Multimedia Tools and Applications (2021) 80:15161–15171
https://doi.org/10.1007/s11042-020-10455-x

Traffic-light sign recognition using capsule network

Xiaoxu Liu[1] · Wei Qi Yan[1]

Received: 12 February 2020 / Revised: 8 October 2020 / Accepted: 22 December 2020 /
Published online: 1 February 2021

Abstract
Automated driving gradually emerges as a real reality, but it still has to face various challenges, including sophisticated and volatile traffic conditions, human operating faults, etc. Amongst them, accurate understanding of traffic signs by using computer vision and deep learning methods has great significance for driving safety. In recent years, the advent of deep learning has made this issue much effectively. In this article, our major goal is to use deep learning models for conducting traffic-light sign recognition related to autonomous vehicles. As far as we know, this is the first time that the capsule neural network is employed as a method for scene understanding so as to effectively identify a class of traffic-light signs. Compared with the well-known convolutional neural networks, capsule networks diminish the demands for training datasets and tackle the spatial relationship much precisely.

Keywords Deep learning · Automatic driving · CapsNet · Traffic-light sign recognition

1 Introduction

With the development and popularization of artificial intelligence, automated driving have been one of the research challenges in the field of computer vision. Due to the complexity of traffic environment, the capability of traffic scene understanding has become a significant determiant of autonomous vehicles.

Scene understanding is a process of cognizing and inferring the environment based on spatial perception [14, 15]. In terms of autonomous vehicles, a scene is an environment in which a vehicle is located, which includes location, focus, event, and relationships between them [16, 20]. For autonomous vehicles to be driven safely and smoothly in the complex traffic environments, the perception and understanding of traffic-light signs in traffic scenes have the paramount importance. Therefore, throughout robust traffic-light sign detection and identification, autonomous vehicles are expected to be truly implemented.

✉ Wei Qi Yan
dcsyanwq@gmail.com

[1] Auckland University of Technology, Auckland 1010, New Zealand

Recently, owing to the development of deep learning paradigram, the performance of traffic-light sign recognition in unmanned driving can be ameliorated to further depth [1, 8, 32]. Besides, deep learning has the merits of transfer learning that a myriad of pretrained networks and public datasets have provided benefits for training traffic scenes. To understand the objects, scenes, and events in a video, deep neural networks emulate the abstraction from visual data and encode them using an effective representation [6, 7, 22]. Therefore, deep learning methods have unique advantages for traffic object recognition.

Although a rich assortment of attainments have been made in pattern recognition, CNN-based deep learning models have to face bottlenecks. Through our surveys, we observe that visual features of the next layer of CNN are the weighted sum of the features of the previous layer by using combinations. The activation functions of the previous layer are nonlinear based on multiplications and additions of the weights of neurons of the next layer. In this architecture, the positional relationship between the features of the next layer and the features of the previous layer becomes blurred. In order to solve this problem, CNN models expand receptive field of the next layer through pooling and convolution operations. Based on this machanism, we see that the focus of CNN is on detecting important features in the given image. As long as there exist visual objects in a feature map with strong stimuli, the spatial relationship of these objects will take effect.

Overall, CNN models neglect the orientation of object parts and the relevant relationship in an image, only heed the characteristics of a given image. Besides, though the pooling layer of CNN cuts down parameters and avoids overfitting, it discards the location information and spatial relationship of the visual object.

Pertaining to the recognition of traffic signs, correct recognition of traffic-sign arrows is very vital. Even if the structure of the arrows is similar, the directions of arrows that point in are diverse. However, owing to the black box characteristics of CNNs being lack of understanding of spatial relationships, it often leads to errors in sophisticated environments.

Therefore, we propose the capsule network for the task of traffic-light sign recognition so as to solve the orientation problem by using spatial relationships. The detailed information in traffic scene will be stored in the neural network, which will not be lost. These details encampuslate the exact position, rotation, thickness, and size of visual objects.

Regading the remaining parts of this paper, the literature review is presented in Section 2; our methods are delineated in Section 3, the results are demonstrated in Section 4, our conclusion is drawn in Section 5.

