Multi-stage Deep Learning Technique with a Cascaded Classifier for Turn Lanes Recognition

Pubudu Sanjeewani
Centre for Intelligent Systems
Central Queensland University
Brisbane, Australia
p.thihagodagamage@cqu.edu.au

Brijesh Verma
Centre for Intelligent Systems
Central Queensland University
Brisbane, Australia
b.verma@cqu.edu.au

Joseph Affum
Transport Safety Australian Road Research Board (ARRB)
Brisbane, Australia
joseph.affum@arrb.com.au

***Abstract*— The accurate recognition of road markings such as lanes and turn arrows is required in many applications including autonomous vehicles. Nevertheless, studies on road markings detection are commonly found in literature, detection and classification of turn lane arrows has not gained much attention. Most of the research which exists on the detection and classification of turn lane arrows have many limitations including low accuracy. Therefore, a novel technique based on two novel concepts for improving the performance of the detection and classification of turn lane arrows is proposed in this paper. Firstly, pixel-wise segmentation of all turn lane arrows into one class instead of each turn lane arrow in a separate class is proposed. Secondly, a novel cascaded classifier that evolves its weights so that it can identify turn lane arrows is proposed. Three turn lane road markings named left turn lane, right turn lane and Continuous Central Turning Lane (CCTL) are evaluated using a real-world roadside image dataset created by video data including all state roads in Queensland provided by our industry partners. The comparative analysis of the experimental results demonstrated outstanding results in terms of accuracy.**

Keywords—*deep learning, turn lane arrow recognition, fully convolutional neural network, cascaded classifier*

I. INTRODUCTION

Recognizing road markings automatically is still an unsolved problem in Advanced Driver Assistance System (ADAS). Traffic signs detected in Traffic Sign Recognition (TSR) related works are mostly placed alongside the road or above the road, and traffic signs lying on surface of the road still lacks investigation. Identifying traffic signs on road surface is challenging when compared with those alongside or above roads. The more severe reasons for difficult detection and classification include variations in scale and viewpoint, brightness conditions, blur caused by motion, occlusions and fading of colors. Traffic signs marked on road surfaces tend to get damaged unlike traffic signs alongside or above road. In example, turn lane arrows, numbers, and messages in word on road surface are subject to damage more than traffic signs due to the paint used to draw the marking which gets eroded by vehicular traffic. The human brain is designed in a way that it is possible to detect and analyze this information to process the response promptly with the desired series of actions. However, there is very limited research that has been conducted in doing this task automatically. Road markings on road surface have the prime purpose of sending an alert to the driver about dangers that may occur, guide the driver to avoid potential obstacles, reminding rules, or directions. If the road markings recognition and reaction process is included in ADAS, there should be a proper way of automatically identifying all road markings visually and sending timely responses to road markings. The response could be either a command for the driver or have authority over the car unilaterally to proceed with desired action. Vehicles on road always use to change lanes. Therefore, it happens to recalculate the drivers' navigation systems based on this information again and again. Therefore, integrating road markings detection to ADAS is of high importance. With regard to other attributes/applications concerned in ADAS, recognition of road markings has been less explored by researchers.

The information provided by proper integration of road markings recognition in ADAS will provide immense support for drivers to avoid many hazards. Even though lane markings detection is commonly seen in previous research [1], other types of road markings are not considered much. These road markings include, arrows, crosswalks, zebras, words, pictograms, continuous and discontinuous lane markings. Among these, arrows typically include several categories: (a) Simple Left Arrow, (b) Simple Right Arrow, (c) CCTL, (d) Left/Right Arrow, (e) Forward/Left Arrow, (f) Forward/Right Arrow, (g) Forward/Left/Right Arrow and (h) Simple Forward Arrow as shown in Fig. 1.

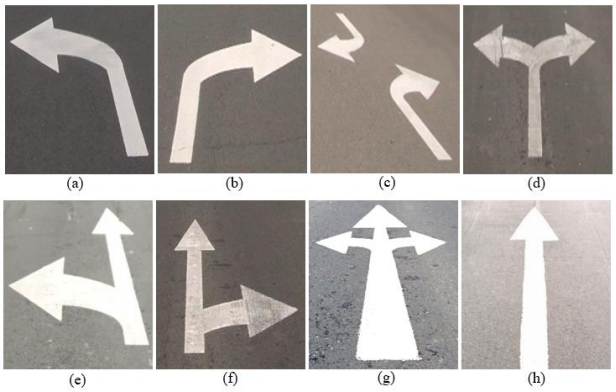

Fig. 1. Different types of arrow markings on road.

There are several reasons for less research conducted on other road markings except lane lines. Once such reason is, road lanes contain only lines, but other road markings have different features and color variations in different datasets. Also, they are subject to differences in length, thickness, or angle of view in consecutive frames when a vehicle is moving. In example, the variation of size of the arrow with the motion of the vehicle forward or away from the arrow can be mentioned. When the vehicle drifts horizontally on the lane, the orientation of the arrow is subject to change in a way that the same arrow will then appear as a straight line of different lengths and relative orientations. For instance, turn arrows such as Forward/Left, Forward/Right, and Forward/Left/Right can be misidentified as the forward arrow in case the left or right part of those arrows are occluded by shadows from nearby objects or paint quality. Moreover, road markings are always misclassified with other similar white color objects, low contrast in shadows, dust and erasing, occlusion by vehicles and other obstacles.

