A MACHINE LEARNING MODEL THAT ANALYZES SURROUNDING ROAD SIGNS TO ELIMINATE DANGERS CAUSED BY HUMAN OPERATIONAL ERROR

Annie Wu[1], Yu Sun[2]

[1]Barrington High School, 616 W Main St, Barrington, IL 60010
[2]California State Polytechnic University, Pomona, CA, 91768, Irvine, CA 92620

ABSTRACT

Autonomous vehicles are a potential solution to preventing crashes caused by human error. Although road signs are intended to attract drivers' attention and help them operate, drivers can still misinterpret signs, resulting in an accident. An autonomous vehicle system can implement artificial intelligence to detect and recognize known patterns in input graphics to minimize the human aspect of driving. In this study, we present an implementation of the CNN architecture to classify four regulatory instruments (stop, crosswalk, speed limit sign, and traffic light) using the TensorFlow library. We used a training dataset of 877 images of the four distinct classes to optimize the model. The goal of the study was to create a lightweight and accessible image classification model. Experimental results show a 92% model accuracy.

KEYWORDS

Machine Learning, Image classification, Autonomous vehicle

1. INTRODUCTION

Autonomous vehicles have the potential to prevent many types of accidents caused by human error [1]. Mistakes, such as rear ending other vehicles and speeding past stop signs, made by drivers who are not paying attention, would not happen if drivers were operating self-driving cars. Human error when driving causes 90% of accidents that happen on the road. By minimizing the driver's role, autonomous cars can reduce the effects caused by human error.

The autonomous vehicle industry rises globally by 16% every year, and the global value of its market is estimated to be worth more than $1 trillion by 2025 [2]. As autonomous vehicles become more popular, the AI that operates these vehicles also needs improvements to develop safer systems. These vehicles rely on cameras and sensors to capture data, which is then analyzed by AI. The AI performs image classification to determine what objects around the vehicle are, and image localization to determine how far the objects are from the vehicle [3].

Through image classification, an autonomous vehicle can see traffic regulatory signs/lights and make the proper adjustments to adhere to them [4]. Autonomous vehicles need to be able to classify each image precisely, as not doing so can result in a mistake while on the road, which may cause a serious accident and the loss of life. It can also be beneficial to implement a road

David C. Wyld et al. (Eds): NLPML, AIAP, SIGL, CRIS, COSIT, DMA -2023
pp. 159-165, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130813

sign recognition software in a regular vehicle so that drivers can be alerted when the AI detects road signs, such as speed limits or stop signs.

Machine learning algorithms are able to analyze pixels in an input image for known patterns [5]. Therefore, the algorithm can make an accurate prediction of the classification of an unknown feature vector by learning from a large set of feature vectors whose classifications are known. Applications of image classification algorithms can be seen in security cameras, facial recognition, and object recognition technologies.

Image classification has been approached with many methods. Computer vision methods, for example, have been proposed to detect and classify road signs. However, computer vision methods often require regular monitoring and heavy manual work to craft important features in images. This, alongside the current lack of computer vision specialists, makes it more difficult to implement the computer vision method to classify road signs.

Component-based machine learning models exist but require much more work to create than an end-to-end machine learning model, which requires a smaller amount of maintenance[6]. Moreover, an end-to-end image classification model can be retrained to classify different images by simply adding more data to the set/importing a new training dataset, while component-based cannot.

The K-nearest neighbors algorithm has also been used to classify road signs, but is slow when handling a large dataset, which is necessary to improve the accuracy of the prediction model.
In a study that aimed to classify road sign images, researchers tried to convert road sign images to grayscale and filtered with SGW, then classified using the support vector machine model (SVM) [7]. Additionally, the study uses datasets of German and Chinese traffic signs, while our implemented training dataset consisted of a wider variety of international traffic signs, therefore resulting in increased accessibility of our model in more countries.