2 Literature review

In recent years, there have been a myriad of applications in machine learning, especially deep learning, to resolve the problem of traffic sign recognition. The current methods for traffic sign detection are originated from twofold, namely, colour-based methods and shape-based methods. For example, Kim et al. [11] combined colour and spatial transformations as well as the integrated network model to implent traffic sign recognition. Tran et al. [29] proffered HSV thresholding, geometrical information, and colour density of an image to identify traffic signs. In order that the traffic signs are easily to be viewed distinctly from surrounding environment, they are usually coated with reflective films having highly visible contrast, especially at night.

Just as the colour-based methods rely on specific colours, the shape-based methods are dependent on clear shapes and edages. The shapes have been applied to detect traffic signs via the algorithms of shape detection, that is, to find contours and implement the classifiers based

on the shapes [19, 24]. The shape-based methods avoid the reduction of recognition accuracy due to the influence of lighting conditions. In addition, shape detection narrows down the search from the entire image to a number of pixels. However, partially occluded, faded, and blurred traffic signs may make it tough for accurate detection, result in a low accuracy [31].

Currently, deep learning methods concentrate on the models such as YOLO, Faster R-CNN, and SSD to implemt traffic sign recognition. Müller et al. [18] propounded the structure of SSD and put forward TL-SSD, which accurately detect small or tiny objects. In TL-SSD, Inception v3 is accommodated to replace VGG as the basic network, which combines contextual information and local information for traffic sign detection and recognition. Han et al. [5] set forth Faster R-CNN, exterminated pooling layer in the original network VGG16, which raises the size of feature maps so that RPN is able to extract the features of small objects much accurately.

However, CNN-based traffic sign recognition always has the characteristics like a black box. Therefore, the changes of lighting conditions, colours of traffic signs, and other similar traffic signs in complicated environments have become determinnants that affect the accuracy. Moreover, both spatial and hierarchical relationships and rotation invariance are the chief problems that have to be faced by CNNs.

CNNs are always vague to the changes of image orientation or posture. Based on the attributes of layer-based squashing function, in this article, the capsule encodes the probability of detected object as the output vector. The state of the identified object is encoded into the direction specified by using the vectors. Therefore, the probability of object recognition remains the same, the direction has changes if the recognized object moves to a new position or state. Therefore, the CapsNets are particularly useful while dealing with multiple types of visual stimuli (e.g., position, size, orientation, speed, reflection, and texture, etc.).

Recently, CapsNets have been applied to autonomous vehicles. In addition to simple classification and recognition tasks [34], a series of dynamic tasks are also implemented by using CapsNets. For example, the centerline detector [10] scans traffic anomaly by combining the positions of autonomous vehicles, the positions of other vehicles, and spatial relationship with each other in the scene.

Besides, in order to solve the loss problem through using correlations between the traffic sign images measured by using max-pooling operations of the CNNs, the model for traffic speed prediction [12] is replaced by using max pooling with a dynamic routing algorithm in capsule network.

Ulteriorly, in traffic sign and congestion prediction [28], ReLU function is replaced by using Leaky ReLU function for the first two convolution layers of CapsNet, dynamic routing is simplified in a way that reduces loops. Moreover, the capsule layer applies squash function as a new activation function.

However, to the best of our knowledge, no research outcomes are available using the capsule network to implement traffic sign recognition. In this paper, traffic-light sign recognition is achieved by using CapsNet. We take the step of a breakthrough by using the CapsNets and feature vectors to represent the spatial relationship and attributes of visual objects, make predictions much interpretable based on a newly created model for traffic-light sign recognition.

3 Methodology

In order to deeply discover the merits of deep learning in traffic-light sign recognition, we detail a computable method in this section based on deep learning.

3.1 Traffic-light sign recognition

Regarding as recognizing visual objects, high-performance models can not only recognize independent objects but also understand the hierarchical structure. Therefore, we make full use of deep learning models so as to simulate the characteristics of our human brain. The semantic understandings are gained through layer-by-layer process and a stepwise abstraction of the feature map [9, 36].