Turn lane arrow road markings are subject to erosion and are damaged by the contact of number of automobiles which is increasing day by day. Therefore, on such occasions, correctly identifying road markings becomes difficult for a driver. A driver who fails in identifying road markings correctly will ultimately results in severe traffic accidents or injury to pedestrian. In real world, there are some situations where navigation systems fail due to complex road conditions such as traffic control, road construction, and unknown private property. Therefore, in order to deal with such unexpected catastrophe, a real-time solution to detect and recognize road markings automatically with high accuracy is needed. Hence, the research in this paper is focused on improving the performance of recognizing three types of turn lane arrows: left (Fig. 1 - (a)), right (Fig. 1 - (b)) and CCTL (Fig. 1 - (c)) which belong to Australian Road Assessment Program (AusRAP) attributes.

We explore a novel concept of cascaded classifier learning and propose a novel technique to accurately detect and classify road surface-based turn lane markings. The main characteristics of our proposed technique include:

1. A Fully Convolutional Network (FCN) with all turn lane arrows into one class to segment turn lane arrow pixels into Region of Interests (RoIs) for accurately identifying turn lane arrows in our AusRAP safety attribute detection system.
2. A novel evolving cascaded classifier that classifies detected pixels into individual turn lane arrows or non-arrows.
3. An evaluation of the proposed technique on a real world industry dataset conducting large number of experiments and comparative analysis.

This paper is organized into seven sections. Section II reports a relevant literature on road markings detection. Section III outlines a description of the new technique with a cascaded classifier. Experimental results are given in Section IV, followed by Section V with result analysis and discussion. Section VI outlines the comparative analysis. Finally, the conclusion and future work are provided in Section VII.

II. Related Work

Even though, a considerable number of research has been conducted in recognizing lane markings using onboard cameras, a far lesser number of literatures exist on recognizing arrows in real world complex road environment. When observing literature, the techniques proposed on detecting arrows marked on road surface using on-board camera images includes techniques like Inverse Perspective Mapping (IPM) [2], [3], [4] of input images, using connected components [4], [5], applying wavelets [2] and matching the extracted features using curve/spline fitting models [5] and geometrical pattern matching [3]. For instance, Wang et al. [2] extracted arrows using IPM and the extracted features were matched using an improved Haar wavelet feature extraction approach recognizing arrow markings effectively while producing promising results in occlusion caused by traffic and in imperfect visual conditions. Chira et al. [3] developed a LabVIEW based system to identify five most common road markings using IPM and geometrical pattern matching. The system could calibrate monocular video sources, define the features of the painted objects, and detect them in real time using geometric pattern matching and edge detection. Foucher et al. [4] detected marking pixels using connected components before IPM and recognized repetitive markings such as crosswalks and single patterns such as arrows based on the comparison with a single pattern or with repetitive rectangular patterns. The results demonstrated that 90% and 78% crosswalks and arrows were detected respectively on a real-world database of high-definition images with a considerable number of difficult cases. However, the authors pointed out the necessity for further investigations leading to optimization of the recognition algorithm.

Maier et al. [5] extracted arrows using connected components and analyzed using curve-based prototype fitting models like splines. Classification of arrows was done using measures like Hausdorff distance. Tang et al. [6] introduced a novel rule-based descriptor and a cascade classifier to recognize road markings. Real life data collected in Ottawa city were used to evaluate the proposed technique and achieved remarkable results. At the same time, Tang et al. [7] also analyzed road markings detection using Histogram of Oriented Gradient (HOG) with SVM on a local database and achieved speed and accuracy improvements over the rule-based detection method. Wu et al. [8] used road marking-based perspective transformation to estimate vehicular speed and obtained an average error of 3.2 kmh^{-1} and an average relative error of 9.94%.

Zhang et al. [9] used deep learning to detect six common types of road surface identifiers including five turn lane arrows, and the crosswalk. R-FCN model with 18-layer depth residual network was identified as the best model for road marking detection based on comparative analysis of results. Qian et al. [10] detected road surface traffic sign with hybrid region proposal and fast R-CNN and demonstrated an overall average precision of about 85.58% on a field-captured dataset with high recall and precision rate. Suchitra et al. [11] proposed a novel method to robustly identify arrow markings on road images using signed edge signatures. The method was validated on test image sequences containing more than 1,100 real images and obtained 97% classification accuracy. However, the analysis of misdetections was left for future scope of work. Vokhidov et al. [12] recognized six types of arrow-road markings, possibly damaged by visible light camera sensor based on Convolutional Neural Network (CNN). Experimental results with six databases of Road marking dataset, KITTI dataset, Málaga dataset 2009, Málaga urban dataset, Naver street view dataset, and Road/Lane detection evaluation 2013 dataset outperformed conventional methods. However, the detection speed was too slow and was not suitable for practical applications.

Hoang et al. [13] used adaptive region of interest and deep learning to detect and recognize road markings. Three open datasets namely, the Cambridge dataset, Daimler dataset, and Malaga urban dataset were used to evaluate the proposed technique. The proposed technique outperformed the state-of-the-art methods in recognition performance. Smaller road markings at a large distance were also identified. However, the proposed technique required improvements over the processing speed to detect vanishing point quickly. Li et al. [14] proposed road markings extraction based on threshold segmentation on urban streets acquired by a camera mounted on a moving vehicle. He et al. [15] used edit distance and junction feature to detect and recognize arrow road markings and obtained better experimental results on datasets collected in various road situations. However, it was recommended that

damaged or indistinguishable road markings detection still requires further research to obtain better results in terms of accuracy. Therefore, a more comprehensive technique that can support very small response times and a high sensitivity to the field of vision is required.

