Vehicle-Related Scene Segmentation Using CapsNets

Xiaoxu Liu, Wei Qi Yan, Nikola Kasabov

Auckland University of Technology, Auckland, 1010 New Zealand

Abstract— **Understanding of traffic scenes is a significant research problem in computer vision. In this paper, we present and implement a robust scene segmentation model by using capsule network (CapsNet) as a basic framework. We collected a large number of image samples related to Auckland traffic scenes of the motorway and labelled the data for multiple classifications. The contribution of this paper is that our model facilitates a better scene understanding based on matrix representation of pose and spatial relationship. We take a step forward to effectively solve the Picasso problem. The methods are based on deep learning and reduce human manipulation of data by completing the training process using only a small size of training data. Our model has the preliminary accuracy up to 74.61% based on our own dataset.**

Keywords— CapsNet, scene segmentation, deep learning, traffic scenes

I. Introduction

With the advancement of autonomous driving technology, the understanding of traffic scenes has become a significant research problem in the purview of computer vision and also a hot topic in the matter of artificial intelligence. The so-called scene is a combination of multiple views of visual objects in an environment. Scene-based feature representation has been considered as a more powerful procedure to interpret image scenes. Scene classification, scene segmentation, and object detection are all considered as important tasks in scene understanding [1, 2, 41].

Compared with indoor [3, 4] and static scene understanding [5], the task of understanding vehicle-related traffic scenes is much more difficult, owing to the accuracy of prediction that will be interfered by more determinants. Firstly, realistic traffic scenes are extremely complex, there are often a myriad of objects in the scene, the problem of occlusion between objects is ubiquitous. Moreover, because vehicle-related scenes are short in a dynamic environment, the same objects often turn up in the scene from different perspectives. At the same time, it is extremely tough to learn various perspectives of the same object using a small dataset [6].

Although there have been many solutions by using various computer vision and machine learning algorithms, deep learning has offered compelling improvement in relevant fields. Consequently, a raft of computer vision problems including scene segmentation have taken into consideration deep learning technology [7, 8, 9, 10, 11].

The layer-by-layer processing mechanism of deep learning, the combination of linear and nonlinear characteristics imitate to a degree of the cognitive process of our human brain, and gradually extract visual features, so that intricate information in traffic scenes is better represented. Thus, the features from deep learning algorithms enable the proposed model to understand the high-level semantics of traffic scenes [12, 13].

In this project, for understanding entire traffic scene, in lieu of a single or multiple objects, we apply semantic segmentation to segment the complete scene into multiple classes. The deep learning models currently are primarily convolutional neural networks (CNNs), such as U-Net [14], SegNet [15], and RefineNet [16]. These CNN-based models have gained epochal accomplishments in segmentation resolution and convergence speed, but they still fail to understand the orientation of components and relevant relationships in the space which only takes account of the characteristics of a given test image. In addition, though the pooling layer of CNN cuts off parameters and prevents over-fitting, it discards the location information of the image [17, 18]. To the best of our knowledge, there are few scene segmentation models dedicated to integrating spatial information with pose information.

Thus, we propose capsule network as a basic framework to solve the above problems. A capsule brings a new component for deep learning to better model the hierarchical relationship of internal knowledge representation in neural networks. The capsule network (CapsNet) imitates the gradual cognitive process of our human brain [19, 20]. From visual information received through our human eyes, our brain parses the hierarchical representation of the objects, matches the relationship between the learned patterns and the knowledge is stored in our brain [21, 22]. The CapsNet-based models are mostly accommodated for visual object classification, which takes use of the advantages of CapsNets to achieve scene segmentation. Moreover, in order to better represent visual features, this model employs a matrix instead of a vector to express the CapsNet-based models. The preferred model combines spatial and location information based on matrix representation to express the characteristics of the visual objects, and improves the accuracy of scene segmentation through rigorous logical relationships. In the dataset related to Auckland traffic scene that we have collected and labelled, the proposed model outperformed the traditional CNN model.

The remaining parts of this paper are organized as follows. Literature review is delineated in Section II, our method is explicated in Section III and the results of this study are demonstrated in Section IV. Conclusions are drawn in Section V.