If we emulate human cognition of traffic signs, the entire process of visual information needs to be understood. Our human has tactile, visual, and auditory information, which is similar to the idea of extracting visual features from scenes and deploying the basic layer for the purpose of ontology; our human brains mingle these features and make judgement from these available data. Ultimately, we obtain implicit semantics of the visual information through reasoning, which conforms to semantic expression and understanding [3, 4].

Therefore, spatial relationships between visual objects are beneficial for understanding advanced semantics. It is also one of the pivotal steps to use deep learning for analyzing traffic signs.

In deep learning, CapsNet is able to better model the hierarchical relationship of internal knowledge represented in neural networks. It utilizes poses to represent the relative relationship between three-bit objects and is numerically represented as a four-dimensional pose matrix. In this way, if these relationships are built into the internal representation of the visual data, it is easy for the model to associate the angle-shifted object with the previous object. Therefore, compared to the data required by CNNs, the CapsNet only needs to learn minor amount of visual data so as to achieve the most advanced results. In this sense, the capsule model is closer to the behavior of our human brain.

3.2 CapsNet

In deep learning, compared to the impressive CNNs or ConvNets, CapsNets have been promoted to solve the problems related to computer vision. For the purpose of mingling the relative relationships existing in visual objects, CapsNets treat visual object recognition as the probability of vectors, such as posture (i.e., position, size, direction), speed, etc. CapsNets mainly tackle internal representation, CNNs can not take consideration of the spatial hierarchy between simple and complex objects. CapsNets utilize various visual features of a class of objects in the unit of capsules composed of multiple neurons [13]. As the number of capsules with consistent output grows, the correct rate of object detection ramps up. The reason why CapsNets are able to identify visual objects by using the poses is that the spatial relationships are treated as one part of the scene. As a result, the CapsNet output appears to be more meaningful than a convolutional neural network in various cases [17].

Figure 1 shows the workflow of a CapsNet, which roughly delineates the working steps as matrix multiplication of the input vectors, the scalar weight of input vectors, the sum of weighted input vectors, and the vector-to-vector nonlinear transformation. The input vector of capsules comes from the output of three capsules at previous lay. The probability of the corresponding object is detected by using the capsules. The vector unveils the internal states of the detected object. These vectors are multiplied by using the corresponding weight matrix $\boldsymbol{W}$, which reflects the spatial relationship between the objects at the previous layer and the next layer of the deep learning model. After multiplying the weight matrix, we get the predicted position $\hat{u}_j$ of the object. Unlike ConvNets that apply backpropagation algorithms to update

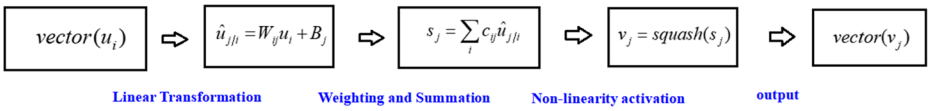

Fig. 1 The workflow of a CapsNet

weights, CapsNets take advantage of dynamic routing mechanisms to determine which capsules should be updated [33].

In comparison with the linear weighted summation of fully connected neural networks, the weighted summation of CapsNet S_j adds a coupling coefficient c_{ij} [25], which is

$$c_{ij} = softmax(b_i) = \frac{exp(b_{ij})}{\sum_k exp(b_{ik})}. \tag{1}$$

Meanwhile,

$$b_{ij} = b_{ij} + \widehat{u}_j \cdot v_j \tag{2}$$

where c_{ij} is coupling coefficients, b_{ij} is logarithmic prior probabilities that capsule i should be coupled to capsule j.

Another major improvement in the CapsNet is that the novel nonlinear activation function, which outputs a vector and normalizes the input vector to the unit length [25].

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{3}$$

where v_j is the vector output of capsule j and s_j is its total input.

3.3 Loss function

This project employs the length of a capsule vector to characterize the probability of a pattern. A margin loss function is provided for each capsule that symbolizes the traffic sign class,

$$L_c = T_c \max(0, m^+ - \|v_c\|)^2 + \lambda(1 - T_c)\max(0, \|v_c\| - m^-)^2 \tag{4}$$

where L_c represents the loss of category c, $T_c = 1$ if and only if there are traffic-light sign belonging to category c in images. In order to minimize the loss when a class of traffic-light signs do not appear, avoid compressing the length of the vector modulus of all capsules at the beginning of training, we set $m^+ = 0.9$ and $m^- = 0.1$. At the same time, we assign $\lambda = 0.5$. The total loss is the sum of the loss of each digital capsule.