When observing the existing research work in literature, it can be seen that the existing research on recognizing road markings achieved low performance. Accuracy and performance in terms of detection, classification, anti-interference, real-time, universality and robustness of proposed techniques are very low. It is necessary to develop novel concepts and techniques to improve road marking algorithms.

III. Proposed Technique

The proposed technique consists of two steps as detection step and classification step. In detection step, an FCN network is designed to output regions with all turn lane arrows, when an input frame is fed to the system. Some parts in the test image could be detected as one of the 3 turn-lane arrows even though they do not belong to any turn-lane arrow. In this way, an RoI can be detected as a turn-lane arrow, but if it does not contain any turn lane arrow then it is called a false alarm. The first classifier in the cascaded classifier is designed to eliminate noise or false alarms from test images. Therefore, the effect of false alarms is minimized in the proposed classifier which can classify an input image as one of the three turn lane arrows or a non-arrow which is a noise/false alarm. The second classifier in the cascaded classifier, classifies RoIs into one of the three turn lane arrows. In the final prediction, the color of an RoI classified as a turn lane arrow is replaced with a color of the recognized turn lane arrow (left, right, CCTL). Each attribute in the system has assigned a different color value. An overview of the full proposed technique is illustrated in Fig. 2 below.

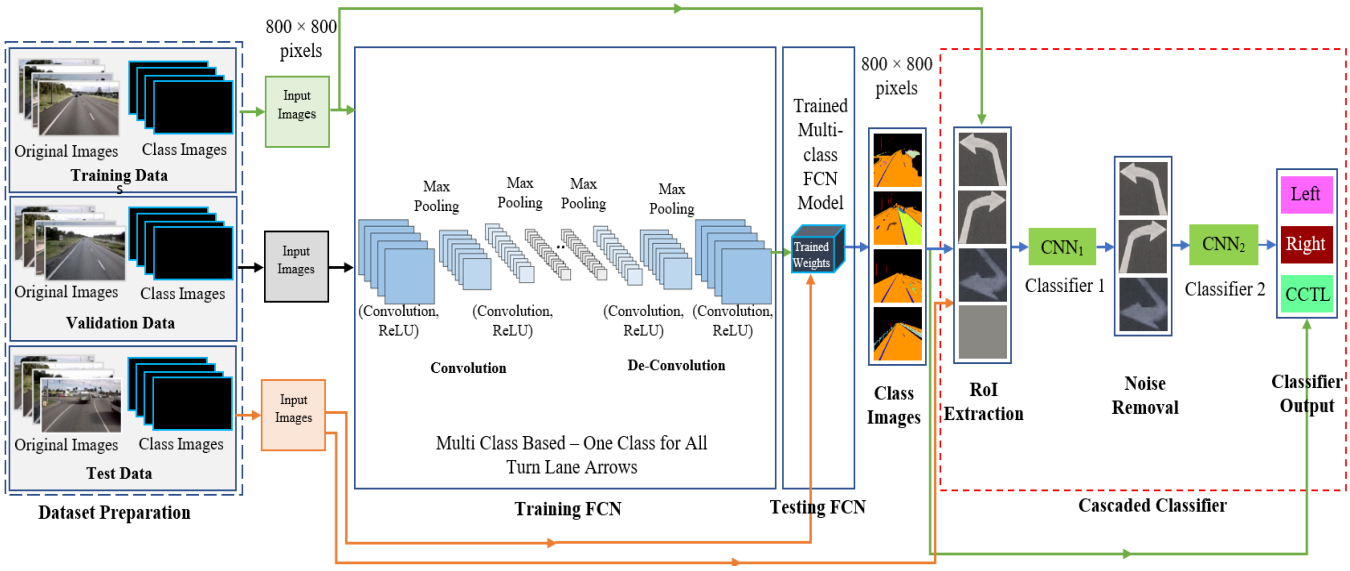

Fig. 2. Proposed technique.

Each part of the proposed technique shown in Fig. 2 is described below in detail.

A. Dataset Preparation

The input to the proposed technique mainly consists of training, validation, and test frames. Accordingly, three datasets for training, validation, and testing were prepared by the extracting images from Digital Video Recordings (DVRs) provided by our industry partners, the Department of Transport and Main Roads (DTMR), Queensland and Australian Road Research Board (ARRB). The input frames in these three datasets cover images from all state roads in Queensland, Australia. Before input to the system, all frames were resized to 800 × 800 pixels as this image size was identified as the optimal image size for the proposed technique in our previous research.

Altogether, 61 attributes are identified for our final system including 3 turn lane arrows. However, improving the performance of turn lane arrows which is a difficult attribute to recognize is the focus of this paper. These attributes include various types of road safety attributes such as, road, defects on road, traffic signs, road markings, many roadside objects such as trees, poles, different types of barriers including wire rope barrier, motor-cycle friendly barrier and metal barrier, different types of medians such as median concrete, median grass, etc., upwards slopes, boulders, segregated bicycle path barriers, guideposts, flexiposts, Vulnerable Road Users (VRUs) such as pedestrians, bicycles, motorcycles, etc. Road markings identified in this research include, pedestrian crossing, dedicated bicycle path, rumble strip, lane line, central hatching, and all turn lane arrows including left turn lane, right turn lane and CCTL.

All 61 attributes were assigned 61 RGB color values and annotated using Adobe Photoshop CC 2019 software. Each RGB color was assigned to an attribute only after validating through a MATLAB program which compares the selected color with the colors in the images of a sample dataset prepared by including frames from input video data. A class was assigned for each attribute. Information other than all the 61 attributes in an image were considered as the background and put into a separate class called “unlabeled”. Therefore, the final system had 62 classes with the unlabeled class. Fig. 3 below illustrates an original input frame for each turn lane arrow along with its corresponding annotation sample.

