SYGNET: A SVD-YOLO BASED GHOSTNET FOR REAL-TIME DRIVING SCENE PARSING

Hewei Wang[1]*, Bolun Zhu*[1]*, Yijie Li*[1]*, Kaiwen Gong*[1]*, Ziyuan Wen*[1]*, Shaofan Wang*[2]*, Soumyabrata Dev*[3,4]

[1]Beijing-Dublin International College, Beijing University of Technology, Beijing 100124, China
[2]Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
[3]The ADAPT SFI Research Centre, Dublin D04V1W8, Ireland
[4]School of Computer Science, University College Dublin, Dublin D04V1W8, Ireland

ABSTRACT

In this paper, we propose SYGNet to strengthen the scene parsing ability of autonomous driving under complicated road conditions. The SYGNet includes feature extraction component and SVD-YOLO GhostNet component. The SVD-YOLO GhostNet component combines Singular Value Decomposition (SVD), You Only Look Once (YOLO) and GhostNet. In the feature extraction component, we propose an algorithm based on VoxelNet to extract point cloud features and image features. In SVD-YOLO GhostNet component, the image data is decomposed by SVD, and we obtain data with stronger spatial and environmental characteristics. YOLOv3 is used to obtain the future map, then convert to GhostNet, which is used to realize the real-time scene parsing by utilizing fewer filters to generate some intrinsic feature maps. We use KITTI dataset to perform our experiments and the results show that the SYGNet is more robust and can further enhance the accuracy of real-time driving scene parsing. The model, dataset, and results of the experiments in this paper are available at: `https://github.com/WangHewei16/SYGNet-for-Real-time-Driving-Scene-Parsing`.

***Index Terms*—** autonomous driving, driving scene parsing, SVD, YOLO, GhostNet

1. INTRODUCTION

Autonomous driving technology is the kind of core technology applied to the in-vehicle Artificial Intelligence (AI) field, and also can be defined as the key to realize smart cars, smart transportation and smart cities. Autonomous driving field consists of scenario generation [1,2], scene recognition [3–5], navigation [6], simulation [7], scene parsing [8, 9], prediction [10, 11] and motion estimation [12, 13]. Among them, scene parsing [14] is a fundamental core part and is used to obtain information of the vehicle itself and the surrounding environment, including vehicles, pedestrians, traffic signs, obstacles, through various sensors. Based on the current research, a large amount of road condition data and a fusion of several algorithms are used during a scene parsing algorithm. This is used to create a robust car management system and thereby achieving smart and auto driving. While the realization of autonomous start technology requires several necessary elements, the most important thing is the massive amount of imaging data which can cover all complex road conditions information. The second most important aspect is to use a robust and accurate algorithm, which can accurately detect complex contour information and dynamic targets.

The usage of autonomous driving has been realized for years but exhibiting the two shortcomings on many occasions: (a) Bad usage accuracy in high-speed driving. Autonomous driving has a higher failure rate for high-speed targets. (b) In close traffic jams, autonomous driving cannot judge the complicated road conditions ahead. Even the above conditions are different, but all of them have got a common point which can be viewed as the inaccuracy of the algorithm. Hence, the promotion of the scene parsing accuracy becomes necessary.

To solve the aforementioned issues, we propose SYGNet for improving scene parsing accuracy. SYGNet based on SVD and YOLO [15] algorithm, and the feature extraction algorithm will use k-means [16] to detect more characters of the images, which classifies the image features twice, and after the feature extraction process completed, combines GhostNet [17] to complete more accurate scene parsing. Our main contribution is in improving the scene parsing of autonomous driving, especially the recognition under traffic jams or complex road conditions.

The main contributions of this paper are as follows:

- A novel autonomous driving recognition model named SYGNet is proposed wherein we introduce feature extraction component and combine SVD-YOLO and GhostNet as a subsequent component.
- SYGNet produces promising results in recognition accuracy, loss value, train and test condition and qualitative figures on KITTI dataset.

This research was conducted with the financial support of Science Foundation Ireland under Grant Agreement No. 13/RC/2106_P2 at the ADAPT SFI Research Centre at University College Dublin. ADAPT, the SFI Research Centre for AI-Driven Digital Content Technology, is funded by Science Foundation Ireland through the SFI Research Centres Programme.