In this project, the CapsNet is used in ten capsule outputs by using DigitCaps, the vectors of the non-target capsules are set to 0, and a 320D vector is employed as an MLP (multilayer perceptron) input. Finally, Euclidean distance between the obtained vector and the vector of the original input image is designated as the reconstruction loss

$$L_r = \sqrt{\sum\sum (I_{ij} - R_{ij})^2} \tag{5}$$

where I_{ij} represents pixel intensity at (i, j) on the original image, R_{ij} stands for the corresponding pixel intensity based on the reconstructed image. In summary, the total loss is

$$L_{total} = \sum_c L_c + \alpha \cdot \sum_r L_r \tag{6}$$

where α is 0.0006 in this project, L_c is the margin loss, and L_r is the reconstruction loss. The margin loss dominates the total loss.

3.4 The CapsNet for traffic-light sign recognition

In this article, we employ a CapsNet for our traffic-light sign recognition. The two most popular datasets currently in the traffic sign dataset are LISA and DriveU. However, due to the single shot angle of the sequences provided by these two types of datasets, they cannot present a sufficient number of classes of traffic-sign images for training, which are "Green Circle", "Green Left", "Green Negative", "Green Right", "Green Up", "Red Circle", "Red Left", "Red Negative", "Red Right", and "Red Up". Therefore, we take use of the richer dataset TL_Dataset available on the Internet. TL_Dataset contains more than 26,000 training images and 20,000 test images, we split these images into ten independent folders based on the classes. The size of the images in this dataset is 32 × 32, because the image size is small, which saves a plenty of training time. We selected 6000 high-quality pictures as our training set and test set at a ratio of 6:1. Figure 2a shows samples of the dataset, in order to enrich the dataset, we enhanced the dataset by using data augmentation in this project, including shifting the images horizontally and vertically by using a magnitude of 0.2 which is shown in Fig. 2b [21, 23, 27, 35].

Thus, our traffic-light sign recognition has been altered from a neuron output to a set of vectors. Each capsule that recognizes the substructure makes the information hidden in the original graphics and achieves a high fidelity if the number of bits becomes smaller. This fidelity is directly evolved from the complex structure. Throughout reconstructing the original images, the model has achieved the same perception using the changes of network structure after the perspective changes. As shown in Fig. 3, the CapsNet for traffic-light sign recognition in this project consists of an input layer, a convolutional layer, a primary capsule, a digital capsule, and an output layer.

There are 5000 training images in this project having the size 32 × 32. One image has three dimensions to store information in multiple colour channels. Each pixel is represented as an integer number between 0 and 255, stored in a 32 × 32 × 1 matrix [13, 14]. The brighter the colour of each pixel, the larger its value.