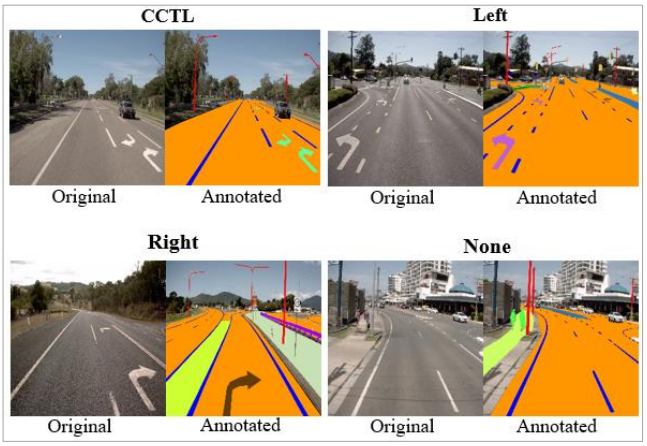

Fig. 3. Turn sign frames and their corresponding annotations.

B. *FCN for Pixel-wise Detection*

As shown in Fig. 1, initially, all turn lane arrows into one class are detected using an FCN architecture which takes input images of size 800 × 800 pixels and produces output images of same size based on imagenet-vgg-verydeep-19 model. The FCN architecture is comprised of number of convolutional and pooling layers, Rectified Linear Unit (ReLU) activation function, Adam optimizer and backpropagation algorithm. FCN8 architecture is used among many variants of FCN due to its outstanding performance over many years.

In our previous work, FCN network parameters in architecture were optimized. The experiments with optimized parameters such as number of iterations, number of layers, image size, pooling type, activation type, backpropagation algorithm type, etc. achieved optimal accuracy. By setting optimal parameters selected by the optimizer, FCN was trained with one class for all turn lane arrow signs and separate classes for other attributes. The number of classes were reduced to 60 from 62 with the use of one class for turn lane arrows. FCN was configured to automatically save trained models and prediction results with increments of 5,000 iterations until 160, 000 iterations.

After evaluating all the 32 trained models on test dataset, the best model was stored and used in our technique. The best model was identified based on the classification accuracy (both attribute wise and pixel wise) as well as the number of misclassifications. In testing stage, RoIs where the turn lane arrows are placed are saved from prediction results obtained by the best trained model. After that, RoIs are interpreted by the cascaded classifiers and produce final prediction results combining results of all the 61 attributes.

Prediction images produced by the FCN are black color class images which contain class number from 0 to 59. Based on the class number used for turn lane arrows, an automatic Python program was written to extract RoIs from corresponding original images. In this way, black color class images and original resized images are used to extract the locations of turn lane arrows detected by the FCN based detector. At this stage, all regions detected as having turn lane arrows are extracted along with misclassified turn lane arrows (false alarms) from training and test images. RoIs are extracted from training images to prepare a training dataset to train the cascaded classifier. These RoIs were saved and passed to our cascaded classifier system with turn lane arrow and false alarm classes which have been evolved with turn lane arrow and non-turn lane arrow inputs. Section C presents a comprehensive description of the introduced cascaded classifier. Some sample RoIs - input to the cascaded classifier are shown in Fig. 4 below.

As observed in Fig. 1 and Fig. 3, the CCTL is a combination of two signs including a right turn lane arrow. Therefore, the output RoIs from FCN contain only images as shown in Fig. 4. The right turn images in test images could be a part of CCTL as well. Therefore, for CCTL, classifiers are trained using only one part of the CCTL sign (except right turn lane sign) and right turn sign is trained with right turn lane sign class. In the final prediction, if both CCTL signs are detected on a single frame, those two signs are detected as a CCTL.

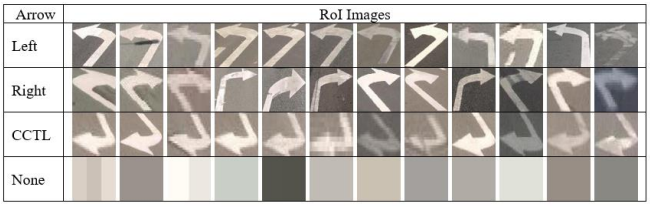

Fig. 4. RoI images as input to cacased classifiers.

C. *Cascaded Classifier*

Two novel cascaded classifiers based on CNN are introduced in this research. CNN architecture was chosen for classification because of its outstanding performance in road attributes detection tasks and its ability to autonomously extract high-quality features. The two classifiers are trained in two different ways. Initially, a CNN classifier was trained for 4 classes: left, right, CCTL with a false alarm class. But it was observed that the test results have many false detections as turn lane arrows leading to many misclassifications in the output. The RoI images on which the classifier was trained on contained many false images/noises which do not contain any turn lane arrow. Therefore, it was decided to remove noise or false detections from test images before feeding test images to the classifier. For this purpose, another classifier (classifier 1) was introduced. Classifier 1 was trained for two classes. One class is for false detections and the other class is for turn lane arrows. Classifier 1, which is a binary classifier takes RoI images resized to 180 × 180 pixels as input and autonomously extracts a set of features and feature values for each class within the dataset and classifies it as a false detection (class 0) or a turn lane arrow (class 1). Classifier 1 had 185 turn lane arrow images and 23 false detection images as input. 80% of the input images were used for training while 20% were used for validation. In this way, noise images (class 0) were first removed from the test images and turn lane arrows were filtered from test images. Now, the test images do not have false detections in the test images and only turn lanes arrows are present. Therefore, the second classifier is trained only for 3 classes: left, right and CCTL. The second classifier in the proposed architecture takes left, right and CCTL images as input, autonomously extracts a set of features and feature values for each class within the dataset and classifies it as left, right or CCTL. Classifier 2 had 65 left turn arrow images, 95 right turn arrow images and 25 CCTL images as input. 80% of the input images were used for training while 20% were used for validation. The architecture of the two classifiers in the cascaded system are same even though they are trained for different number of classes. The architecture of the second classifier is described below.

