Artificial Intelligence Application in Automated Odometer Mileage Recognition of Freight Vehicles

Pornpimol Chaiwuttisak

Department of Statistics, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10520, Thailand
Corresponding author. E-mail address: pornpimol.ch@kmitl.ac.th

Received: 22 July 2021; Revised: 12 May 2023; Accepted: 29 May 2023; Available online: 26 August 2023

Abstract

Transportation cost management is necessary for entrepreneurs in industry and business. One way to do this is to report the daily mileage numbers read by the employees of a company, but still encounter errors in human mileage reading, resulting in incorrect information received and difficulties in planning effective revenue management, as well as increased workload and complications for employees in checking mileage information. Therefore, the objective of this research was to create a machine-learning model for detecting and reading the mileage numbers of both digital and analog odometer displays of freight vehicles. The model has two processes; 1) To identify the position of the odometer in the speedometer image and remove unrelated backgrounds from the image, and 2) To read the mileage figure from the isolated odometer image. Both processes use object detection with the Faster-RCNN. To detect the position of the odometer , isolate and then read the mileage correctly, 220 test images were used and 187 of these images were correctly identified (85% accuracy). The accuracy of object detection on the analog odometer was 98.53% and the accuracy of object detection on the digital odometer was 97.37%. The accuracy of classification of the analog odometer was 83.82% with 85.53% accuracy of classification of the digital odometer. The results of the study demonstrated satisfactory performance that meets the requirements needed for real-life applications in the transportation and logistics industry.

Keywords: Object Detection, Mileage Reading, Faster-RCNN, Freight Vehicles

Introduction

Freight costs are a major component of supply chain costs. Fuel consumption and therefore costs can be estimated from the number of miles (kilometers) traveled shown on a vehicle's odometer, which can be either analog or digital. Auditing daily mileage is important to monitor the length of journeys and calculate fuel usage, particularly to avoid fuel fraud.

The objective of this research was to develop a learning model based on Artificial Intelligence (AI) to automate mileage readings and record mileage figures from the odometer on freight vehicles. The case study used in the research was a company operating delivery services from several distribution centers. Before leaving a distribution center to commence a delivery trip, the mileage will be manually read by staff. The mileage was recorded on arrival at the delivery point and again as the delivery vehicle returned to the distribution center. These mileage figures were used to calculate the amount of fuel used on each round trip. However, problems of inaccuracy and incompleteness of data often occurred, as a result of human error when recording the odometer readings. The possibility of actual fraud in odometer readings could also not be discounted.

Artificial Neural Networks (ANN), first presented by McCulloch and Pitts (1943), are based on the simulation of the human brain as an intelligent learning machine. ANNs have been deployed in many applications, particularly where image processing and image content identification are required.

The architecture of the neural network with one hidden layer is shown in Figure 1. In principle, it can be divided into 3 layers as follows:

1. The input layer is the first layer of the neural network. It has a neuron that acts like a human neuron to receive and transmit information. The number of neurons in this layer is equal to the number of input data items. This layer is responsible for receiving data into the neural network model.

2. The hidden layer is an intermediate layer of the neural network, which affects the model's learning efficiency. It serves the function of multiplying the input data values and their weight and sending the result into the output layer.

3. The output layer consists of output neurons. The number of neurons in this layer is equal to the number of output variables that represent the model's output of the model.

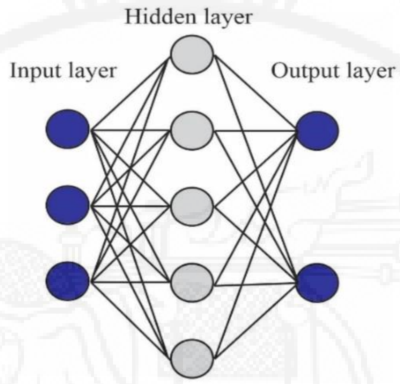


Figure 1 A structure of Neural Network (Dadras & Huang, 2022)

Convolutional Neural Network (CNN) introduced by LeCun (1989) is very similar to a neural network. The difference is that the convolutional neural network is designed specifically for the task that uses images as input. The layer of a convolutional neural network is different from that of the artificial neural network. The convolutional neural network has neurons arranged in three dimensions consisting of width, height, and depth. It is used in object detection, identifying the locations of objects in images.