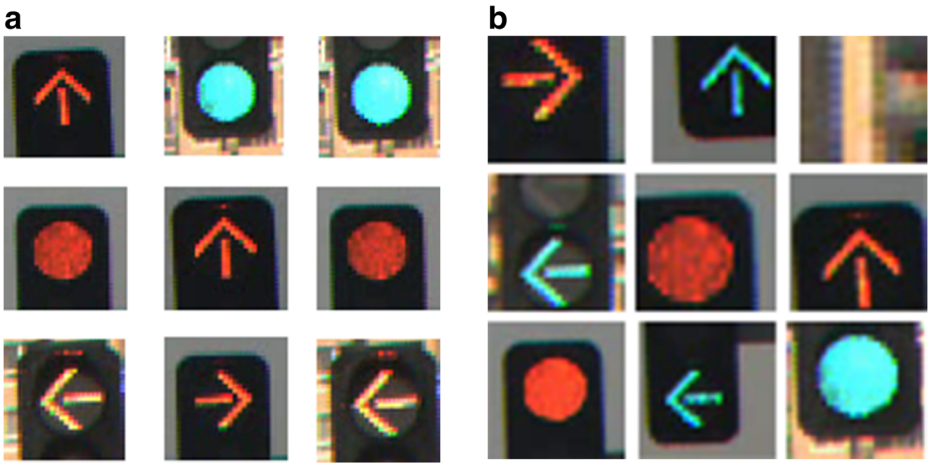

Fig. 2 Dataset and its augmentation

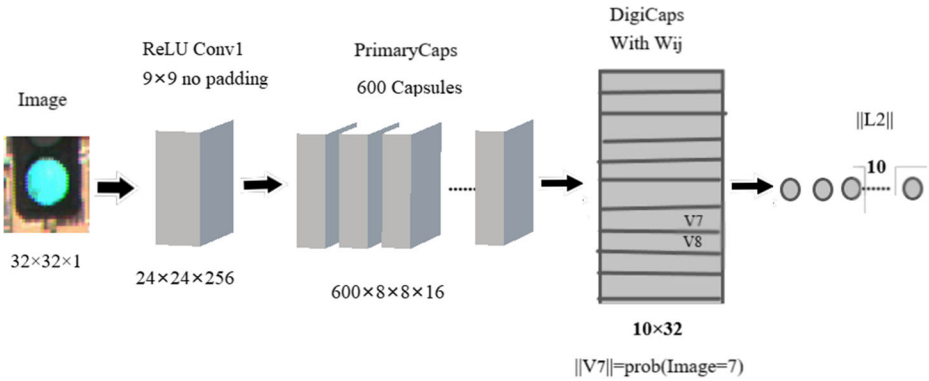

Fig. 3 The network structure for traffic-light sign recognition

In this proposed network, the layers connected to the input layer are the convolution layer and capsule layer. The convolution layer extracts basic features from the input image, e.g., edges, corners, motif, parts, etc. We also utilize ReLU to cope with and output the input values. In order to find more complex information from these visual features. We provide a convolutional layer as the primary capsule layer, because we need to operate on the 256 feature maps output by the previous layer, we take use of a $9 \times 9 \times 256$ convolution kernel instead of a $9 \times 9 \times 1$ convolution kernel. Since the main capsule layer is especially designed for traffic sign recognition, we utilize a special type of activation function called squeezing function to normalize the size of the vector in order to avoid gradient vanishing during model training. The next layer is a digital capsule layer, and its output is ten 32D vectors; each vector corresponding to a traffic sign category is the final predicted result.

The chief optimization method of this network is to update the weight $\boldsymbol{w}$ based on the backpropagation algorithm, a dynamic routing algorithm between the primary capsule layer and the digital capsule layer is accommodated to update the values of b_{ij} and v_j as addressed in Section 3.2.

In addition, this article presents an additional image structure to train the model. The reconstructed structure makes use of the output of DigitCaps to reconstruct the original image. In this way, the network is utilized to store image features. The reconstructed network employs three fully connected layers, including two ReLU layers and one sigmoid layer.

In order to evaluate a more accurate and robust detection model, accuracy (ACC), precision, and recall are exploited in this article to evaluate the performance of this model. In Table 1, p, i.e., $TP + FN$ shows the acutal numbers of positive samples labelled in the dataset, p', i.e., $TP + FP$ indicates the predicted number of positive samples using a classifier and this dataset. Similarly, n, i.e., $FP + TN$ stands for the actual numbers of negative samples in the dataset, and n',i.e., $FN + TN$ refers to the predicted number of negative samples using a classifier and the dataset.

The accuracy represents the percentage that each class is correctly identified. According to the fact, the accuracy is the ratio of the correctly classified objects to the total number of objects. The mathematical expression of accuracy is shown as eq. (7).

Table 1 Confusion matrix for the classification

Actual		Actual	
Predicted		p	n
Predicted	p'	TP	FP
	n'	FN	TN
Total		P	N

$$\mathrm{ACC} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \tag{7}$$

Precision refers to the number of correct positive predictions. The calculation is shown as eq. (8).

$$\mathrm{Precision} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \tag{8}$$

Recall is the true positive rate (TPR), it is defined as the percentage of positive cases which is shown as eq. (9).

$$\mathrm{Recall} = \mathrm{TP}/\mathrm{P} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \tag{9}$$

4 Results

From the experimental results of traffic sign recognition, we see that the proposed model is able to successfully identify color differences and and classify the class of a traffic sign. Figure 4 shows the test images and the prediction results, the bar charts on the right side of each image unveil the corresponding probabilities of 10 classes that the sign has been correctly recognized by using this model. This project applies average accuracy and average loss rate to measure the performance of the proposed model, which is represented in Fig. 5 as an accuracy curve and loss curve.