The data input to the proposed architecture is first passed to the feature learning component of the system where the

input data is decomposed into different levels of features. The feature learning component of the proposed architecture consists of three convolution blocks with a maximum pooling layer in each of them followed by a flatten layer. On top of it is a ReLU activation function activated fully connected layer with 128 units. A different set of features are extracted by each convolutional layer from the input image data. In order to do this, convolutional operation is performed over its input data from previous layer using 3 × 3 kernals. The reason for selecting ReLU activation function for each convolutional layer is to remove feature values with negative values which are not important and replace them with 0. Unlike other activation functions such as TanH, the ReLU activation function solves the vanishing gradient problem. This leads to a quicker learning and a better performance and that is the main benefit of having ReLU as the activation function in this task. The maximum pooling layer reduces the resolution of learned features by determining the maximum values which contain the most important information in regions of the convolutional input. Computational costs will be lowered by implementing a maximum pooling layer eventually resulting in better results. Dimensionality of the discovered features from the previous convolutional layer is lowered using kernels of size 2 × 2 in maximum pooling layer before input to the next convolutional layer. The flatten layer present after the last maximum pooling layer flattens the identified discovered features into more comprehensive features in one-dimensional array or a feature vector. First, the model has to be trained for feature extraction and for this purpose, a training dataset is presented to the network. During the training process, the network learns different features from the convolutional component. Then, the discovered features are flattened and given to the classification component of the network. The classification component comprises of two fully connected layers used to determine whether the features given represent a left, right or CCTL arrow.

CNN training process is as follows: Initially, the parameters of the network are arbitrarily predetermined with arbitrary values. For example, the number of epochs was set to 15 in this case. Then, each input image from the training dataset is passed through the model and the model seek to deliver a prediction outcome (left, right or CCTL) which contains class label (0, 1, 2). Then, the predicted class label (0, 1, 2) and actual class label (0, 1, 2) of each image are compared to evaluate a loss function. By backpropagating through the network, parameters are updated based on the results of the loss function using an adaptive optimization algorithm named Adam. When a better loss value is produced in an iteration, the model parameters are saved before moving to the next epoch. For feature extraction, a truncated version of the learned model which contains all the trained layers of the CNN except the classification component is extracted after the model is trained. Training and test datasets are provided to this model to extract features where each image is passed to the extracted model at a time. Eventually, a set of extracted feature values for each input are produced by the model and stored in a file for later analysis. Using the stored values, a plot diagram of each extracted feature vector is plotted and stored. The architecture of the second classifier in the cascaded system is shown in Fig. 5 and the training and feature extraction procedure is described in Fig. 6. Table I below presents the parameters used in the CNN classifier.

D. Classification and Visual Interpretation

As the next step, the classification is performed to test the model after the extraction process is over. In classification step, a prediction based on the highest confidence value of the extracted features is produced. As the first step in the classification process, a test dataset is fed to the trained CNN. The test dataset includes images which the classifier has not seen before. Features are autonomously extracted from the input data and a prediction is produced for each input image as left, right or CCTL by the trained CCTL. The trained model is evaluated using the predicted class values and the actual class value using commonly used metrics (attribute wise accuracy, pixel wise accuracy). Furthermore, visual interpretation of final prediction results obtained by turn-lane arrow classification are discussed in Section IV below.