The most common layers of the convolutional neural network consisted of 5 layers, each of which is detailed as follows:

1. An input layer is often used to express visual data as a matrix. For a color (RGB) image, there will be 3 matrices associated with the image. For example, the color image is 32 x 32 (width x height), so the input layer will have an input dimension of 32 x 32 x 3.

2. A convolution layer consists of a filter or kernel to calculate the weight of the area which the filter has moved through. The filter can have multiple layers to increase the number of feature maps. Each feature map learns different information about the image, such as a border or dot pitch.

3. An activation layer is composed of a function that is in between the input feeding and the output connected to the layers. A popular activation function in convolution neural networks is the Rectified Linear Unit (ReLU) function, where all the negative value is turned to 0, while the positive value is still retained and the size does not change. In addition, one popular function, the SoftMax function, which is a classification function, is used in the last layer of the network to classify inputs into multiple categories based on considering the maximum probability as in Equation 1.

$$f(x)_j = \frac{e^{x_j}}{\sum_{i=1}^{k} e^{x_i}} \tag{1}$$

4. The Pooling layer is a layer that reduces the width and height of the image but retains the depth of the image. The popular pooling techniques consist of 2 types: Max pooling and Average pooling.

5. The fully connected layer is the last layer for classification. Namely, the neurons in this layer connect to the activation of the neurons in the previous layer.

The concept of R-CNN first proposed by Girshick, Donahue, Darrell, and Malik (2014) is to use a selective search (SS) approach to select around 2,000 Regions-Of-Interest (ROI) that contain the object and then fed into a convolutional neural network to extract features and receive a feature map. These features were used to classify the images and their object boundaries using support vector machines (SVM) and regression methods. Whereas Fast R-CNN which is improved over the R-CNN loads the whole image as an input to a convolutional neural network instead of using a selective search algorithm to identify the region proposals and provide a feature map including reshaping a feature map of all the region proposals. The predicted region proposals are then adjusted using an ROI pooling used to classify the image within the proposed region and predict the offset values for the bounding boxes. Therefore, Faster R-CNN is faster than R-CNN because it does not have to feed 2,000 region proposals to the convolutional neural network every time (Girshick, 2015).

Acharya and Fung (2020) studied the mileage extraction from odometer images for automating insurance claims processing. The company provided customers with an easy and flexible way to provide all the information required when reporting a claim or request for a quote. It also reduced human errors and speeds up the process of data collection, where accurate mileage readings are essential to apply to car insurance quotes and process claims in sequence. Therefore, the study objective was to extract the mileage from the odometer image using object detection which will be divided into two parts to locate the miles in the image by using the structure of Faster R-CNN and single shot detector (SSD), As a result, the Faster R-CNN structure was more accurate than the model built with the SSD structure.

Bulan, Kozitsky, Ramesh, and Shreve (2017) Masood, Shu, Dehghan, and Ortiz (2017) improved the accuracy of automatic license plate recognition (ALPR) systems by applying CNNs to synthetically generated images. Girshick (2015) introduced the Faster R-CNN which used a selective search for region proposals and convolutional feature maps as input. This system improved the speed of object detection. Nagaoka, Miyazaki, Sugaya, and Omachi (2017) presented a text detection method based on a multi-region proposal network (Multi-RPN) to detect texts of various sizes simultaneously. As a result, the proposed Multi-RPN method improved detection scores and kept almost the same detection speed as compared to the original Faster R-CNN.

Ravirathinam and Patawari (2020) studied Indian road license plate detection and reading using Faster-RCNN to detect and read Indian license plates. It is challenging due to the diverse nature of license plates such as font, size, and the number of lines. It applied the structure of the Faster R-CNN and extracted the features with Resnet50 and VGG16. As a result, Res-net50 provided higher mAP than VGG16. An 88.5% accuracy in detecting and reading car license plates was achieved.