II. Literature review

Understanding of traffic scenes involves perceiving and representing surrounding environment of a vehicle. Currently, automotive environment models combine a wealth of complementary representations, each of which focuses on a specific aspect of the traffic scene. Considered camera-based environment perception, multitask model presents an effective method that is able to generate complementary representations in an integrated manner. Moreover, pertaining to understanding dynamic environments such as traffic environments, current predictions for most objects are based

978-1-7281-8579-8/20/$31.00 ©2020 IEEE

on motion features extracted from the trajectories of moving objects [23]. The trajectory feature contains more semantic information, but the accuracy of trajectories deteriorates the performance of crowded scene analysis. In extremely crowded areas, tracking algorithms could fail and generate inaccurate trajectories.

Therefore, a traffic environmental model [24] was generated based on a single CNN. The model uses a shared encoder stage with a specific decoder for road segmentation and target detection to estimate the direction of the detected object based on analytical geometry. Therefore, the model copes with spatial information of dynamic objects and the space of regions obtained from road segmentation. Compared with the dedicated model for each individual task, even if based on low-cost embedded systems, redundant calculations of feature maps are avoided, resulting in fast reasoning.

However, due to the complexity of actual traffic environment, it is impossible to understand a series of dynamic activities of the object by only relying on spatial information. In order to solve the occlusion problem of complex traffic scenes, deeply understand object motion information an end-to-end deep architecture of convolutional DLSTM (Con-vDLSTM) was developed for crowded scene understanding [25]. Taken advantage of this semantic representation of CNN and the memory states of LSTM to effectively analyse both crowded scene and motion information, the existing solutions for LSTM-based crowded scene deeply explore temporal information and are claimed to be "deep in time". Con-vDLSTM not only probes deep temporal information but also exploits the spatial and temporal information in a unified architecture so as to achieve deep in spatial and temporal domain.

For understanding the entire traffic scene, in lieu of a single or several objects, semantic segmentation is usually applied to separate the complete scene into regions. The earliest semantic segmentation model was FCN (fully convolutional network) which does not involve fully connected layers in image-related classification [26]. However, because the resolution generated by using the encoder module has 1/32 original inputs, it is difficult for the decoder to restore perfectly the original resolution. The IoU of fully convolutional network based on the PASCAL VOC 2011 and 2012 datasets accomplished accuracy of 62.7% and 62.2%, respectively. In [27], skip connections are added in the first layer for improved accuracy.

Subsequently, the full convolution structure was improved by fleshing out the capacity of the decoder module [14]. This U-Net structure encompasses a contraction path that captures context and a symmetric extension path that enables precise positioning. The basic convolution module was substituted with a residual module for stacking [27, 28]. This residual module encapsulates skip connections inside the module, meanwhile, retains the same skip connections between the corresponding feature maps of the encoder and decoder as U-Net. This method is able to converge the network faster which is able to be applied to deeper network structures. Thus, the use of dense blocks is proposed which conforms to the U-Net structure, the attributes of dense blocks make them better and are suitable for semantic segmentation because they naturally carry skip connections and multiscale supervision. These dense blocks are easily to be deployed because they bring in the low-level features gained from the pre-order layer, including the high-level features obtained from the subsequent layers, thereby achieved more competent features. At present, the most advanced technology combines U-Net and EfficientNet as a traffic scene segmentation network, the test result based on the IDD dataset is up to 62.76% accuracy.

The FCN model was designed with deconvolution layers and multiple shortcut connections, notwithstanding the segmentation graphs are still rough. Therefore, SegNet [15] has more shortcut connections. Unlike FCN, which directly replicates the characteristics of encoders, SegNet replicates the indices generated by adding the max-pooling layer. This makes SegNet more efficient than FCN.

Basically, the scene segmentation model, downsampling the feature map, sets the same filter size to flesh out the receptive field of the network based on the input. This approach is indeed more competent than simply increasing the filter size, but it will result in a reduction in spatial resolution. Therefore, numerous models replace the last few pooling layers with dilation rates gradually expanding convolutional layers [29] to minimize the loss of spatial details and eke the receptive field.

In addition, the pyramid scene parsing network [30] was set for the popular pyramid pooling structure. Its significance is that it presents clues to the distribution of segmentation classes. The pyramid pooling model captures this information by applying a pooling layer of large-sized cores.