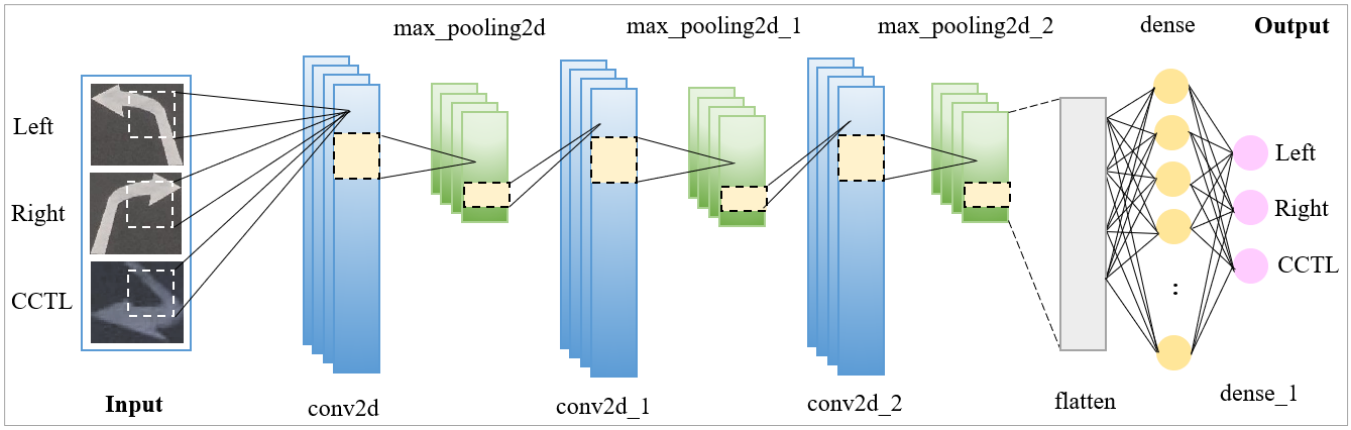

Fig. 5. Architecture of the second classifier in the proposed cascaded classifier.

```
Algorithm 1: Feature Extraction Procedure
Input: Training dataset, testing dataset
Output: Feature vectors, feature plots
Begin
    Step 1: Train the CNN using training data
    Step 2: Extract the learned model without the fully connected layer
        Clone the trained CNN model
        Remove the fully connected layer (dense layer)
    Step 3: Extract and plot features from the training and testing datasets
        foreach dataset do
            foreach image in dataset do
                Process the image through the extracted model
                Append the output feature vector to a file
                Plot each feature vector
End
```

Fig. 6. Training and feature extraction procedure.

TABLE I. ARCHITECTURAL PARAMETERS FOR THE PROPOSED CLASSIFIERS

Layer	Activation Function	Padding	Kernal Size	Output Size	Parameters
rescaling_1 (Rescaling)	-	-	-	(None, 180, 180, 3)	0
conv2d (Conv2D)	ReLU	Same	3	(None, 180, 180, 16)	448
max_pooling2d (MaxPooling2D)	-	-	2	(None, 90, 90, 16)	0
conv2d_1 (Conv2D)	ReLU	Same	3	(None, 90, 90, 32)	4,640
max_pooling2d_1 (MaxPooling2D)	-	-	2	(None, 45, 45, 32)	0
conv2d_2 (Conv2D)	ReLU	Same	3	(None, 45, 45, 64)	18,496
max_pooling2d_2 (MaxPooling2D)	-	-	2	(None, 22, 22, 64)	0
flatten (Flatten)	-	-	-	(None, 30976)	0
dense (Dense)	ReLU	-	-	(None, 128)	3,965,056
dense_1 (Dense)	-	-	-	(None, 5)	645

E. Process for Cascaded Classifier Training

Step 1: Prepare (pre-processing and resizing) RoIs used for training and testing the first classifier with two classes (arrow, noise).
Step 2: Load the training (80% of the training data), validation (20% of the training data) and testing data.
Step 3: Define initial parameters for the first classifier model.
Step 4: Train the first classifier model.
Step 5: Visualize the results.
Step 6: Save the best model.
Step 7: Analyse the model on testing data and remove images with noise (false arrow detection) from the testing data.
Step 8: Prepare (pre-processing and resizing) RoIs used for training and testing the second classifier with three classes (left turn lane, right turn lane and CCTL).
Step 9: Train the second classifier model.
Step 10: Visualize the results.
Step 11: Save the best model.
Step 12: Evaluate the second classifier model with noise removed test data.

IV. EXPERIMENTS AND RESULTS

To evaluate the proposed technique, we implemented it in Python using TensorFlow and Anaconda environment on our high-performance computing facility. Many experiments were conducted using the training and test data. The size of the training and test data are 784 training images and 332 test images respectively. The dataset was prepared by extracting videos provided by our industry partners and this custom dataset is a part of a large industry funded project on AusRAP attribute detection. Along with the original test images, annotated images are also included in test dataset in order to compare the prediction results with the actual Ground Truth (GT) data and obtain classification accuracy metrics. The experiments are divided into two parts as described below.

Initially, FCN network is evaluated for all the AusRAP attributes in separate classes. Therefore, FCN is trained for all the 61 AusRAP attributes (62 classes) first. Then, all the turn lane arrows are put into one class and assign one color for all the turn lane arrows and train FCN for 59 attributes (60 classes). The purpose of training FCN in this way is to detect all turn lanes arrows in one class. Because of the complexity in road environment, many turn lane arrows are misclassified with extreme similar attributes and fail to recognize correctly when trained separately using existing techniques. The proposed technique avoids this problem. Now the cascaded classifier is trained, and the detection results obtained by the second experiment are passed through the trained cascaded classifier for testing. Table II lists the classification accuracy obtained by the first part of the technique while evaluation results obtained by the second part of the technique which is our cascaded classifier, are listed in Table III.

In the first experiment, prediction results are evaluated both pixel wise (detection) and attributes wise (classification). Here, the pixel wise accuracy is the percentage of accurately classified pixels over the total number of pixels of that object in the test dataset. Attribute wise accuracy is calculated as the percentage of correctly classified objects over the total number of objects of that attribute in the test dataset. For attribute wise accuracy, objects' final classification results are based on the majority prediction of that object within the region which belongs to that attribute. Table III does not list

pixel wise accuracy. This is because, the cascaded classifier performs only the classification of detected RoIs in first part. Therefore, only classification accuracies are calculated and presented.