Chauhan, Rastogi, Gaur, Singh, and Gupta (2020) studied a traffic sign detection system using a convolutional neural network trained for the extraction of the characteristics of traffic signs using You Only Look Once (YOLO). Nine types of mandatory traffic signs were tested, including overtaking warnings and prohibitions, U-turns, signs controlling left and right turns, and signs prohibiting lane change. The design of this traffic sign detection system consists of 4 parts: preparation of training data sets, building a model, testing the model, and performance evaluation. The model was tested with a total of 108 traffic sign data sets and had a precision value of 0.87%, a recall value of 0.89%, and an F-Measure of 0.88%.

Methods and Materials

Data used in this research were RGB images saved as a JPEG file format with a variety of resolutions starting from 720 x 720 pixels and up to 2730 x 1536 pixels. Four different types of freight vehicles were in use in the case study company with either analog or digital odometers. The analog odometers are either blue or black (Figure 2), and there were two different types of digital odometers (Figure 3).

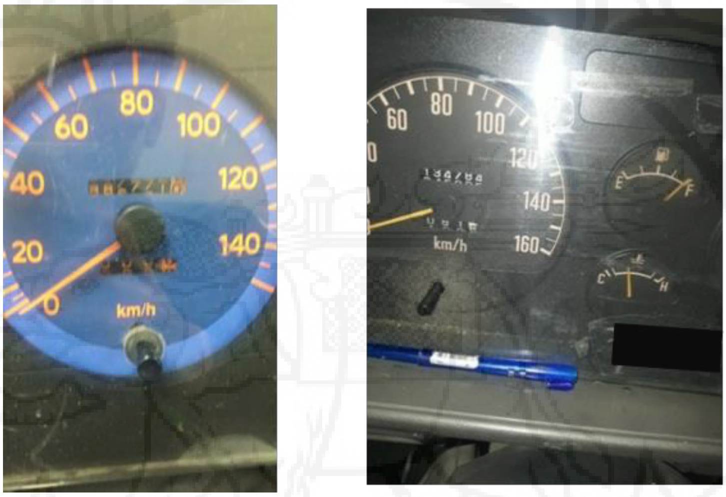


Figure 2 Analog Odometer

Figure 3 Digital Odometer

There were a total of 635 images used in the study. There were 376 blue and 14 black analog odometer images, 390 images in total, with 183 images digital odometer images taken from digital dials below the speedometer and 62 images taken from red digital dials, 245 digital images in total. A simple random sampling (SRS) technique was used to divide the images into two datasets, based on the different odometer types of odometers. Training datasets contained 80% of the data with the remaining 20% used for testing. Data augmentation was applied to build the various images such as the number of pixels in images to make images lighter or darker, noise enhancement, image quality degradation, and image rotation to increase the size of the training dataset for improving the model. Data augmentation results in more data in the training dataset.

Figure 4 shows the process to detect the area of images containing the mileage based on the Faster-RCNN algorithms (Liu et al., 2020). It starts with using the algorithm to propose areas that contain an interesting object with a selective search. It will present large areas, and repeatedly combine similar areas into a large area. All areas are scaled evenly before presenting the final region to the convolutional neural network. The convolutional neural network extracts different features and then uses a support vector machine (SVM) to classify objects and create bounding box regression to improve the object's coordinates and create an enclosed square box as shown in Figure 5.

