Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/31/20230138

Optimizing road sign detection using the segment anything model for background pixel exclusion

Zhimeng Liu

Department of Computer Science, University of Manchester, Manchester, M13 9PL, British

zhimeng.liu@student.manchester.ac.uk

Abstract. Road sign recognition plays a great role in automatic driving. At present, for the task of road sign recognition, the structure related to CNN is mainly used to classify the detected road signs. However, due to the complexity of road conditions and weather, the detected images often contain complex background pixels, which affects the accuracy of classification. Due to the release of segment anything model, there is a powerful tool for eliminating background pixels quickly and conveniently. We optimize the data to retain only the road sign part of the images, and directly let the model train the image without back- ground pixels to realize the recognition task. This paper constructs two CNN classification models with the same structure through comparative experiments. One model uses the segmented data set for training, and the other model uses the original data set for training. The performance of the two models is evaluated and compared to verify the optimization effect brought by the segmented data set. It is found that segment anything model can accurately cut most of the images in the data set. And the performance of the model trained by the segmented data set is better than that of the model using the original data set. Moreover, the optimized model can achieve high accuracy in less training times.

Keyword: road sign recognition, automatic driving, convolutional neural networks, complex background elimination.

1. Introduction

Road sign recognition is integral to autonomous driving, serving to inform vehicles of speed limits, direction guidance, and potential hazards. These signs hold the information necessary for vehicles to adjust their driving status and update their navigation routes continuously. Typically, a road sign recognition system consists of two specific components [1]. The first phase is detection, wherein a machine scans the input video or image using a computer vision algorithm to extract features pertinent to road signs. Once the machine identifies the preset features, it locates and bounds the pixels in the target area and transmits them to a Convolutional Neural Network (CNN) [2] for the identification phase. This latter phase employs a CNN, a deep learning model designed for grid-like data structures such as road sign images. After a convolution operation, the model learns the features and structures in the images and recognizes their content.

CNN, especially the latest R-CNN model, exhibits impressive performance in both detection and identification tasks [3]. The R-CNN model, using the Selective Search algorithm, divides the input

DOI: 10.54254/2755-2721/31/20230138

image into around 2000 candidate regions [3]. Each region undergoes feature extraction through a pre-trained CNN, preserving and classifying the detected regions. Yet, developing an accurate and well-performing CNN model necessitates feeding vast amounts of images for training. Most dataset images, including the GTSRB dataset utilized for training road sign recognition in this study [4], contain not only object pixels but also substantial background pixels. As background pixels vary due to different weather, time, positions, and shooting angles, they present complex challenges. During CNN training, the model may learn some features from the background, leading to overfitting, particularly with smaller datasets [5]. Furthermore, the confusion between background and foreground information may result in incorrect classification [6]. Therefore, acquiring a high-performing road sign recognition system demands high-quality training datasets. Recent developments by the Meta AI team resulted in the Segment Anything Model (SAM), a promptable segmentation system that extracts target objects at the pixel level, excluding background and other interfering pixels with zero-shot generalization [7]. This paper applies the SAM model to the GTSRB dataset in batches to eliminate confounding in all background pixels, thereby obtaining higher-quality training data for the CNN model, thus improving accuracy.

In the ensuing experiment, two identical structured CNN models, A and B, are established. Model A trains through the original dataset, while Model B trains through the optimized dataset, ensuring that the only difference lies in the training datasets. Following this, a series of evaluations and comparisons on the two models are conducted, observing their application effects through standard road sign images. The paper concludes by discussing the impact of the optimized dataset on the training model based on their differential performance.

2. Method

For the task of road sign recognition, the first step is to select a suitable dataset for training the model. Currently, the most widely used dataset is The German Traffic Sign Recognition Benchmark. It was originally created by researchers at the University of Marburg in Germany and published on Kaggle. The dataset contains more than 50000 pictures of different traffic signs, covering a variety of traffic sign categories, including 43 categories, including speed limit signs, prohibition signs, Warning sign, etc. Each image has corresponding label in- formation to indicate the category of traffic signs. The GTSRB data set has excellent structure and can be easily and quickly applied to model training. The data set is divided into a training set containing 39000 images, in addition to 5000 validation set images, for adjusting model parameters. And a test set containing approximately 12000 images to evaluate the performance and generalization ability of the model.

