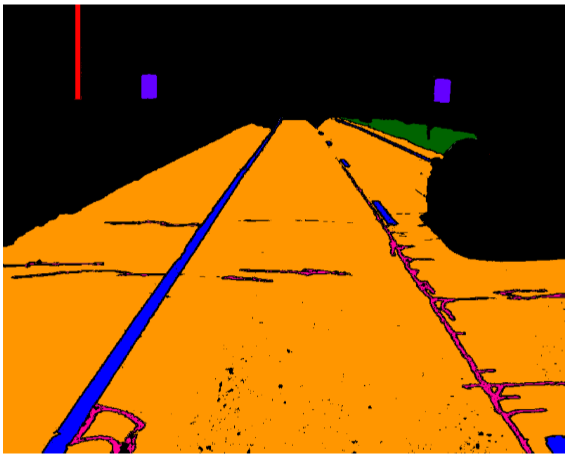
(a) Labelled image of sample 1



(b) Labelled image of sample 2



(a) Ground truth image of sample 1



(b) Ground truth image of sample 2

4.4 EXPERIMENTS AND RESULTS ON DVR DATA

We have implemented the framework shown in Figure 8 and conducted experiments with the attributes listed in Table 3. Figures 4.4 to 4.7 show the test results of some example attributes.

Figure 4-4. Attributes: road, line, metal barrier and tree (left: original image, right: recognition result)

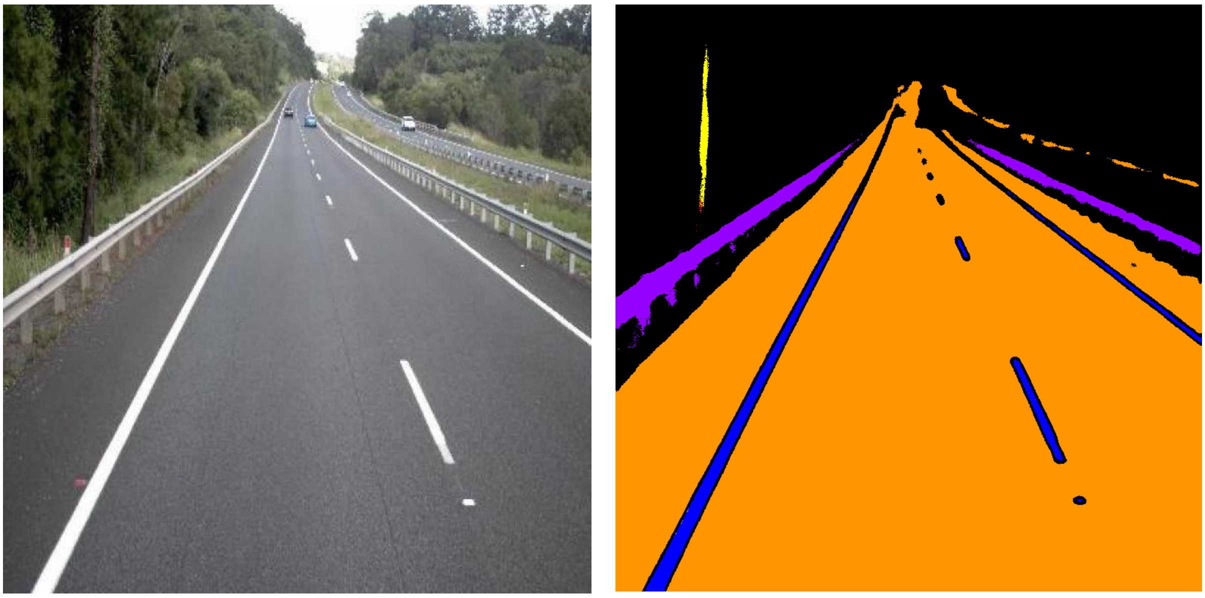


Figure 4-5. Attributes: road, line, metal barrier, 100 km/h speed sign and pole (left: original image, right: recognition result)