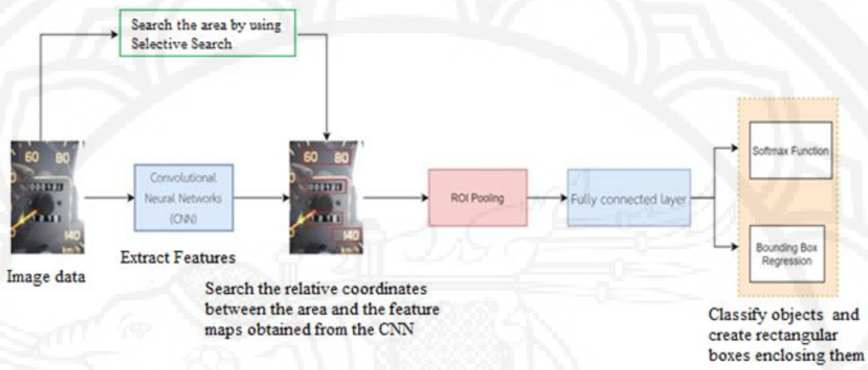


Figure 4 The process of object detection

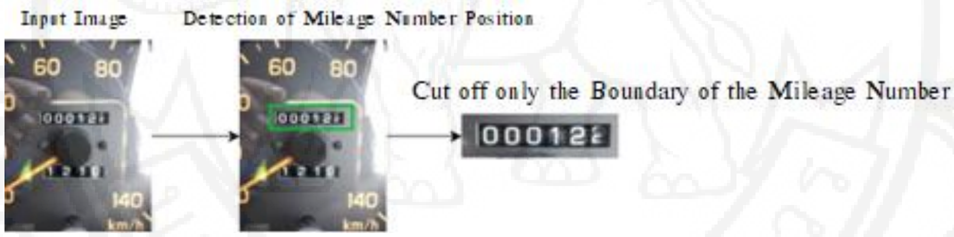


Figure 5 An example image of the result of the mileage's position detection procedure

After the object detection process, the images obtained from the mileage detection process are loaded and features are extracted with Inception V2 by labeling 0 through 9. This dataset is trained with the Faster RCNN Inception v2 COCO pre-trained model after completing the training process. Next, the test dataset is examined. It will look like a square frame that is used to enclose an object overlaid with a numbered area.

The mileage reading procedure is based on data labels obtained by detecting the numbers in the image and then starting to read each data as a square to enclose the object that overlaps the numbered area in the image. which will start reading from the far left to the far right. The rectangles enclosing each object give reference to the x- and y-axis coordinates, so the x (min) coordinates of all rectangles enclosing the object are placed in ascending order and the numbers are read based on the x (min) coordinates of each rectangle enclosing the smallest to largest object one by one. Figure 6 illustrates the example of detecting the numbers in the image.

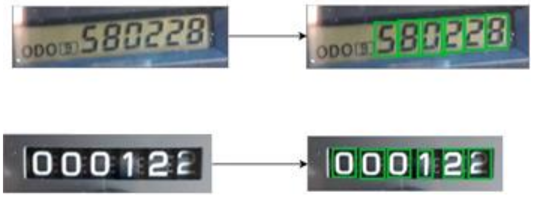


Figure 6 An example image of the result of the mileage number detection procedure

The xmin, xmax, ymin, ymax coordinate values are recorded in an extensible markup language (XML) file format. It is converted to a comma-separated value (CSV) file for building the model, as shown in Figure 7.

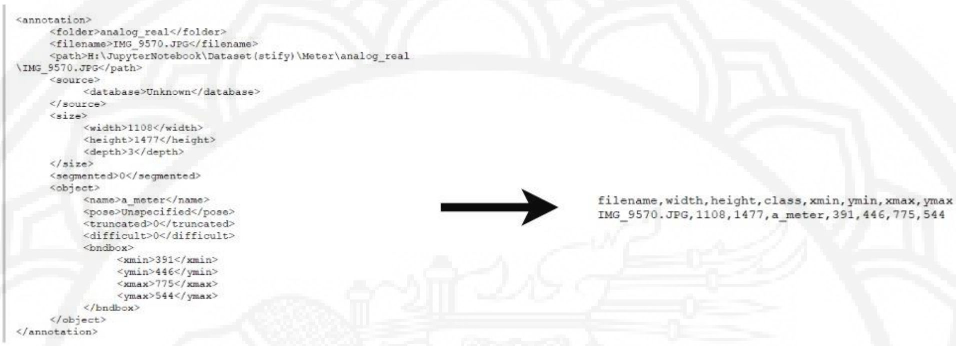
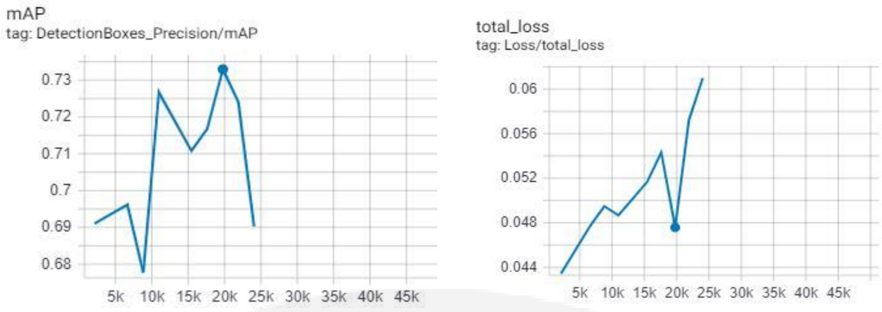