After compared the segmentation performance of each model under the premise of using the same dataset and operating environment, IoUs of FCN, U-Net, SegNet and PSPNet are 56.19%, 74.19%, 70.10% and 71.67% [31]. U-Net achieved its best performance with splicing feature vectors, encoding-decoding structure, and elastic deformation for data augmentation.

However, all these models utilized pooling layer and weight sharing. Although a wealth of models have improved the pooling layer so that it retains a part of the location information, these methods raise the workload on the network and the location information that it retains is incomplete.

Recently, in order to resolve the loss due to the correlation between the traffic data measured by using the sparsely placed sensors in max pooling operations of the CNNs, there have been a few models which replaced the max pooling with a dynamic routing algorithm of the capsule network [21, 32]. Capsule network is able to capture the part-to-global relationships from the dataset.

SegCaps [33] were proposed to limit the max pooling of convolutional neural networks so as to preserve detailed information. SegCaps were proposed to extend the concept of convolution capsules through locally connected routes and set the new concept related to deconvolution capsules. In addition, SegCaps has been developed to the mask reconstruction so as to reconstruct the positive input class, which reflects a strong binary segmentation if the parameter space is notably diminished [34]. The average dice score of SegCaps has 0.030% growth which is better than U-Net.

The capsule network [21] was initially employed for character classification. A capsule is a group of artificial neurons that perform intricate calculations based on inputs, which encapsulates its calculation and results in a vector. Each capsule has the pose information of the objects in the scene. If the relative position or posture of the object changes, the direction of the output vector will also be altered accordingly. Therefore, the input and output of a capsule are equally variable [34].

CapsNet is comprised of multiple functional layers. Each capsule in the main capsule layer of the first layer is responsible for receiving a part of the receptive field as input and detecting its posture. The capsule encapsulates its posture and other information into a vector output, harnesses a dynamic routing mechanism to transmit the output to the corresponding parent node in this layer.

In addition, CapsNet applies an additional structure to train the model. The structure makes full use of output of the activities in DigitCaps to reconstruct the original image. In this way, the network is utilized to remember more image features. The reconstructed network employs three fully connected layers, including two ReLU layers and a sigmoid layer [21].

The workflow of CapsNet describes the working steps as matrix multiplications with the input vectors, scalar weight of input vectors, the sum of weighted input vectors, and vector-to-vector nonlinear transformation. The input vector of the capsules comes from the output of three capsules at low level. The probability of corresponding object is segmented by using the low capsules. The vector reflects internal states of the detected object. These vectors are multiplied by using the corresponding weight matrix, which reflects the spatial relationship between the objects at low and high levels. After multiplying the weight matrices, we get the predicted position of the object. Unlike convolutional neural networks that apply backpropagation algorithms to update weights, CapsNets take advantage of dynamic routing mechanisms to determine which high-level capsules should be promoted [35].

In comparison with the linear weighted summation of fully connected neural networks, the weighted summation of CapsNet [36] S_j adds a coupling coefficient c_{ij} as

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

Meanwhile,

$$b_{ij} = b_{ij} + \hat{\mu}_{j|i} \cdot v_j \tag{2}$$

where c_{ij} is coupling coefficients, b_{ij} is logarithmic prior probabilities that capsule i should be coupled to capsule j. Another major innovation in CapsNet is nonlinear activation function, which outputs a vector and normalizes the input vector to the unit length [37].

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

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

Different from the existing deep learning methods, in this paper, our contribution is that the proposed model based on CapsNets offers better scene understanding based on pose and spatial relationship of the objects, owning to the attributes of the CapsNets [21].

III. Our Methods

Even if traditional capsule network and routing algorithm are much simpler than the convolutional neural network and back propagation algorithm in the training set and network structure, they still require a large amount of memory and running time. Under normal circumstances, using U-Net to achieve semantic segmentation requires approximately 3,100k parameters, and InvertedNet needs 3,141k parameters. After the child is routed to the coefficients of the *parents* of the next layer, the output is stored in the given layer by using intermediate representations. When this operation occurs between each parent and each possible child, the additional storage space required is the batch size of a given layer multiplied with the number of capsule types in that layer, so that the number of parameters required will inevitably exceed the control range. Thus, we dissolve the problem of memory allocation by expanding the network structure and ameliorating the routing algorithm [38].