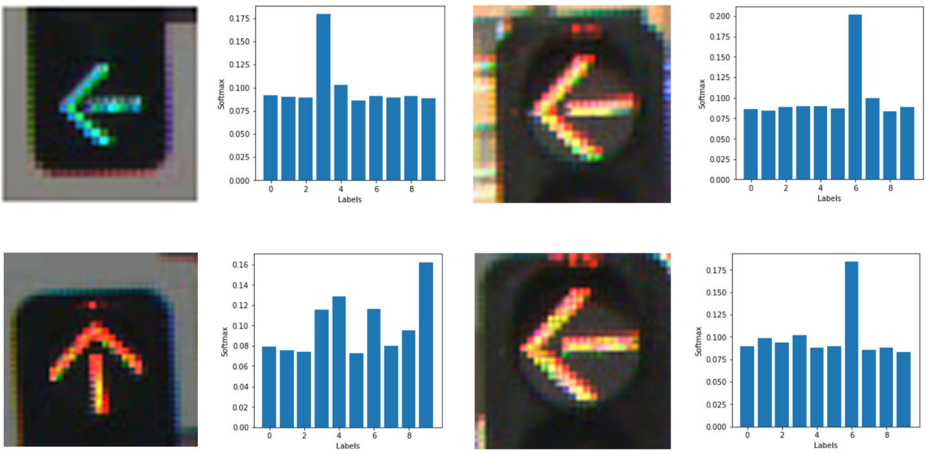

Fig. 4 The results of traffic-light sign recognition

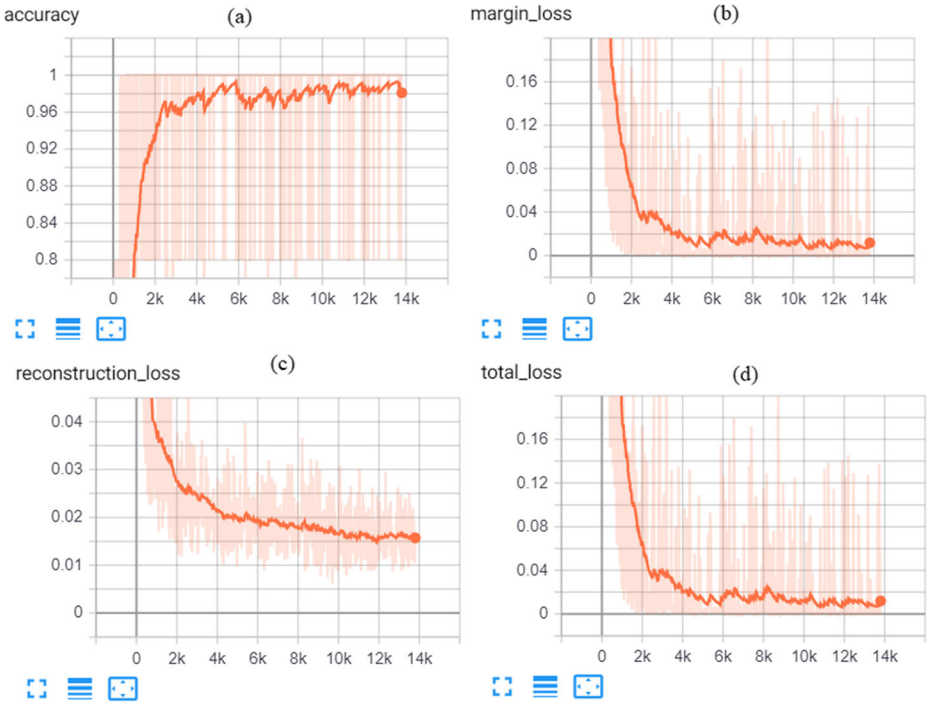

Fig. 5 Accuracy and losses of traffic-light sign recognition

Both accuracy and loss are based on our datasets. Ideally, the accuracy is 1.0 at all levels. The best average accuracy for this project by using the Capsule network is 98.72% and F1-score is 99.27%. At the same time, the precision and recall are 90.83% and 19.6%, respectively. Meanwhile, because the CapsNet is the neural network that allows multiple classifications to coexist, this project does not utilize the traditional cross-entropy loss instead takes advantage of the margin loss, which is shown as Fig. 5b.