Figure 7 The coordination of mileage number in an XML file and converting it to a CSV file.

Results and Discusssion

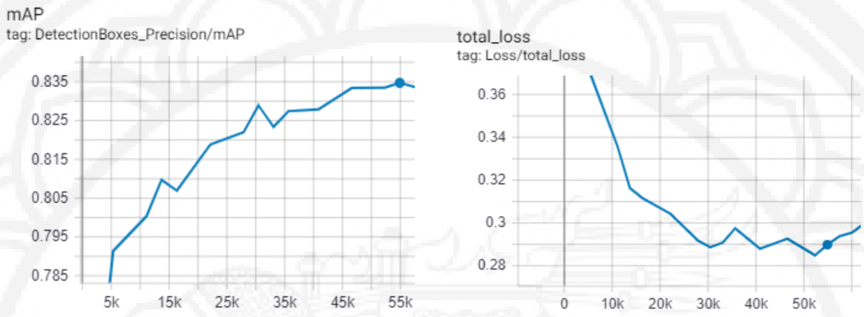
During the training of the model, the model checkpoint is defined every 10 minutes to evaluate the model performance and determine the mean average precision (mAP) and the loss value. Training the model was stopped when the loss increased while the mAP decreased. The learning rate for detecting mileage is defined as 0.0005. The results as shown in Figures 8(a) and 8(b) indicate that the total learning cycle is equal to 24,046 iterations. It takes about 1 hour 39 minutes and 57 seconds before the training stops since the loss increases while the mAP decreases. Furthermore, the highest precision was recorded at 19,774 cycles. Figure 8(a) illustrates the mAP. The Y-axis denotes the mAP and the X-axis denotes the number of the training cycles, whereas the circle dot on the chart denotes the model's mAP is 0.733 at the cycle of 19,774. Figure 8(b) illustrates the total loss. The Y-axis represents the total loss, and the X-axis represents the number of training cycles, whereas the circle dot on the chart represents the model's total loss is 0.0476 at cycle 19,774.

From Figures 9(a) and 9(b), the total time used to train the model for mileage reading was 3 hours 30 minutes 6 seconds before the model training was stopped because the loss increased while the mAP decreased. The highest precision was recorded at 54,907 cycles. Figure 9(a) illustrated the mAP. The Y-axis denotes the mAP and the X-axis denotes the number of the training cycles, whereas the circle dot on the chart denotes the model's mAP is 0.835 at the cycle of 54,907. Figure 8(b) illustrated the total loss. The Y-axis represents the total loss, and the X-axis represents the number of training cycles, whereas the circle dot on the chart represents the model's total loss is 0.2898 at the cycle of 54,907.



(a) (b)

Figure 8 The detection object model



(a) (b)

Figure 9 The mileage reading model

Figure 10 shows an example of testing the object detection model (a) the mileage detection and (b) cutting specific areas of mileage numbers. Figure 11 shows the rectangular frame to predict mileage numbers by the model. The number on the top of the rectangular frame shows the predicted digit with the highest percentage of the confidence score.