We use U-Net as the main structure of the model but replace neurons with capsules. The children in our proposed model are routed to their parents only within the local kernel of the defined space. In addition, the transformation matrix is shared for a member of the grid within the capsule type, but not between the capsule types. Besides, in order to compensate for global connectivity loss caused by routing, we extend the capsule network by using a "deconvolution" capsule, which applies a convolution operation transposed by using a routing protocol. Furthermore, through the proposed deep convolution deconvolution architecture, we retain global context information, meanwhile, tremendously cut off the number of parameters in the network, alleviate the memory burden, and attain the most advanced the results. Our capsule net is comprised of pose information and location information.

Different from CNNs that scalars represent the features of the objects, CapsNet is well known as a neural network of vector representation. If a lower-level capsule selectively agrees on its parent capsule, the capsule outputs a vector. The prediction of the parent capsule is consistent with the actual output of the parent capsule which will affect the connection between the lower capsule and the parent capsule [39]. CapsNet learns the dint by dynamic routing to calculate the matrix $T_{i\to n}$ of size $N \times N$ which transforms the child vector v_i of size N to the parent vector v_n of size N. At the same time, there is a new method that applies vector P_i which describes the post and location of an object for feature representation [21, 34, 38].

In order to optimize the performance of this model as much as possible, take dual routing [38] as a reference, a matrix is applied to combine pose matrix and location matrix to represent features. Through combining with the transformation parameter matrix, the pose vector [38] P_i of the child i is converted into the pose vector $P_{i\text{-}n}$ of the parent n which is defined as

$$P_{i-n} = P_i T^P_{i-n} \tag{4}$$

where T^P_{i-n} is the transformation matrix of pose, image coordinates x, y are combined with each T^P. The final parental pose matrices P_n and location information are the combination of all their transformation matrices which are defined as [38]

$$P_n = Psquash(\textstyle\sum_i \alpha_{i-n}\, P_{i-n}) \tag{5}$$

where α_{i-n} is the weighting factor defined in the routing algorithm. The nonlinear function [38] $Psquash(\cdot)$ is defined as

$$\text{Psquash(P)} = \frac{P}{\max abs(P)} \qquad (6)$$

The dynamic routing mechanism defines $c_{i \to n}$ through the cross-correlation iterative optimization strategy between the child and parent vectors. Based on this concept, we express the pose features [38] that have been processed as

$$c_{i-n} = \langle P_{i-n}\,, P_n \rangle_F \qquad (7)$$

where $\langle \, , \rangle_F$ denotes the Frobenius inner product. Finally, the weighting factors $\alpha_{i\text{-}n}$ for each child are calculated by applying the sigmoid function to $c_{i \to n}$.

For each pixel $(x,\ y)$ or intermediate layer in the input image, we define a set of capsules $P_i\ (x,\ y)$. In addition, a convolution kernel with the size $N \times N$ is proffered for the pose vector P_i in order to integrate the local neighborhood information into the network. The predicted multilabel subdivision L at each position (x, y) corresponds to the index of the last activated capsule [38]

$$L(\text{x}, \text{y}) = arg_i \text{max} \left(\|P_i(x, y)\|_F\right). \qquad (8)$$

The evaluations of the proposed network are based on our own dataset which was collected from Auckland highways by using a vehicle camera.

IV. Results and Analysis

We have used the GTX 1660 Ti GPU to train a model and 16.00G memory so as to accomplish file storage and also used MATLAB 2019a and TensorFlow 2.0.

Our dataset consists of 200 Auckland motorway images with a resolution of 1024 × 1024 as shown in Fig. 1. The segmentation labels as the ground truth were manually annotated including vehicle, sky, tree, building, road, and traffic sign, leading to six classes of labels. Therefore, there are 1,200 training images for our segmentation model. We split the dataset into a training set and a test set in a ratio 3:1. In order to save memory and time, we resize all data to a resolution of 128×128 pixels. In addition, we also perform intensity preprocessing on the dataset. In order to make the pixels of each image within interval [-1.0, 1.0], we resized the input image. We conducted data enhancement based on the dataset, transformed it with space (translation, scaling, rotation, elastic deformation) and intensity (translation, scaling of the input image) to avert overfitting.