In this study, we aim to classify a road sign image. Firstly, we trained the model with a dataset of 877 images of 4 distinct classes of road signs, which are traffic light, crosswalk sign, stop sign, and railroad sign. Then, we created an algorithm using the TensorFlow training environment to identify a road sign type. Our model does not rely on manual aid to identify important features in road sign images, resulting in ease of use. Furthermore, our model can be quickly implemented and used without frequent monitoring, eliminating the costs of a technical team dedicated to maintenance. All the user has to do is upload an image, and the model will return the predicted road sign type and how certain it is of the prediction. With the large dataset of images and the practicality, we believe that our method is maintainable, easy to employ, and capable of consistently identifying road signs with high accuracy.

We conducted various evaluations to find the outstanding image classification algorithm that yielded the highest accuracy. We split the dataset into two groups, 90% training data and 10% testing data. By running multiple epochs using this organized dataset, we passed the dataset to the network many times and were able to find the optimal road sign classification model. Additionally, we have tested the model with individual road sign images of different types. The accuracy of the predictions were recorded and analyzed. Increasing the accuracy of our prediction model means the autonomous car employing the technology would be able to be more aware of its surroundings. This would assist in making driving safer.

The rest of the paper is structured as follows: Section 2 describes the challenges that were met during the process of finding models that can predict road signs with high accuracy; Section 3 focuses on our solutions to the challenges presented in Section 2; Section 4 details the relevant details of the experiments we did; Finally, Section 5 gives our concluding remarks as well as future work of this project.

1.2. RELATED WORK

Shao, F. et al developed a real-time road sign recognition algorithm using simplified Gabor Wavelets [8]. They converted input data to grayscale, extracted regions of interest using the maximally stable extremal regions algorithm to classify each image. Their model is trained to make predictions for Chinese and German road signs, while ours includes American road signs as well.

Sermanet, P. et al applied ConvNets to the process of traffic sign classification [9]. Their system consists of biologically inspired architectures to automatically learn features from data. Input images are in the dimensions 32x32, while our input images are in the scale 224x224. Both our models automatically learn features from training data.

Zaibi, A et al created a classification model using an enhanced LeNet-5 network, a kind of convolutional neural network [10]. They acquired training data from German and Belgian resources, while our training data consists of both European and American road signs, leading to a wider variety of input that the model can interact with.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Finding A Model with High Accuracy

In order to improve road safety, the classification model needs to provide accurate predictions of detected road signs. As weather conditions, light quality, and sign visibility may vary, the model needs to be trained with a dataset that allows it to then operate effectively in these various conditions. We intend to add more varied data to its training set and increase training epochs, which subsequently will increase the model’s accuracy rate.

2.2. Computer Needs High Settings in Order to Classify Images

Digital images are made up of many pixels. Because of this, a computer needs high settings to classify images. A computer can put out a lot of heat if its GPU resources are overworked. The AI needed to be quick without overwhelming the computer’s resources. We tested multiple IDE’s to determine which would have the best processing power, and changed the hardware accelerator to GPU.

2.3. Finding the Best Image Size

It can be challenging to determine the image dimensions that produce the most accurate results. If the image size is too small, the model may be unable to notice features characteristic of a class that help it make the correct prediction. If the image size is too large, it can cause overfitting and seize a large portion of the model’s processing power usage without improving its accuracy.