Send correspondence to S Dev, E-mail: soumyabrata.dev@ucd.ie.

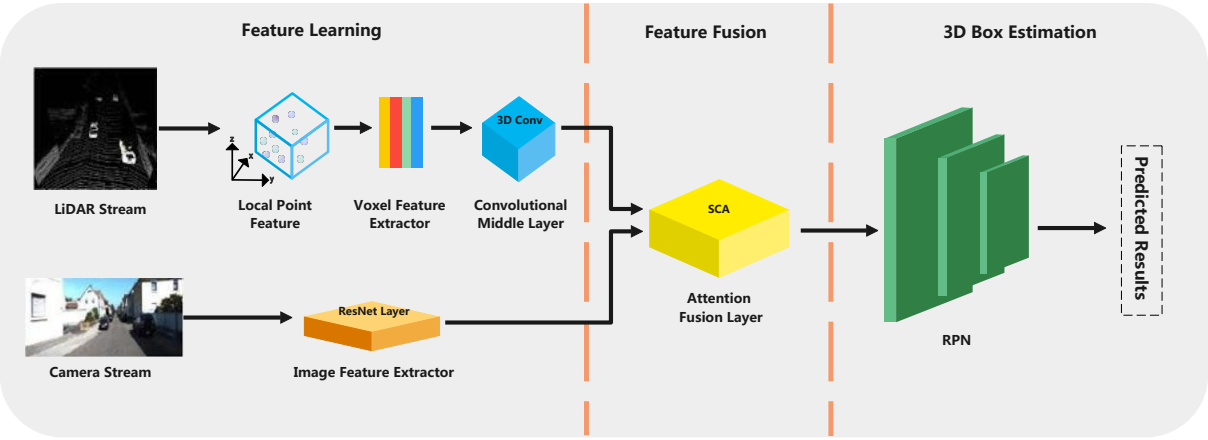


Fig. 1: Architectural overview of proposed Feature Extraction Component in SYGNet.

- The model, diagrams, dataset, and the three types of experiment's results in this paper are available at: `https://github.com/WangHewei16/SYGNet-for-Real-time-Driving-Scene-Parsing`.

2. SYGNET

SYGNet is composed of two components: feature extraction component and SVD-YOLO GhostNet component. The first component is used to extract important perceptual scene features. The second component is responsible for using the model and training parameters to obtain high accuracy perceptual scene parsing results. Figure 1 demonstrates the architectural overview of the feature extraction component in SYGNet. We will introduce the internal architecture of these two components in SYGNet in Section 2.1 and Section 2.2 in detail.

2.1. Feature Extraction Component

The feature extraction learning includes two branches: LiDAR Stream and Camera Stream, which extract point cloud features and image features respectively. For the LiDAR branch, suppose a 3D object contains N points, and the point cloud data of the object is (x_i, y_i, z_i, r_i), where (x_i, y_i, z_i) denotes the Cartesian product of the i-th point. The coordinate, r_i is the reflection value corresponding to the point. Similar to VoxelNet [18], we divide the original point cloud into equal voxel grids, and then uses the spatial coordinates and the relative offset of random sampling points as the representation of each voxel. The relative offset of the points in the j-th voxel grid is the offset of each point from its centroid, the centroid of the j-th voxel $\left\{v_x^j, v_y^j, v_z^j\right\}$ with this format:

$$v_x^j = \frac{\sum_{i=1}^{M} x_i^j}{M}, v_y^j = \frac{\sum_{i=1}^{M} y_i^j}{M}, v_z^j = \frac{\sum_{i=1}^{M} z_i^j}{M} \quad (1)$$

where M is the number of point clouds in the j-th voxel. Although this representation can capture the global spatial information of the point cloud, it ignores the local structure information of the midpoint of each voxel. In order to capture local structural information, this paper designs local directional features, which can be formulated by the following equation:

$$d^j = \frac{\sum_{i=1}^{M} arc\tan\left(\frac{y_i^j - v_y^j}{x_i^j - v_x^j}\right)}{M} \quad (2)$$

The final representation of the i-th point in the j-th voxel grid can be formulated as below:

$$V_{in} = \left\{x_i^j, y_i^j, z_i^j, r_i^j, x_i^j - v_x^j, y_i^j - v_y^j, z_i^j - v_z^j, d^j\right\} \quad (3)$$

Next, the new feature representation is provided to the voxel feature extractor in the feature learning component. The feature extractor described in this paper is composed of the voxel feature encoding layer (VFE) proposed in VoxelNet.