As shown in Fig. 2, visually, the capsule model in semantic segmentation exports the results such as buildings, sky, roads, trees. According to the experimental results in Table I, the performance of the proposed model in sky and building is notably excellent, 96.06% and 96.82% accuracy rates, respectively. Secondly, the segmentation results of this model based on the two segmentation objects, roads, and trees, are also creditable, 79.67% and 86.31%, respectively. Fig. 3. shows the training loss of our model. As the number of training iterations rises, the loss gradually diminishes, which converges after the number of iterations reaches to 10,000. The training loss achieves 0.014 after 30,000 epochs.

In order to ensure that our model performs well, we accomplish the same experiments on U-Net and SegNet, which are considered excellent in semantic segmentation. In order to ensure a fair comparison between the capsule network and CNN, we exchange the deconvolution operations of U-Net and SegNet with linear upsampling, cut down the number of intermediate convolution outputs to 16, the number of levels to 4. The number of parameters is as same as the model we proposed. At the same time, we take use of the pixelwise softmax cross entropy as the loss function. The experimental results show that the overall mean IoU of U-Net is 74.18%, which is significantly better than the SegNet with IoU 60.54%. Moreover, the segmentation performance of U-Net in each class is at least 11% better than SegNet. Comparison between U-Net and mean IoU of our model shows that our model performs only 0.43% higher than U-Net based on this dataset. However, our model took advantage of fewer parameters than SegNet while ensuring accuracy. Therefore, our model is also faster in training. Nevertheless, the performance of this segmentation model in vehicles and traffic signs needs to be further improved, owing to the generally lower pixel intensities of vehicles in the image and less TrafficSign training data.

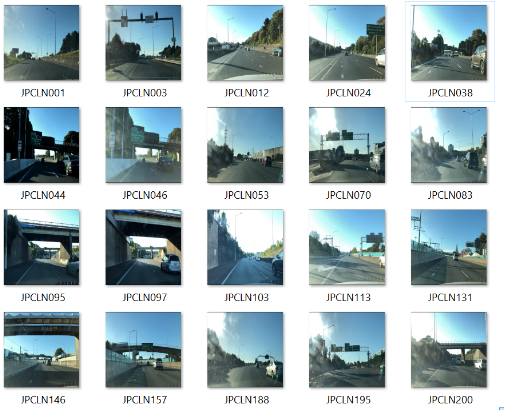

Fig. 1. The raw images of the training data

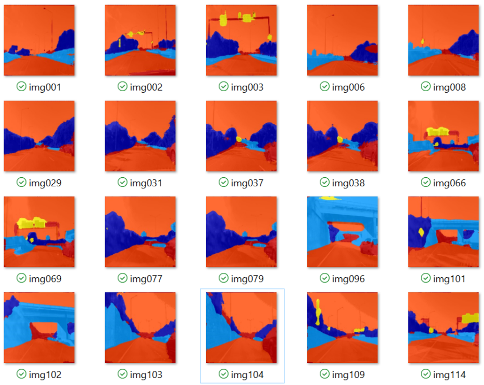

Fig. 2. The results of capsule network segmentation

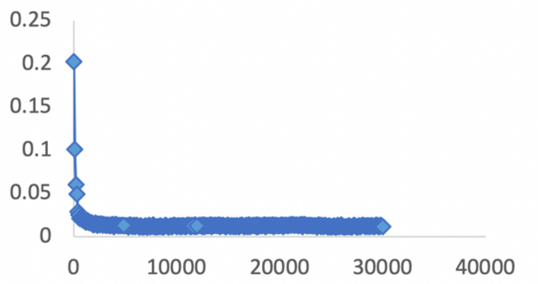

Fig. 3. The loss curve of training process

In order to evaluate our model much accurately, we compared the performance of this model by using a single vector feature expression and traditional dynamic routing mechanism. We substituted the matrix with a vector which takes use of dynamic routing (CapVec-DR), and our proposed VS routing (CapVec-VS) algorithm to conduct the same experiment. As shown in Table Ⅱ, by using a single vector feature expression method, CapVec-DR with dynamic routing mechanism has the lowest accuracy 72.35%. The accuracy of CapVec-VS by using VS routing mechanism is higher than CapVec-DR (1.18%). Nevertheless, the accuracy of these two models based on vector feature expression is lower than ours. The performance of our proposed model is better than other two models.