(a) (b)

Figure 10 An example of the result while training the model for mileage detection

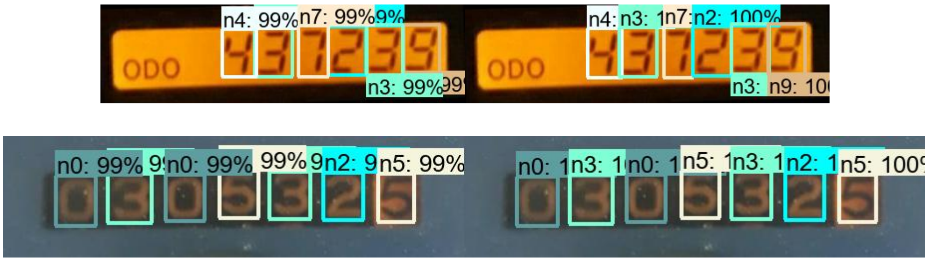


Figure 11 An example of the result while training the model for reading mileage

A total of 220 images were used to test the accuracy of object detection and reading the 6-digit mileage in both the analog and digital odometers. The experimental results are shown in Table 1.

Table 1 Results of object detection and mileage reading

Type of Odometer	# Images of Testing dataset	Accuracy of object detection	%	Accuracy of mileage reading	%
Analog	68	67	98.53	57	83.82
Digital	152	148	97.37	130	85.53

Table 2 of the Chi-square hypothesis testing result shows that the accuracy ratio in detecting and reading the miles on both analog and digital odometers is equal to 85:15 at the significant level of 0.05.

Table 2 Chi-Square hypothesis testing

Chi-Square	df	Asymp. Sig.
0.074	1	0.786

The Precision, Recall, and F-Measure of reading mileages on both analog and digital odometers are shown in Tables 3 and 4 respectively. The result shows the precision of numbers 2, 4, and 8 in the analog odometer is equal to 1. It can say that the model accurately and precisely detected and classified every time the predicted numbers 2, 4, and 8 of the analog numbers shown in Table 3, as well as the digital numbers 6 and 9 shown in Table 4, while digital numbers 0 and 1 have less precision than other numbers. There are a few small errors in the classification of the numbers, for example, the number 0 or 1 cannot be detected or classified properly in some images. Recall shows that the model can predict the numbers correctly with the actual solution relative to the total number of actual solutions of that number. From Table 3, it is found that the recall of the analog numbers 1, 2, 5, and 7 is equal to 1. Namely, the model can correctly detect analog numbers that contain numbers 1, 2, 5, and 7 as well as digital numbers consisting of 0, 2, 3, 6, and 7 shown in Table 4, while recall values for analog numbers 4 and 8 are equal to 0.948 and 0.933 respectively which are less than the recall values of other numbers. Moreover, F-Measure is preferred to evaluate the performance of the learning model because it is the mean between precision and recall.

Table 3 Result of the Precision, Recall, F−measure for analog mileage

Number	Precision	Recall	F-Measure
1	0.941	1.000	0.970
2	1.000	1.000	1.000
3	0.931	0.982	0.956
4	1.000	0.948	0.973
5	0.983	1.000	0.991
6	0.978	0.978	0.978
7	0.976	1.000	0.988
8	1.000	0.933	0.965
9	0.955	0.977	0.966
0	0.946	0.981	0.963
Average	0.971	0.98	0.975

Table 4 Result of the Precision, Recall, F−measure for digital mileage

Number	Precision	Recall	F-Measure
1	0.917	0.989	0.952
2	0.993	1.000	0.996
3	0.989	1.000	0.994
4	0.976	0.976	0.976
5	0.987	0.987	0.987
6	1.000	1.000	1.000
7	0.963	1.000	0.981
8	0.990	0.970	0.980
9	1.000	0.969	0.984
0	0.921	1.000	0.959
Average	0.974	0.989	0.981

Figures 12 and 13 show some errors in that the model cannot correctly read mileage numbers and cannot detect all numbers in the image. A mileage detection error is the inability of the model to detect the mileage position in the image shown in Figure 14. It is found that in Figure 12, there was one digit mistake. Namely, the predicted mileage number obtained from the model was 5810352 but the actual mileage number on the odometer was 5810362. While there was the last digit that cannot be detected in Figure 13.