However, these images were collected from actual traffic scenes in Germany. Each images have a different resolution and lighting conditions. Some photos also have some occlusion, blurring, and complex background information, which corresponding to the realistic and complex road conditions. These complex photos may cause the model to extract non subject features during training, thereby affecting the overall accuracy of the model. As shown in Figure 1.

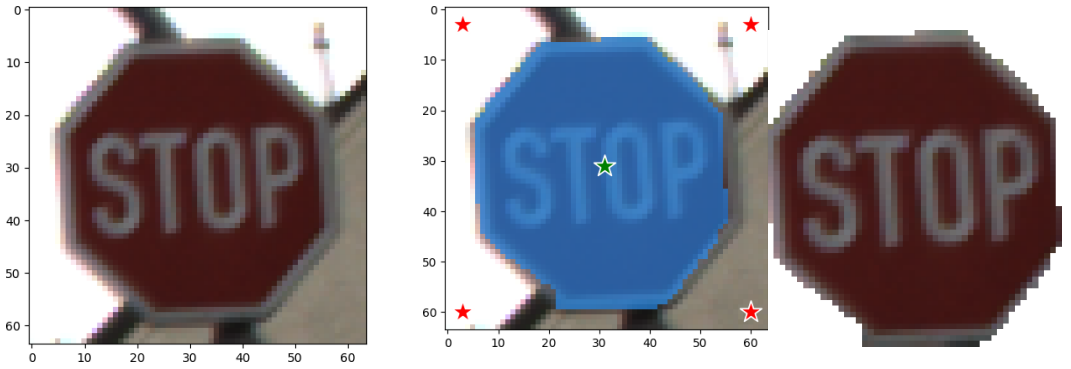
Figure 1. Images in GTSRB dataset [4].

2.1. Processing data set

Segment Anything model provides three methods to deal with images, which segment the objects in the coordinate position in the image by inputting points. Cut the items selected by the box through the box selection picture or input the text model to find and segment the objects in the picture that meet the text description through semantic understanding. This article will use the point method to segment the image.

DOI: 10.54254/2755-2721/31/20230138

There are two categories in the input points. One is the selection point, where the selected object is considered the target object and its pixels are retained. Another method is to remove points, where the selected object is considered a background element and will be removed. Due to the data set, the road sign is located in the center of the image and occupies most of the image position. The corresponding data processing in this article will achieve segmentation tasks by inputting 5 points. The main selection point is in the center of the image to select the road sign. The remaining four points are placed in the four corners of the image to remove background pixels. By using this method, As shown in Figure 2.



(a) Original Image (b) Input with 5 point (c) Output image

Figure 2. The processing of removing background (Photo/Picture credit: Original).

the entire data set was successfully brought in for processing, and the images of the entire data set were effectively separated. In the updated data set, the image content only contained the main landmark pixels and excluded unnecessary interference pixels.

2.2. Model establish

Upon processing the dataset, it is introduced into the CNN model to establish an identification system. This model is a fundamental and straightforward CNN model, mainly composed of five layers. Adjustments to the model's hyperparameters are made according to the data size and the tasks to be accomplished [8,]. The first layer is a convolution layer (Conv2D), employing a convolution kernel of size 3x3 for convolution operations, with 128 convolution kernels. This implies that it will produce 128 feature maps, using the ReLU function as its activation function. The second layer, MaxPooling2D, down-samples the feature map to reduce its dimension, thus decreasing the network's parameter count. This results in a smaller model with reduced computational complexity, while also mitigating the risk of overfitting [9].

Following this is a Flatten layer, which performs another dimension reduction operation, converting the two-dimensional output from the previous layer to one-dimensional. This transformation allows the subsequent Dense layer to process this data. The final output result requires passing through two Dense layers. The first Dense layer integrates 128 different types of features from the previous convolution and max pooling layers. This information is then output to the next layer via the non-linear activation function, ReLU. The last Dense layer accomplishes the classification task. The neuron count is set to 43, equivalent to the final output classes. The softmax function is used as the activation function, outputting the prediction probability of each class. Ultimately, the image class is predicted by the model through a comparison of these probabilities.

During the model compilation phase, Use a very common ADAM optimizer. It combines two random gradient descent algorithms, Adarad and Root Mean Square Propagation (RMSProp). This optimizer can effectively update each weight of the model.