In Fig. 4, we compare the test loss of CapVec-DR, CapVec-VS, and our model. In the tests, after 2,000 iterations, CapVec-VS had the highest loss value, our model had the lowest loss value. The drop curves of CapVec-VS and our model are roughly similar, the CapVec curve at each point is notably lower than that of other two models. This alludes that the VS routing mechanism may make the model converge faster than the dynamic routing mechanism. After 30,000 iterations, the loss values of CapVec-DR and CapVec-VS are similar, our model has the lowest loss value. Compared with the average losses obtained from the experiments, our model has the lowest value 0.38. The average losses of CapVec-DR and CapVec-VS are 0.65 and 0.57, respectively.

V. Conclusion and Future Work

At present, most computational methods for semantic segmentation mainly employ CNNs as the basic model. In this paper, we take use of a capsule network with a combination of matrices to achieve higher performance of semantic segmentation experimented based on the Auckland traffic dataset that we collected and labeled by ourselves. This dataset was designed according to the characteristics of the capsule network. Therefore, this dataset is able to be utilized for CapsNet-based image segmentation in the near future. Moreover, the VS routing mechanism is able to make the model converge quickly than the dynamic routing mechanism by using CapsNets. In the experiments, IoUs of our models and the segmentation results are higher than that of U-Net and SegNet. Currently, our model has an IoU up to 74.61% based on our own dataset. The proposed method aims to accurately segment out various visual objects in the driving environment to better implement the safety of autonomous vehicles.

In future, for the sake of understanding scene, we will collect more samples from different weather and lighting conditions as well as locations in Auckland. We will acquire more and higher-quality images as our training dataset with advanced video equipment. It provides assistance for acquiring more detailed features of the object for scene understanding. More classes will be imported into the dataset, such as streetlamps, lane lines, and traffic lights. We will add LSTM module to extract the temporal information in the scene to understand the activities of surrounding vehicles. This not only solves the occlusion problem, but also provides a deeper understanding of vehicle-related scenes. We will also explore the neuromorphic approach to deal with both spatial and temporal information by using evolving spiking neural networks [40] and will compare the used approaches in terms of accuracy, time of processing, and level of science understanding [41].

References

[1] G. Yating, W. Yantian, and L. Yansheng, "A survey on deep learning-driven remote sensing image scene understanding: Scene classification, scene retrieval and scene-guided object detection." IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. Vol. 13, pp, 3735–3756, May 2019

[2] J. Yao, S. Fidler and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation, *In IEEE Conference on Computer Vision and Pattern Recognition*, pp. 702-709, 2012

[3] M. Naseer, S. Khan and F. Porikli, "Indoor Scene Understanding in 2.5/3D for Autonomous Agents: A Survey, " IEEE Access, vol. 7, pp. 1859-1887, 2019

[4] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser and M. Nießner. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5828-5839, 2017

[5] J. Xiao, B.C., Russell, J. Hays, K. Ehinger, A. Oliva and A.Torralba. Basic level scene understanding: from labels to structure and beyond. *SIGGRAPH Asia 2012 Technical Briefs*, pp. 1-4, 2012

[6] R. Rishi and D. S. Sisodia. Real-time data augmentation based transfer Learning model for breast cancer diagnosis using histopathological images. In *Advances in Biomedical Engineering and Technology*, pp 473-488, 2020

[7] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann and G. Wang. Boundary-aware feature propagation for scene segmentation. In *IEEE International Conference on Computer Vision*, pp. 6819-6829, 2019

[8] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang and H. Lu. Dual attention network for scene segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3146-3154, 2019

[9] A. Dai and M. Nießner. 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In *European Conference on Computer Vision (ECCV)*, pp. 452-468, 2018

[10] L. Li, B. Qian, J. Lian, W. Zheng and Y. Zhou "Traffic scene segmentation based on RGB-D image and deep learning." IEEE Transactions on Intelligent Transportation Systems, 19(5), 1664-1669, 2017