The VFE layer designed in VoxelNet is inspired by PointNet and it is composed of a large number of stacked full connection layers (FCN). In this paper, the voxel feature coding V_{in} of the above design is transformed into the feature space through FCN. Through the mapping operation, the features of the internal points of each voxel grid can be aggregated to encode the surface shape information inside the voxel. Here, FCN is composed of linear layer, batch normalization (BN) layer and ReLU layer, when a point wise feature representation is obtained, element by element maximization is used (Element-wise MaxPooling) to obtain the local aggregation features. Finally, in order to strengthen the point level features, the point level features output by FCN layer are spliced to obtain the final point level combination features. The voxel feature extractor designed in this paper consists of two VEF layers. For the camera branch, a 2D convolution neural network and ResNet layer are designed as the image feature extractor to extract the point level features corresponding to the

point cloud data, which can capture deeper image texture features to achieve better modal fusion (cf. Fig. 1).

Then the result of two streams will merge in feature fusion stage, where has an attention fusion layer to deal with the combined data. After that, it will convert to Region Proposal Network (RPN) in 3D box estimation stage. The predicted result will help to get more accurate feature data range. Finally, the future map with extracted feature will be generated.

2.2. SVD-YOLO GhostNet Component

Massive image data and environmental data can be input as vector data. Suppose we get the following equation:

$$A_{m*n} = U_{m*m}\beta_{m*n}{V_{n*n}}^T \tag{4}$$

where U represents an $m * m$ square matrix, the orthogonal vector in U is defined as left singular vector. β is an $m * n$ matrix, the diagonal elements are defined as singular values except the zero elements, V^T is an $n*n$ square, the orthogonal vector in V is defined as right singular vector. We multiply the transpose of a matrix A by A and find the eigenvalues of A^TA, then we can get $\left(A^TA\right)v_i = \lambda_i v_i$, where v is the right singular vector, and the singular value $\sigma_i = \sqrt{(\lambda_i)}$, the left singular value $u_i = (Av_i)/\sigma_i$, in this case, σ is the singular value and u is the singular vector. The singular value σ is similar to the eigenvalues. In the matrix β, it is also arranged from large to small, and the reduction of σ is extremely fast, so we can describe the matrix with the top r singular value. In this way, part of singular values can be decomposed into:

$$A_{m*n} \approx U_{m*r}\beta_{r*r}{V_{r*n}}^T \tag{5}$$

Hence, put the SVD decomposed data into YOLOv3 neural network and layer function can be formulated as below:

$$h_{W,b}(x) = f\left(W^Tx\right) = f\left(\sum_{i=1}^{3} W_ix_i + b\right) \tag{6}$$

Fine-grained features are added through the passthrough layer in YOLOv2. While in YOLOv3, the feature map obtained from the previous two layers is up-sampled twice, and the feature map obtained before is connected with the feature map obtained after the up-sampling which can be given as:

$$len_{out} = \left[\frac{len_{in} - (kernel_{size} - 1) + 1}{stride} + 1\right] \tag{7}$$

SYGNet performs the same operation again to predict the new size, benefiting from all the previous calculations and the fine-grained nature of the network. After finishing the SVD-YOLO algorithm stage, we connect GhostNet at the end of the component. GhostNet reduces the computational costs of deep neural networks by utilizing fewer filters to generate some intrinsic feature maps, which improves efficiency and accuracy, and make SYGNet achieves real-time.

3. EXPERIMENTS AND RESULTS

3.1. Dataset

In experiment section, we use the KITTI dataset for training and prediction. Based on 7400 KITTI dataset images containing different vehicles and pedestrians, this experiment detects vehicles and pedestrians in road images. Each image contains upto 15 cars and 30 pedestrians, with various degrees of occlusion and truncation. Two categories of data ("Car" and "Pedestrain") in the object detection data in the KITTI dataset are selected as the dataset for training the neural network. There are 7400 images in total, which are divided into training set and testing set according to the ratio of 4:1. The number of iterations is 100000 and the batch size is 64.