TABLE II. BEST ACCURACY OBTIANED BY THE FCN CLASSIFICATION TECHNIQUE

Turn Lane Arrow	Accuracy (%)	
	Attribute wise	***Pixel wise***
CCTL	71.06	82.00
Left	72.00	56.97
Right	56.59	48.00

TABLE III. BEST ACCURACY OBTIANED BY THE PROPOSED TECHNIQUE

Turn Lane Arrow	Accuracy (%)
CCTL	100.00
Left	90.38
Right	90.57

V. RESULT ANALYSIS AND DISCUSSION

The analysis of experimental results showed that the proposed techniques with our cascaded classifier evolved its weights very quickly to achieve a high degree of prediction accuracy. Figures below illustrate some of the sample prediction results obtained followed by an analysis of results. The best prediction results obtained for right turn lane, left turn lane and CCTL from existing technique based on multi-class turn lane arrow (Prediction 1) and proposed technique with one class for all turn lane arrows (Prediction 2) are illustrated in Fig. 7, Fig. 8, and Fig. 9 respectively. The corresponding original image and GT image are also presented in front of each prediction result to make the comparison easier. Accordingly, the prediction results obtained for turn lane arrows by the proposed cascaded classifier appear to have better predictions which look almost similar to the GT image actual result. Prediction results obtained by training all turn lane arrows in separate classes contain much noise and many misclassifications.

We have also conducted experiments with state-of-the-art technique called DeepLabV3+ and results are shown in Table IV. When observing the results listed in Table II-IV, the results obtained from the proposed technique show excellent performance on all turn lane arrows.

TABLE IV. BEST ACCURACY OBTIANED BY DEEPLABV3+ TECHNIQUE

Turn Lane Arrow	Accuracy (%)	
	Attribute wise	***Pixel wise***
CCTL	44.75	74.00
Left	52.00	50.00
Right	43.41	50.97

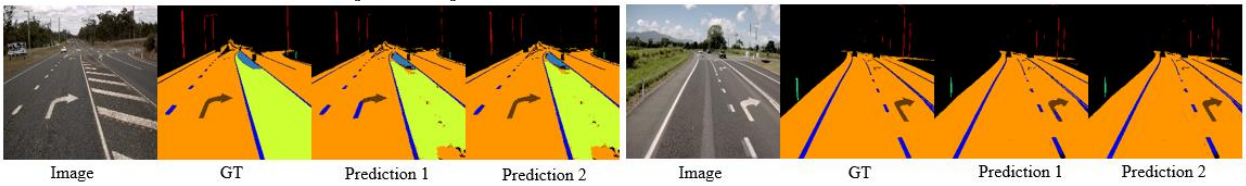

Fig. 7. Prediction results of right turn arrow obtained by multi-class turn lane arrow model (prediction 1) and the proposed technique (prediction 2).

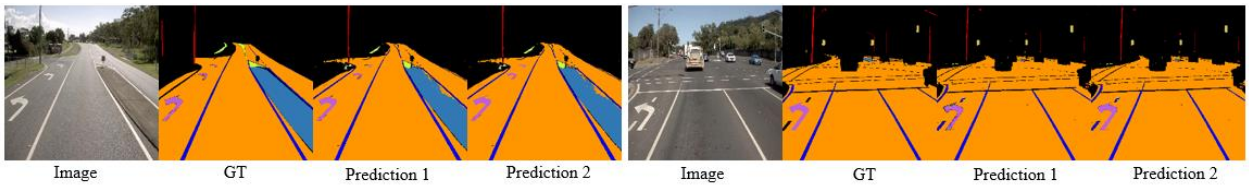

Fig. 8. Prediction results of left turn arrow obtained by multi-class turn lane arrow model (prediction 1) and the proposed technique (prediction 2).

Fig. 9. Prediction results of CCTL obtained by multi-class turn lane arrow model (prediction 1) and the proposed technique (prediction 2).

VI. COMPARATIVE ANALYSIS

A comparative analysis was performed to compare the results obtained by the proposed technique and some existing state-of-the-art techniques. The proposed cascaded classifier based deep learning technique demonstrated high accuracy and low misclassifications because it has the ability to eliminate the false alarms by classifying non-arrow RoIs into the false alarm class using the first classifier. Analysis and comparison of results has been conducted on the results obtained by the proposed technique. A comparison of attribute wise accuracy obtained by the proposed technique, with a false alarm class [16], DeepLabV3+ [17] and FCN8 [18] based majority pixel classifier is shown in Table V. Some of the sample prediction results obtained from pixels detected as turn lane arrows by DeepLabV3+, FCN8 and proposed technique are illustrated in Fig. 10 below. As observed below, predictions of DeepLabV3+ and FCN8 images have lots of noise and unclear predictions in all three cases of turn lane

arrows. Sample images which were obtained by the proposed technique show significant improvement.

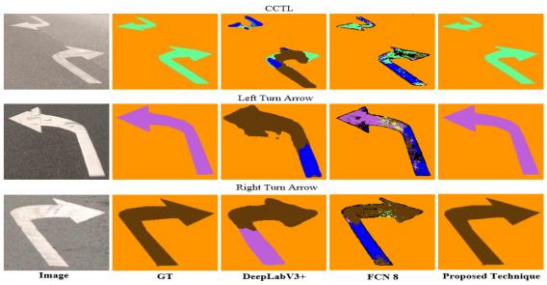

Fig. 10. Sample prediction results obtained by DeepLabV3+, FCN8 and the proposed technique.

TABLE V. COMPARISON OF CLASSIFICATION ACCURACY (%)

Turn Lane Arrow	Attribute Wise Accuracy (%)			
	Proposed technique	***GA Classifier [16]***	***DeepLabV3+ [17]***	***FCN8 [18]***
CCTL	**100.00**	100.00	**44.75**	71.06
Left	**90.38**	80.77	**52.00**	72.00
Right	**90.57**	81.13	**43.41**	56.59
Average	**93.65**	87.30	**46.72**	66.55

As can be seen, by the above results, our proposed technique outperformed all comparative techniques. The results indicate that our proposed technique can be used for turn lane arrows detection in our AusRAP attribute detection system. Overall, the proposed technique with our cascaded classifier has achieved an average accuracy of 93.65% while GA classifier-based technique has achieved an average accuracy of 87.30%.