3. SOLUTION

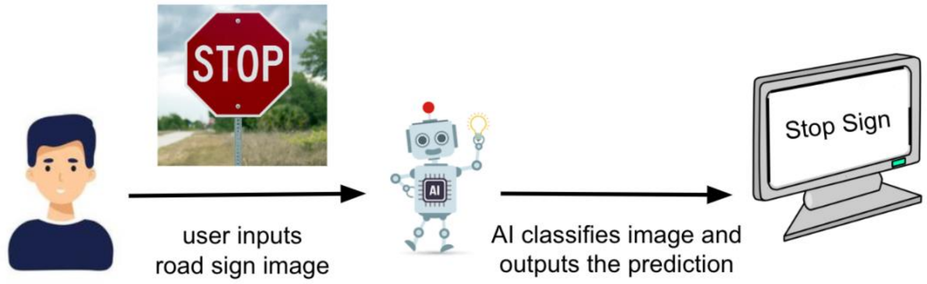

Figure 1. Overview of the solution

We have created a machine learning model that classifies a road sign from an image provided by the user. The processing flow is shown in Figure 1. To perform this, we first imported a dataset consisting of various images of four different types of traffic signs from Kaggle [11]. All the images are regrouped into directories with the names: stop, speed limit, crosswalk, or traffic light, depending on how the contents of the image is to be classified.

To train the model, we first organized the aforementioned dataset into 90% training and 10% testing data, which allowed us to take the machine learning algorithm through multiple full training iterations. We created a visual representation of the accuracy of the prediction by plotting a graph composed of images from the dataset, and used a system of red text for incorrect classification and black test for correct classification of a given traffic sign. After completing five epochs, the results show 91.8% of the given training images were classified correctly.

After being trained by the system, the program was exported as a TFLite model [12]. We loaded the finished model, and tested for accuracy by inputting web images of stop signs, speed limit signs, crosswalk signs, and traffic light signs, to see if the model's predictions were correct.

When the model is implemented, a user only needs to input an image. The AI in the backend makes the prediction and returns to the console the predicted road sign and its confidence level.

```
train_data, test_data = data.split(0.90)
model = image_classifier.create(train_data)
loss, accuracy = model.evaluate(test_data)

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 hub_keras_layer_v1v2 (HubKe  (None, 1280)             3413024
 rasLayerV1V2)

 dropout (Dropout)           (None, 1280)              0

 dense (Dense)               (None, 4)                 5124

=================================================================
Total params: 3,418,148
Trainable params: 5,124
Non-trainable params: 3,413,024
_________________________________________________________________
None
Epoch 1/5
24/24 [==============================] - 18s 228ms/step - loss: 0.9242 - accuracy: 0.7539
Epoch 2/5
24/24 [==============================] - 6s 258ms/step - loss: 0.6414 - accuracy: 0.8737
Epoch 3/5
24/24 [==============================] - 6s 259ms/step - loss: 0.5786 - accuracy: 0.9102
Epoch 4/5
24/24 [==============================] - 6s 259ms/step - loss: 0.5456 - accuracy: 0.9180
Epoch 5/5
24/24 [==============================] - 6s 257ms/step - loss: 0.5247 - accuracy: 0.9336
3/3 [==============================] - 5s 409ms/step - loss: 0.6119 - accuracy: 0.8750
```

Figure 2. The code to train the model

We split the dataset into 90% training data and 10% testing data, as shown in Figure 2. The machine learning algorithm was created in Google Colab [13]. Using the Tensorflow library, we ran five epochs, allowing the algorithm to process the samples multiple times.

4. EXPERIMENT

4.1. Experiment 1

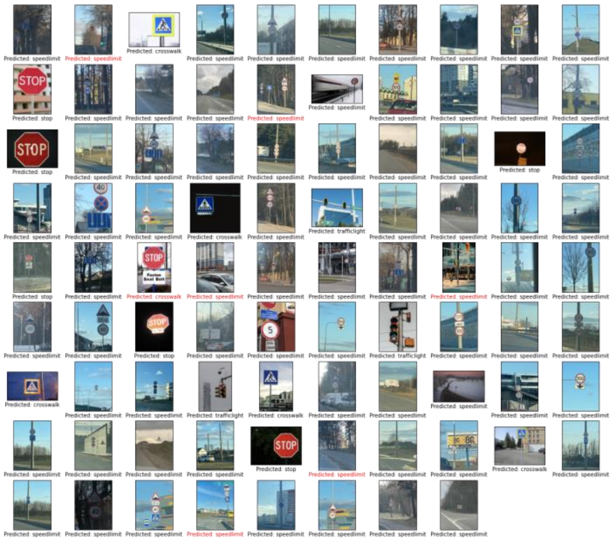

Figure 3. Sample images

After the algorithm has learned using the training data, we input testing images to the model in order to evaluate the result. The predicted results are plotted using matplotlib, shown in Figure 3. If a “Predicted: ” result is different from the label in the dataset, the text is printed in red. Predictions labeled using green are otherwise correct.