3.2. Ablation experiment

This paper adopts an ablation experiment to perform autonomous driving's accuracy to verify and analyze the effect of algorithms in our SYGNet. In SVD-YOLO GhostNet component, we need to decide which model algorithm we use in the first stage of this component. Therefore, we perform an ablation experiment to compare the performance of YOLOv3, RCNN, SVD-YOLOv2, SVD-RCNN and SVD-YOLOv3 on KITTI dataset. Fig. 2 illustrates the SVD-YOLOv3 algorithm achieves the best performance in the whole test process.

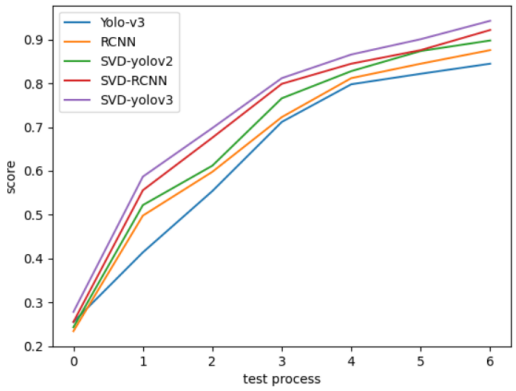
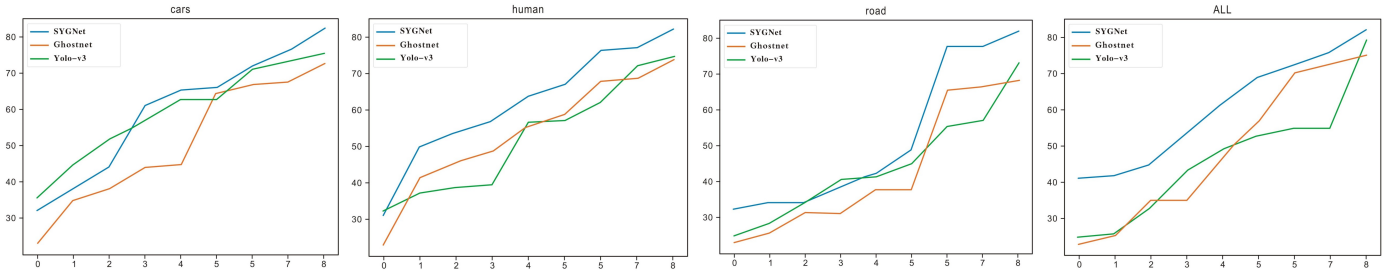


Fig. 2: Ablation experiment on the test process of component in SVD-YOLO GhostNet Component.

Table 1 demonstrates the SVD-YOLOv3 algorithm has the best performance in five different modes ("Cars", "People", "Edge", "Side" and "Light" mode) in KITTI dataset. This further confirmed that SVD-YOLOv3 has promising performance in driving scene parsing. Hence, we decide to adopt SVD-YOLOv3 algorithm in this component.

3.3. Quantitative evaluation

By comparing different methods in three different scene modes ("Cars", "People" and "Edge" mode) in Table 2, it



(a) Cars mode (b) Human mode (c) Road mode (d) All mode

Fig. 3: We observe that the accuracy of different models in various modes increases with the increase of training time. The x-axis represents training time unit and y-axis represents accuracy values.

Algorithms	Cars	People	Edge	Side	Light
YOLOv3	78.9	76.4	77.1	73.4	79.0
RCNN	82.8	79.9	81.1	77.3	82.6
SVD-YOLOv2	86.7	83.4	85.1	81.2	86.2
SVD-RCNN	90.2	87.4	89.0	84.8	86.2
SVD-YOLOv3	94.1	90.9	93.0	88.7	89.8

Table 1: Ablation experiment on the accuracy of component in SVD-YOLO GhostNet Component.

can be found that the SYGNet has promising performance in terms of reliability, which verifies the reliability under the same dataset. In terms of the experimental design, we have also added feature extraction component to the comparison methods. This experiment can prove that the accuracy of the model is higher than that of the other state-of-the-art methods, and also further prove the importance of feature extraction component to improve its accuracy.