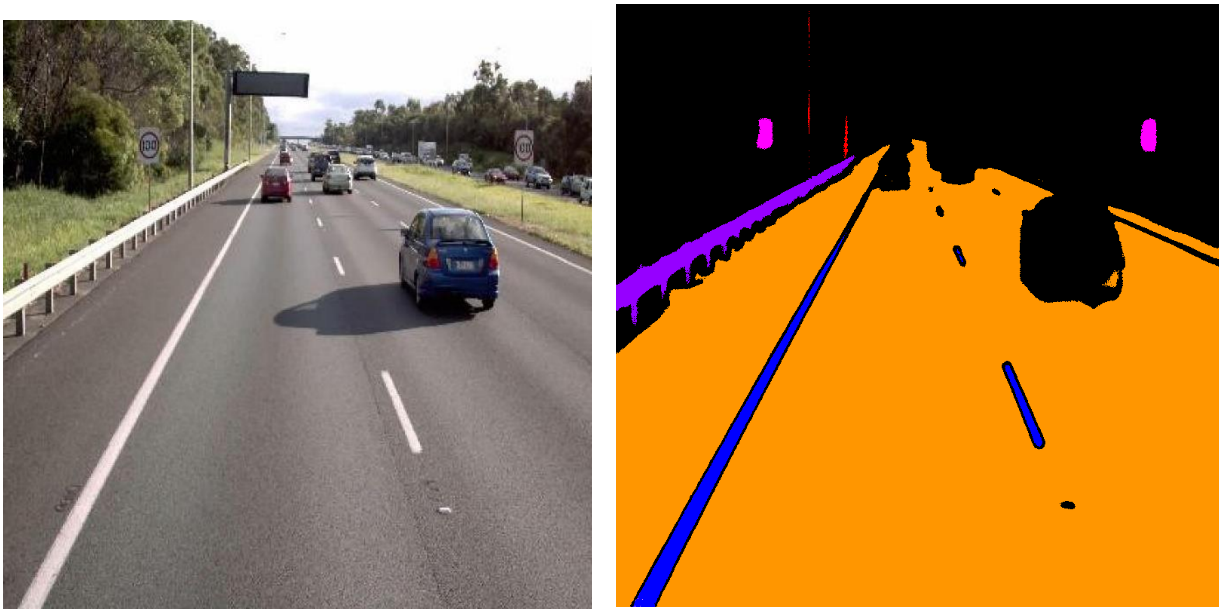


Figure 4-6. Attributes: road, line, line rumble strips, defect and grass (left: original image, right: recognition result)

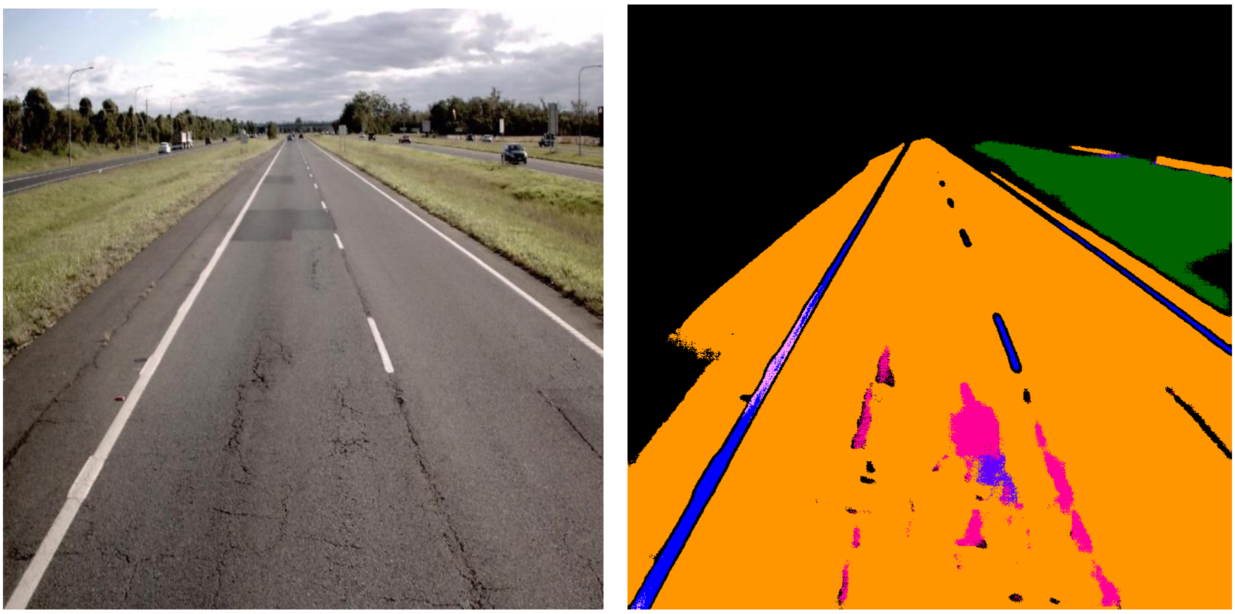


Figure 4-7. Attributes: road, line, bicycle lane and pole (left: original image, right: recognition result)

5 DISCUSSION/FUTURE DEVELOPMENTS

A system based on 3-D segmentation and classification method has been developed for identifying AusRAP attributes using MLS data, and currently the 3-D system is able to identify 6 AusRAP attributes. The preliminary system based on 2-D segmentation and classification method has been extended and now it is able to identify 29 AusRAP attributes/sub-attributes using DVR data. In general, both systems for MLS and DVR have achieved reasonable performance on identifying AusRAP attributes. As the number of training samples for some attributes is still small, the corresponding performance on identifying these attributes is limited. However, with the increasing number of training samples, the accuracy is expected to improve. The distance calculation techniques have been developed for both MLS data and DVR data. The distance calculation using MLS data is more accurate as the computation is based on actual geometric locations, compared with DVR data. The future work will focus on following using DVR data only due to limitation of MLS data as listed in Section 5.1.

- Revisit and optimise the framework for all AusRAP attributes.
- Develop algorithms/criteria to measure performance and comparative result analysis.
- Develop software system prototype.
- Identify AusRAP attributes over state-wide networks.

5.1 ISSUES WITH MLS DATA

Although distance calculation is more accurate on MLS data than DVR data, there are many issues with MLS data. We list below some major issues:

- Annotation is very difficult and time-consuming. Annotation on point cloud data requires assigning a class label to every point in the cloud point data. To develop a reasonable training data, we need to annotate millions of points.
- A limited number of attributes can be recognized. As the current MLS data only contains the geographical coordinates and intensity of each point, some attributes cannot be identified. For example, the characters on the signs are not visible using MLS/LiDAR data only with intensity.
- MLS data is available for limited number of roads, whilst DVR data is collected every year for all state-controlled roads.
- Visualization of LAS files is also very time-consuming. For example, visualizing a LAS file with around 1GB requires 15-30 minutes on normal desktops.

The above issues limit the use of MLS data as a viable data source for automating AusRAP data as required now. Hence, subsequent development of the software will be based solely on the use of DVR data.