DOI: 10.54254/2755-2721/31/20230138

2.3. Model traning

Before the training, the data needs to be further preprocessed, because the resolution of the images has been adjusted to 64x64 and converted to RGB format in the segment of processing the database. Next, only need to convert the image into a numpy array and normalize it [10]. The first step is to facilitate the subsequent mathematical calculation of the image through numpy. Normalization is to scale all pixel values to range [0-1] by dividing by 255, which is helpful for model convergence and training. Finally, the data will be feuded into the model for training. Each training will traverse the data set many times. After each epoch, the weights of neurons will be updated, and the loss function will be minimized. At the end of signal epoch, the model will be evaluated through the validation set. When the performance of a model declines, the training cycle can be ended to prevent overfitting [11,12].

2.4. Experiment

In this paper, we will construct two models described above by comparative experiment, construct them strictly according to the described parameters, and train them in the same way, so as to ensure that the two models are identical except for the training parameters. After that, the model will be evaluated in three aspects.

Performance metrics: accuracy is the most basic measurement. It can reflect the ability of the model to correctly detect images, but other accuracy rates, recall rates and F1 scores such as macro average and micro average can also provide more comprehensive information. Fusion matrix: because the GTSRB dataset contains 43 categories. In order to reflect the details of each category, the confusion matrix can be used to view which categories are correctly predicted by the model, which categories have errors in prediction, and which categories are mainly distributed in the wrong prediction.

The above methods can comprehensively evaluate the overall performance of a model, and then compare the two models according to the evaluated data to determine whether the optimization of the data set has a positive impact on the recognition system.

3. Result

3.1. Segment data set

This processing segmented over 50000 images from the entire data set without using an external Graphics Processing Unit. The process took a total of over 40 hours and achieved satisfactory results. Most of the images in the entire data set have successfully remained road sign parts and removed background pixels. But there are still a small number of images that have not been properly segmented. One is that too many background pixels are retained in the image without being completely removed, and the other is that excessive segmentation leads to the removal of the road sign pixels. Here, class 3 is randomly selected to make a rough estimation of the whole segmentation result. Among the 1829 images, 59 were not completely segmented, and 17 were over segmented, with a segmentation success rate of 95.84%. As shown in Figure 3.

(a) Successfully segmented (b) Non-completely segmented (c) Over segmented

Figure 3. Result of segment (Photo/Picture credit: Original).

DOI: 10.54254/2755-2721/31/20230138

3.2. CNN model evaluation

After many times of model training, the accuracy of the validation set at the end of each epoch was counted and observed, and the best training epoch relative to the two models was 5. The accuracy of the validation set of the model using the segmented data set was greater than that of the model using the original data set at the end of each cycle. At the end of the final model epoch, the validation accuracy of the model using the segmented data set was 89.60%, while the validation accuracy of the model using the original data set was 84.85%. As shown in Table 1 and 2.

Table 1. Training data using segmented dataset.

Epoch	Accuracy	Validation Loss	Validation accuracy
1	80.05%	0.6622	81.68%
2	94.35%	0.5384	86.33%
3	96.79%	0.6453	87.08%
4	97.69%	0.6655	87.52%
5	98.22%	0.6341	88.76%

Table 2. Training data using original dataset.

Epoch	Accuracy	Validation Loss	Validation accuracy
1	64.79%	0.7251	78.97%
2	88.59%	0.6895	79.76%
3	92.70%	0.5346	84.62%
4	94.90%	0.6153	85.87%
5	95.91%	0.6415	85.85%

After completing the model training, the classification performance of each class is evaluated through the Confusion matrix, and then the classification performance of the entire model is calculated through the average value. As shown in Table 3.

Table 3. Average evaluated values of two different model.

Model	Average precision	Average Recall	Average F1 score
Segmented	86.57%	83.86%	83.98%
Original	83.86%	80.94%	81.46%

4. Discussion

4.1. Result analysis

For the whole segmentation process, the result is very satisfactory, but some are not segmented correctly. After observation and summarize, images that are not correctly segmented always have the following characteristics. Picture taken under the condition of insuf- ficient light environment so the The overall brightness is low, which makes it difficult to distinguish the road sign from the background. The image with a low original pixel that causes the picture to be blurred, so that there is no strong boundary between the road sign and other objects, which will make it difficult for the model to judge correctly. In addition to the road signs to be identified, there are other road signs in the images. Because both of them have the same material and color, they will be judged as the same object by the segmentation model. In addition, there are some special cases, During the using of five input points to segmentation model, the points will coincide with some objects and make the wrong segmentation. If the point is on the icon in the road sign, the segmentation model will consider whether the user wants to segment the icon on the road sign or the road sign itself. Sometimes the model will misjudge the icon or number in the center of the road sign rather than the road sign itself. Also, when the road sign

DOI: 10.54254/2755-2721/31/20230138

is blocked by something and the input point is at the blocked thing, the model will segment the centre object instead of road sign.