Moreover, a robust model needs to have the ability to be reconstructed. If the model can be reconstructed, that means, it has an excellent capability of representation. In this project, our reconstructed loss result is shown in Fig. 5c.

We mingle the margin loss and the reconstruction loss to obtain the total loss value as shown in Fig. 5d. Table 2 shows the accuracy of multiple CNN-based methods if TL_Dataset is employed to compare the precision and recall. The accuracy and precision of our model are 0.89% and 1.4% higher than the other three methods.

Table 2 Comparisons of multiple methods

Methods	Evaluation Metrics		
	Accuracy	Precision	Recall
Tsoi, et al. [30]	97.21%	87.79%	15.63%
Behrendt, et al. [2]	95.94%	84.91%	32.89%
Sermanet, et al. [26]	97.83%	89.43%	35.03%
The Proposed Method	98.72%	90.83%	19.6%

5 Conclusion

In this paper, our contributions are:

(1) The proposed CapsNet is employed for traffic sign recognition.
(2) This paper resolves the misunderstanding problems due to use CNN in traffic sign recognition and the loss of max pooling.
(3) By using vectors to represent visual features, the posture information such as position and shape of the input image and the spatial relationship are retained, the network has the merit of robustness and generalization.

The goal of this project is to exploit CapsNets in traffic sign recognition, which are emerging in the field of deep learning for understanding traffic scenes. By adjusting the parameters and weights, the CapsNet has been applied to solve the problem of traffic sign recognition in this article. Throughout the dynamic routing mechanism, we believe the CapsNets are promising to resolve the problems of complex scene understanding.

References

1. Bach M, Stumper D, Dietmayer K (2018) Deep convolutional traffic light recognition for automated driving. Intelligent Transportation Systems, 851–858
2. Behrendt K, Novak L, Botros R (2017) A deep learning approach to traffic lights: Detection, tracking, and classification. IEEE International Conference on Robotics and Automation (ICRA), Singapore, 1370–1377
3. Bernstein D (2010) Essentials of psychology. Cengage Learning, 123–124
4. Brefczynski-Lewis JA, Lewis JW (2017) Auditory object perception: A neurobiological model and prospective review. Neuropsychologia, 223–242
5. Han C, Gao G, Zhang Y (2018) Real-time small traffic sign detection with revised faster-RCNN. Springer Multimedia Tools and Applications. 78
6. Husain F, Dellen B, Torras C (2017) Scene Understanding Using Deep Learning. Academic Press, 373–382
7. Jeon HS, Kum DS, Jeong WY (2018) Traffic scene prediction via deep learning: Introduction of multi-channel occupancy grid map as a scene representation. IEEE Intelligent Vehicles Symposium (IV), 1496–1501
8. John V, Yoneda K, Qi B, Liu Z, Mita S (2014) Traffic light recognition in varying illumination using deep learning and saliency map. Intelligent Transportation Systems, 2286–2291
9. Kheradpisheh S, Ghodrati M, Ganjtabesh M et al (2016) Deep networks can resemble human feed-forward vision in invariant object recognition. Nature Sci Report 6:32672
10. Kim M, Chi S (2019) Detection of centerline crossing in abnormal driving using CapsNet. Journal of Supercomputer, 189–196
11. Kim H, Park J, Jung H (2018) An efficient color space for deep learning based traffic light recognition. Advanced Transportation, pp.1–12
12. Kim Y, Wang P, Zhu Y, Mihaylova L (2019) A capsule network for traffic speed prediction in complex road networks. Sensor Data Fusion: Trends, Solutions, Applications (SDF), 1–6
13. Kwabena M, Felix A, Abra A, Edward Y (2019) Capsule networks – A survey. Journal of King Saud University - Computer and Information Sciences, 1–16
14. Lewicki MS, Olshausen BA, Surlykke A, Moss CF (2014) Scene analysis in the natural environment. Frontiers in Psychology, 199
15. Malcolm GL, Groen I, Baker CI (2016) Making sense of real-world scenes. Trends in Cognitive Sciences, 843–856
16. Morris T (2014) Computer vision and image processing. Palgrave Macmillan
17. Mukhometzianov R, Carrillo J (2018) CapsNet comparative performance evaluation for image classification. Computational Intelligence and Security, 1–14
18. Müller J, Dietmayer K (2018) Detecting traffic lights by single shot detection. Intelligent Transportation Systems, 266–273