VII. CONCLUSION AND FUTURE WORK

This paper presents a novel technique for improving the accuracy of turn lane markings detection and classification. The system was implemented, and performance of the proposed technique was validated using a real-world dataset from industry. Performance evaluation was performed with state-of-the-art techniques such as FCN8 and DeepLabV3+ with majority pixel-based classifier. When compared with state-of-the-art, the proposed technique produced promising results with the highest classification accuracy and low misclassifications. The overall accuracy achieved by the proposed technique for turn lane arrows is 93.65% which is much higher than the accuracy obtained by other existing techniques (GA classifier – 87.30%, FCN8 - 66.55%, DeepLabV3+ - 46.72%).

In our future work, road marking detection will be further extended by adding other road markings such as pedestrian crossing, central hatching, dedicated bicycle lanes, parking slots, rumble strip, etc., to address issues in these attributes and improve the accuracy. Attributes other than road markings similar in color, size and shape will be investigated by further improving the performance of the proposed technique.

ACKNOWLEDGMENT

This research was supported under Australian Research Council's Linkage Projects funding scheme (project number LP170101255).

REFERENCES

[1] F. Rakotondrajao and K. Jangsamsi, "Road boundary detection for straight lane lines using automatic inverse perspective mapping," in *Int. Symp. Intell. Signal Process. and Communication Syst. (ISPACS)*, 2019, pp. 1-2.

[2] N. Wang, W. Liu, C. Zhang, H. Yuan and J. Liu, "The detection and recognition of arrow markings recognition based on monocular vision," in *Chin. Control Decis. Conf.*, Guilin, China, 2009, pp. 4380-4386.

[3] I. M. Chira, A. Chibulcutean and R. G. Danescu, "Real-time detection of road markings for driving assistance applications," in *Int. Conf. Comp. Eng. & Syst.*, Cairo, Egypt, 2010, pp. 158-163.

[4] P. Foucher, Y. Sebsadji, J. Tarel, P. Charbonnier and P. Nicolle, "Detection and recognition of urban road markings using images," in *14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Washington, DC, USA, 2011, pp. 1747-1752.

[5] G. Maier, S. Pangerl and A. Schindler, "Real-time detection and classification of arrow markings using curve-based prototype fitting," in *IEEE Intell. Vehicles Symp. (IV)*, Baden-Baden, Germany, 2011, pp. 442-447.

[6] Z. Tang and A. Boukerche, "A novel video-based application for road markings detection and recognition," in *IEEE Int. Conf. Commun. (ICC)*, Paris, France, 2017, pp. 1–6.

[7] Z. Tang and A. Boukerche, "An improved algorithm for road markings detection with SVM and ROI restriction: comparison with a rule-based model," in *IEEE Int.Conf. Commun. (ICC)*, Kansas City, MO, USA, 2018, pp. 1-6.

[8] W.-P. Wu, Y. -C. Wu, C. -C. Hsu, J. -S. Leu and J. -T. Wang, "Design and implementation of vehicle speed estimation using road marking-based perspective transformation," in *IEEE 93rd Veh. Technol. Conf. (VTC)*, Helsinki, Finland, 2021, pp. 1-5.

[9] F. Zhang, X. Wu and C. Gu, "Detection of road surface identifiers based on deep learning," in *Int. Conf. Artif. Intell. and Adv. Manuf. (AIAM)*, Dublin, Ireland, 2019, pp. 66-70.

[10] R. Qian, Q. Liu, Y. Yue, F. Coenen and B. Zhang, "Road surface traffic sign detection with hybrid region proposal and fast R-CNN," in *12th Int. Conf. Natural Computation, Fuzzy Syst. Knowl. Discovery (ICNC-FSKD)*, Changsha, China, 2016, pp. 555-559.

[11] S. Suchitra, R. K. Satzoda and T. Srikanthan, "Detection & classification of arrow markings on roads using signed edge signatures," in *IEEE Intell. Vehicles Symp. (IV)*, Madrid, Spain, 2012, pp. 796-801.

[12] H. Vokhidov, H. G. Hong, J. K. Kang, T.M. Hoang, and K. R. Park, "Recognition of damaged arrow-road markings by visible light camera sensor based on convolutional neural network", *Sensors*, vol. 16, no. 12, 2016.

[13] T. M. Hoang, S. H. Nam and K. R. Park, "Enhanced detection and recognition of road markings based on adaptive region of interest and deep learning," *IEEE Access*, vol. 7, pp. 109817-109832, 2019.

[14] Z. Li, Z. Cai, J. Xie and X. Ren, "Road markings extraction based on threshold segmentation," in 9th *Int. Conf. Fuzzy Syst. Knowl. Discovery*, Chongqing, China, 2012, pp. 1924-1928.

[15] Y. He, S. Chen, Y. Pan and K. Ni, "Using edit distance and junction feature to detect and recognize arrow road marking," in *17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Qingdao, China, 2014, pp. 2317-2323.

[16] P. Sanjeewani, B. Verma and J. Affum, "A novel evolving classifier with a false alarm class for speed limit sign recognition," in *IEEE Congr. Evol. Computation (CEC)*, Kraków, Poland, 2021, pp. 2211-2217.

[17] L.-C. Chen, Z. Zhu, G. Papandreou, F. Schroff and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation." in *Eur. Conf. Comput. Vision (ECCV)*, Munich, Germany, 2018, pp 833-851.

[18] J. Long, E. Shelhamer, T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, 2015.