[11] P. Zhang, W. Liu, H. Wang, Y. Lei and H. Lu "Deep gated attention networks for large-scale street-level scene segmentation." Pattern Recognition, 88, 702-714, 2019

[12] N. Kriegeskorte and P. K. Douglas "Cognitive computational neuroscience." Nature Neuroscience, 21(9), 1148–1160, 2018

[13] U. Güçlü and M. A. Gerven. "Modeling the Dynamics of Human Brain Activity with Recurrent Neural Networks." Frontiers in Computational Neuroscience, pp. 1-7, 2018

[14] Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp.234-241, 2019

[15] A. Badrinarayanan, A. Kendall, and R. Cipolla. "SegNet: A deep convolutional encoder-decoder architecture for image segmentation." IEEE Transactions on Pattern Analysis and Machine Intelligence, (12), pp. 2481, 2017

[16] G. Lin, F. Liu, A. Milan, C. Shen, & I. Reid. "RefineNet: Multi-path refinement networks for dense prediction." IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(5), 1228–1242, 2016

[17] H. Su, F. Liu, Y. Xie, P. Yuan, F. Xing, S. Meyyappan and L. Yang Region segmentation in histopathological breast cancer images using deep convolutional neural network. In *IEEE International Symposium on Biomedical Imaging* (ISBI), pp. 55-58, 2015

[18] F. Saeedan, N. Weber, M. Goesele and S. Roth, Detail-preserving pooling in deep networks, In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9108-9116, 2018

[19] H. Yao, P. Gao, J. Wang, P. Zhang, C. Jiang and Z. Han. "Capsule network assisted IoT traffic classification mechanism for smart cities." IEEE Internet of Things, 7515–7525, 2019

[20] Y. Ma, J. Ren, H. Chen, Z. Liu, G. Z, J. Li, Z. Zheng, Z. Guo, F. Mou, F. Zhou, R. Kong, A. Hou, M. Zhu and Y. He. Classification based on capsule network with hyperspectral image. In *Geoscience and Remote Sensing Symposium*, pp. 2750–2753, 2019

[21] S. Sabour, N. Frosst, and G. Hinton. Dynamic routing between capsules. In *Neural Information Processing Systems*, pp. 3859–3869, 2017

[22] J. Yin, S. Li, H. Zhu and X. Luo. "Hyperspectral image classification using CapsNet with well-initialized shallow layers." IEEE Geoscience & Remote Sensing Letters, 1095, 2019

[23] L. Huang. "Pattern recognition and computer vision." *Chinese Conference on Pattern Recognition and Computer Vision* (PRCV), pp. 517-518, 2018

[24] M. Oeljeklaus, F. Hoffmann and T. Bertram. A fast multi-task CNN for spatial understanding of traffic scenes. *International Conference on Intelligent Transportation Systems* (ITSC), pp. 2825-2830, 2018

[25] Z. Naifan, Y. Jun and A. Kien. Convolutional DLSTM for crowd scene understanding. *IEEE International Symposium*, 61–68, 2017

[26] E. Shelhamer, J. Long and T. Darrell "Fully convolutional networks for semantic segmentation." IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4), 640–651, 2017

[27] K. He, X. Zhang, S. Ren. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016

[28] S. J´egou, M. Drozdzal, D. Vazquez, A. Romero and Y. Bengio. The one hundred layers: Fully convolutional DenseNets for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition Workshops* (CVPRW), 1175–1183, 2017

[29] Y. Fisher and K. Vladlen. "Multi-scale context aggregation by dilated convolutions." CoRR, pp 34-46, 2015

[30] X. Qi et al. Pyramid scene parsing network. In *IEEE Conference on Computer on Vision and Pattern Recognition*, 2881-2890, 2017

[31] H. Junxing, L. Ling, L. Yijun, W. Fengge and Z. Junsuo. "A comparison and strategy of semantic segmentation on remote sensing images." In Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery, pp. 21-29, 2019

[32] B. Shruthi, S. B, and M. El-Sharkawy. 3-level residual capsule network for complex datasets. In *Latin American Symposium on Circuits & Systems* (LASCAS), 1–4, 2020

[33] R. LaLonde, and U. Bagci, "Capsules for object segmentation." arXiv:1804.04241