19. Nandi D, Saif AS, Prottoy P, Zubair KM, Shubho SA (2018) Traffic sign detection based on color segmentation of obscure image candidates: a comprehensive study. Int J Mod Educ Comput Sci 10(6): 35–46
20. Park H, Jang S, Jeong H, Ha Y (2019) Roadway image preprocessing for deep learning-based driving scene understanding. IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 1–4
21. Peixinho AZ, Benato BC, Nonato LG, Falcão AX (2018) Delaunay triangulation data augmentation guided by visual analytics for deep learning. SIBGRAPI Conference on Graphics, Patterns and Images, 384–391
22. Qiao K, Chen J, Wang L, Zhang C, Zeng L, Tong L, Yan B (2019) Category decoding of visual stimuli from human brain activity using a bidirectional recurrent neural network to simulate bidirectional information flows in human visual cortices. Frontiers in Neuroscience, 692
23. Qu H, Zhang L, Wu X, He X, Hu X, Wen X (2019) Multiscale object detection in infrared streetscape images based on deep learning and instance level data augmentation. Applied Sciences, 553–565
24. Saadna Y, Behloul A (2017) An overview of traffic sign detection and classification methods. Multimedia Information Retrieval, 1–18
25. Sabour S, Frosst N, Hinton GE (2017) Dynamic routing between capsules. International Conference on Neural Information Processing Systems, pp.3859–386
26. Sermanet P, LeCun Y (2011) Traffic sign recognition with multi-scale convolutional networks. International Joint Conference, 2809–2813
27. Stivaktakis R, Tsagkatakis G, Tsakalides P (2019) Deep learning for multilabel land cover scene categorization using data augmentation. IEEE Geosci Remote Sens Lett 16(7):1031–1035
28. Tampubolon H, Yang C, Chan S, Sutrisno H, Hua K-L (2019) Optimized CapsNet for traffic jam speed prediction using mobile sensor data under urban swarming transportation. Sensors, 5277
29. Tran T, Pham C, Phuoc N, Duong T, Jeon J (2016) Real-time traffic light detection using color density. IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia) 1–4
30. Tsoi T, Wheelus C (2020) Traffic signal classification with cost-sensitive deep learning models. IEEE International Conference on Knowledge Graph (ICKG), 586–592
31. Wali SB, Abdullah MA, Hannan MA, Hussain A, Samad SA, Ker PJ, Mansor MB (2019) Vision-based traffic sign detection and recognition systems: Current trends and challenges. Sensors (Basel), 2093
32. Wu N, Fang H (2017) A novel traffic light recognition method for traffic monitoring systems. Asia-Pacific Conference on Intelligent Robot Systems, 141–145
33. Zhang F, Wang Y, Ye M (2018) Network traffic classification method based on improved capsule neural network. International Conference on Computational Intelligence and Security, 174–178
34. Zhang Z, Zhang D, Wei H (2019) Vehicle type recognition using capsule network. Chinese Control and Decision Conference, 2944–2948
35. Zhang Y, Li J, Guo Y, Xu C, Bao C, Song Y (2019) Vehicle driving behavior recognition based on multi-view convolutional neural network with joint data augmentation. IEEE Trans Veh Technol 68(5):4223–4234
36. Zhao Z, Zheng P, Xu S, Wu X (2019) Object detection with deep learning: A review. Neural Networks and Learning Systems, 3212–3232