In this sample, only 7 out of 88 images are predicted incorrectly. 92% of the testing data was predicted correctly.

Figure 4. Examples of analysis result

The result in Figure 4 shows that the network speed is the main factor which affects the average connection delay.

5. Conclusions

In this project, we propose an application to predict the name of a road sign from user input graphics using artificial intelligence. We developed a machine learning model that is trained using a large dataset that consists of our 4 main road signs. The system is then able to classify images of road signs provided by the user. Experiments show that our model was able to classify 92% of training data correctly.

The model is only able to classify correctly images that are either: a stop sign, crosswalk sign, stoplight, pedestrian sign. This is due to our training dataset having only four classes. Our future research will include gathering a larger training dataset consisting of more classes to include every road sign currently in use.

References

[1] Schwarting, Wilko, Javier Alonso-Mora, and Daniela Rus. "Planning and decision-making for autonomous vehicles." Annual Review of Control, Robotics, and Autonomous Systems 1 (2018): 187-210.

[2] Duarte, Fábio, and Carlo Ratti. "The impact of autonomous vehicles on cities: A review." Journal of Urban Technology 25.4 (2018): 3-18.

[3] Lu, Dengsheng, and Qihao Weng. "A survey of image classification methods and techniques for improving classification performance." International journal of Remote sensing 28.5 (2007): 823-870.

[4] Haralick, Robert M., Karthikeyan Shanmugam, and Its' Hak Dinstein. "Textural features for image classification." IEEE Transactions on systems, man, and cybernetics 6 (1973): 610-621.
[5] Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9 (2020): 381-386.
[6] Geyer, Philipp, and Sundaravelpandian Singaravel. "Component-based machine learning for performance prediction in building design." Applied energy 228 (2018): 1439-1453.
[7] Lessmann, Stefan, Robert Stahlbock, and Sven F. Crone. "Genetic algorithms for support vector machine model selection." The 2006 IEEE International Joint Conference on Neural Network Proceedings. IEEE, 2006.
[8] Shao F, Wang X, Meng F, Rui T, Wang D, Tang J. Real-Time Traffic Sign Detection and Recognition Method Based on Simplified Gabor Wavelets and CNNs. *Sensors*. 2018; 18(10):3192.
[9] Sermanet, Pierre & Lecun, Yann. (2011). Traffic sign recognition with multi-scale Convolutional Networks. 2809 - 2813. 10.1109/IJCNN.2011.6033589.
[10] Zaibi, Ameur & Anis, Ladgham & Sakly, Anis. (2021). A Lightweight Model for Traffic Sign Classification Based on Enhanced LeNet-5 Network. Journal of Sensors. 2021. 10.1155/2021/8870529.
[11] Bojer, Casper Solheim, and Jens Peder Meldgaard. "Kaggle forecasting competitions: An overlooked learning opportunity." International Journal of Forecasting 37.2 (2021): 587-603.
[12] Neural Networks: Artificial Intelligence: A Modern Approach, by Stuart Jonathan. Russell and Peter Norvig, pp. 736–748. Prentice Hall Pearson Education International (2003).
[13] Alves, Francisco Regis Vieira, and Renata Passos Machado Vieira. "The Newton fractal's Leonardo sequence study with the Google Colab." International Electronic Journal of Mathematics Education 15.2 (2019): em0575.