[34] R. Zhang, M. Li and L. Wang. "Fusion of images and point clouds for the semantic segmentation of large-scale 3D scenes based on deep learning." ISPRS Journal of Photogrammetry and Remote Sensing, pp.85–96, 2019

[35] F. Zhang, Y. Wang, and M. Ye. Network traffic classification method based on improved capsule neural network. In *International Conference on Computational Intelligence and Security* (CIS), pp.174-178, 2018

[36] D. Amara, R. Arthika and P. Latha "Novel deep learning model for traffic sign detection using capsule networks." International Journal of Pure and Applied Mathematics, 118 (20): 4543-4548, 2018

[37] H. Qu, L. Zhang, X. Wu, X. He, X. Hu and X. Wen. Multiscale object detection in infra-red streetscape images based on deep learning and instance level data augmentation. Applied Sciences, pp.553-565, 2019

[38] S. Bonheur, D. Štern, C. Payer, M. Pienn, H. Olschewski, M. Urschler. Matwo-CapsNet: A multilabel semantic segmentation Capsules network. Medical Image Computing and Computer Assisted Intervention, pp.664-672

[39] S. Sabour, N. Frosst and G. Hinton. Matrix capsules with EM routing. *International Conference on Learning Representations*, pp. 749–761, 2018

[40] N. Kasabov, Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence, Springer, 2018

[41] W. Yan, Introduction to Intelligent Surveillance, Springer, 2019

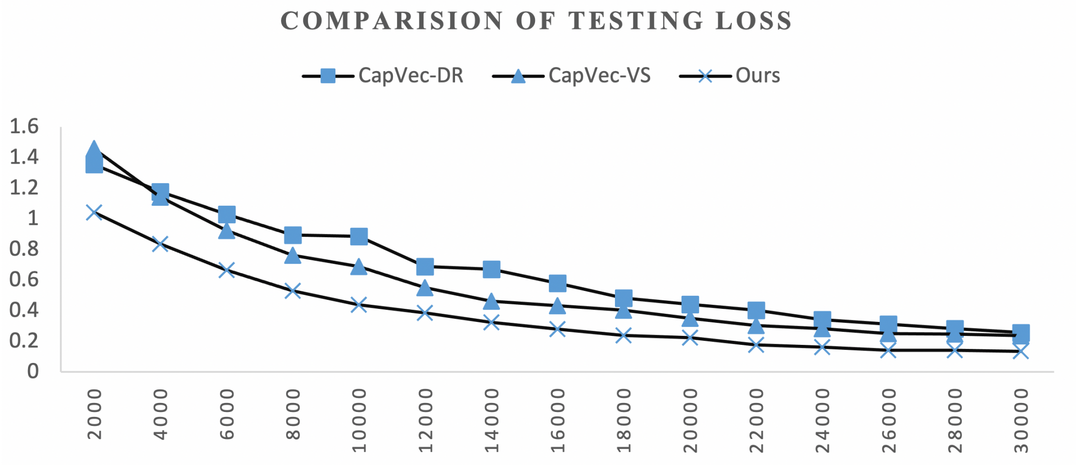

Fig. 4. Comparison of the testing loss

TABLE I. COMPARISONS OF DIFFERENT MODELS BASED ON THE SAME DATASET

Networks	**Classes**						**Mean IoU**
	Vehicle	***Road***	***Sky***	***Building***	***TrafficSign***	***Tree***	
SegNet	28.98%	64.5%	82.87%	84.81%	27.52%	74.57%	60.54%
U-Net	40.23%	77.28%	95.98%	96.67%	48.63%	86.28%	74.18%
Ours	39.26%	79.67%	96.06%	96.82%	49.56%	86.31%	74.61%

TABLE II. COMPARISONS BETWEEN TWO MODELS

Networks	**Classes**						**Mean IoU**
	Vehicle	***Road***	***Sky***	***Building***	***TrafficSign***	***Tree***	
CapVec-DR	36.52%	77.64%	94.18%	94.29%	47.31%	84.17%	72.35%
CapVec-VS	37.9%	79.17%	95.25%	95.75%	48.89%	84.24%	73.53%
Ours	39.26%	79.67%	96.06%	96.82%	49.56%	86.31%	74.61%