Figure 12 An example of an image that reads the mileage incorrectly

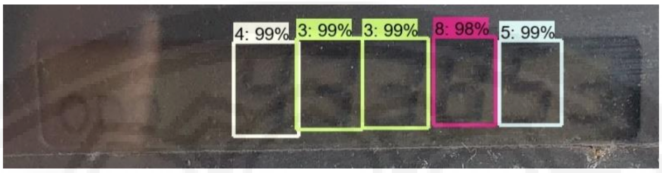


Figure 13 An example of an image that does not detect all numbers

Figure 14 An example of an image that cannot detect the mileage

Conclusion and Suggestions

When comparing the model of the company applied to the same dataset, it is found that the model for mileage detection and mileage read with the Faster-RCNN framework provides considerable accuracy. The accuracy of the model for detecting and reading analog and digital mileage was 87.5% and 93% respectively, while the accuracy of the company's model based on the commercial tools was 50% and 87% respectively. Thus, the developed machine learning model could replace the relatively expensive commercial software. This enables the company to detect errors in the mileage reading by drivers. The mileage figures are also used to provide a time frame for fleet maintenance on time and to plan efficient logistics management strategies. It can also be applied in conjunction with GPS data analysis to prevent fuel theft.

However, it is found that incorrect mileage detection and readings were detected in 33 images. The 3 types of errors found in this work while detecting mileage, detecting numbers, and reading miles were recorded at 15%, 70%, and 15% respectively.

It was found that some images have interference in detecting mileage and reading mileage such as reflection on an odometer and scratches on an odometer. These are some external factors that cannot be controlled. Consequently, it causes an error in mileage detection and reading of the mileage. Moreover, one of the problems with the detection and reading of analog mileage is that the number is halfway between the two values. The detection and mileage reading model developed in this research cannot fully support the problem, as there are not enough numerical images to support the development of a machine learning model effectively

References

Acharya, S., & Fung, G. (2020). *Mileage Extraction from Odometer Pictures for Automating Auto Insurance Processes*, Retrieved from https://doi.org/10.3389/fams.2019.00061

Bulan, O., Kozitsky, V., Ramesh, P., & Shreve, M. (2017). Segmentation- and Annotation-Free License Plate Recognition with Deep Localization and Failure Identification. *IEEE Transactions on Intelligent Transportation Systems*, *18*(9), 2351-2363.

Chauhan, A., Rastogi, A., Gaur, A., Singh, A., & Gupta, S. (2020). Traffic Sign Detection Using Deep Learning. *International Journal of Engineering Applied Sciences and Technology*, *5*, 355-358.

Dadras, E. A., & Huang, S. (2022). Artificial-Neural-Network-Based Surrogate Models for Structural Health Monitoring of Civil Structures: A Literature Review. Buildings, *12*(12), 1-28.

Girshick, R.B., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, *2014*, 580-587.

Girshick, R. B. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015 (pp. 1440–1448), Beijing.

LeCun, Y. (1989). Generalization and Network Design Strategies. In R. Pfeifer, Z. Schreter, Fogelman, F., & L. Steels, (Eds.), *Connectionism in Perspective*, (pp. 132-134). Zurich, Switzerland: Elsevier.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). *Deep Learning for Generic Object Detection: A Survey*. Retrieved from https://doi.org/10.1007/s11263-019-01247-4

Masood, S.Z., Shu, G., Dehghan, A., & Ortiz, E. (2017). *License Plate Detection and Recognition Using Deeply Learned Convolutional Neural Networks.* Winter Park, FL: Computer Vision Lab, Sighthound.

McCulloch, W. S., & Pitts, W. A. (1943). logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115-133.

Nagaoka, Y., Miyazaki, T., Sugaya, Y., & Omachi, S. (2017). Text Detection by Faster R-CNN with Multiple Region Proposal Networks. *Proceedings14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 9-15 November 2017 (pp. 15-20). Kyoto, Japan.

Ravirathinam, P., & Patawari, A. (2020). *Automatic License Plate Recognition for Indian Roads Using Faster-RCNN.* Retrieved from https://www.researchgate.net/publication/338116738_Automatic_License_Plate_Recognition_for_Indian_Roads_Using_Faster-RCNN