After comparing various of CNN evaluation data, it is obvious that the model using data set after segmentation is better than the model with original data set. The first model outperforms the latter in terms of accuracy, recall and F1 score. And by observing the validation learning rate of each epoch, it can be observed that the model using segmented data sets reached a high-precision state earlier, which means that high-quality data sets can not only improve the accuracy of model prediction, but also shorten the training time and improve the efficiency of model training.

4.2. Limitation

Due to the limitation of equipment, this experiment processed a relatively small data set for deep learning. After processing, only a portion of images that were excessively segmented that completely losing valid information were manually removed. some images that have not been correctly segmented are retained and brought into the data set for subsequent model training. These images can be further processed in detail, but it still needs a lot of time for such a data set manually. In this experiment, a basic and simple CNN model was designed, and only small-scale experiments were carried out to adjust some of the hyper-parameters. Furthermore, the over fitting problem of the model after image segmentation is larger than that of the model with complex background, which needs to be focused on in the construction of the model. The model in this experiment may not be able to represent a wide range of CNN models, especially for those larger and more complex models. Although the experiment has obtained a high-quality CNN model using the segmented dataset, the impact of using the segmented dataset in a broader neural network model remains to be considered.

4.3. Future prospects

Currently, background removal is rarely implemented at the application level. It is almost impossible to remove the background accurately and automatically due to technical limitation. But with the help of segment anything model, this task can be easily completed. For those object detection tasks with serious background impact. Like, to detect and recognize some hollowed objects or objects with small area in the square bounding box, the learning efficiency and accuracy of the recognition model can be greatly improved by removing the background. But it also needs more accurate segmentation methods. It can combine a variety of input methods, such as input text for semantic understanding and achieve more accurate segmentation with the help of input more points and boxes. After the segmentation is completed, the image that is not correctly segmented can be further processed by manual operation to obtain a data set containing only the detected object.

5. Conclusion

Contrasting the model utilizing the original data, the model leveraging the segmented dataset displays higher accuracy. Owing to the correct segmentation of the majority of images, an effective dataset emerges that successfully eliminates background pixels interfering with classification results. However, further processing of the segmented dataset could yield even higher accuracy. Furthermore, this experiment also reveals that a model using a segmented dataset can achieve high accuracy with fewer training epochs. After background removal, the model's neurons are updated solely based on the object, indicating that dataset optimization can reduce model training time or simplify complex structured models.

Presently, this paper represents one of the scarce studies that enhance data models by removing background pixels from the entire dataset. It holds substantial reference value for CNN model optimization and precise improvement, particularly for tasks involving the recognition of objects with hollow structures such as sparsely leaved trees or meshes, or objects that occupy a small proportion in the recognition bounding box. The method can effectively eliminate the disturbance of complex background pixels on the model. Nevertheless, due to equipment and time constraints, this study exclusively employs the segmentation method of input points and does not conduct additional

DOI: 10.54254/2755-2721/31/20230138

processing on the dataset post-segmentation, potentially impacting the model's accuracy. The experiment also only used a comparatively simple CNN model. Future research could implement more thorough dataset processing and engage more complex neural network models for training and study.

References

[1] ALorsakul, A., & Suthakorn, J. (2007). Traffic sign recognition for intelligent vehicle/driver assistance system using neural network on opencv.

[2] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Commun. ACM, 60(6), 84–90.

[3] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation.

[4] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural Networks.

[5] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2015). Object detectors emerge in deep scene cnns.

[6] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE International Conference on Computer Vision (ICCV).

[7] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W. Y., Doll´ar, P., & Girshick, R. (2023). Segment anything.

[8] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. J. Mach. Learn. Res., 13, 281–305.

[9] Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. International Conference on Artificial Intelligence and Statistics.

[10] Zeiler, M. D., & Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks.

[11] Kingma, D. P., & Ba, J. (2017). Adam: A method for stochastic optimization.

[12] Prechelt, L. (2012). Early Stopping — But When?, pages 53–67. Springer Berlin Heidelberg, Berlin, Heidelberg.