Algorithms	Cars	People	Edge
GhostNet	78.9	76.4	77.1
RCNN [19]	82.8	79.9	81.1
Feature Extraction - GhostNetv2	86.7	83.4	85.1
Feature Extraction - RCNN	90.2	87.4	89.0
SYGNet	94.1	90.9	93.0

Table 2: Comparison results of different methods.

Figure 3 shows the experiment about the accuracy's changes of different models in four different scene modes ("Cars", "People", "Road" and "All" mode) with the increase of training time. We can see that at the end of the training, SYGNet has far surpassed other comparison methods. In the "All mode" scene, SYGNet ranks first among these methods in the whole process, which can prove the superiority of SYGNet during the training optimization process.

3.4. Qualitative evaluation

Figure 4 shows the qualitative results on the KITTI dataset. Different colors mark different categories of objects recognized. We can see that our perceptual recognition effect reaches the expectation, and we have promising recognition processing on the edges of the different types of objects.

Fig. 4: Qualitative results on the KITTI dataset. We observe that the generated segmented masks conform well with the boundaries of the different objects in the scene.

4. CONCLUSION & FUTURE WORK

This paper studies the autonomous driving scene parsing technology, analyzes some inherent problems related to training time of the neural nets, and proposes SYGNet. In the feature extraction component, we propose an algorithm based on VoxelNet to extract point cloud features and image features. In SVD-YOLO GhostNet component, SVD decompose the image data into YOLOv3, which are used to obtain the future map, then convert to GhostNet, which reduces the computational costs of deep neural networks to improve the efficiency. Finally, the experimental results show that SYGNet can effectively and significantly improve the scene parsing and recognition ability of autonomous driving under traffic jams or complex road conditions, so as to make the autonomous driving technology more safe and reliable. In the future, we will focus on the reuse and fusion of visual transformer and combination with DeepBillboard so as to greatly improve the accuracy of autonomous driving technology.

5. REFERENCES

[1] S. Tan, K. Wong, S. Wang, S. Manivasagam, M. Ren, and R. Urtasun, "SceneGen: Learning to Generate Realistic Traffic Scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 892–901.

[2] J. Wang, A. Pun, J. Tu, S. Manivasagam, A. Sadat, S. Casas, M. Ren, and R. Urtasun, "AdvSim: Generating Safety-Critical Scenarios for Self-Driving Vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9909–9918.

[3] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14 141–14 152.

[4] H. Wang, Y. Li, S. Xi, S. Wang, M. S. Pathan, and S. Dev, "AMDCNet: An attentional multi-directional convolutional network for stereo matching," *Displays*, vol. 74, p. 102243, 2022.

[5] S. Dev, M. Hossari, M. Nicholson, K. McCabe, A. Nautiyal, C. Conran, J. Tang, W. Xu, and F. Pitié, "Localizing adverts in outdoor scenes," in *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 2019, pp. 591–594.

[6] A. Badki, O. Gallo, J. Kautz, and P. Sen, "Binary TTC: A Temporal Geofence for Autonomous Navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 12 946–12 955.

[7] Y. Chen, F. Rong, S. Duggal, S. Wang, X. Yan, S. Manivasagam, S. Xue, E. Yumer, and R. Urtasun, "GeoSim: Realistic Video Simulation via Geometry-Aware Composition for Self-Driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7230–7240.

[8] S. Li, Z. Yan, H. Li, and K.-T. Cheng, "Exploring intermediate representation for monocular vehicle pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1873–1883.

[9] S. Dev, H. Javidnia, M. Hossari, M. Nicholson, K. McCabe, A. Nautiyal, C. Conran, J. Tang, W. Xu, and F. Pitié, "Identifying candidate spaces for advert implantation," in *Proceedings of the IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, 2019, pp. 503–507.

[10] C. Choi, J. H. Choi, J. Li, and S. Malla, "Shared cross-modal trajectory prediction for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 244–253.

[11] B. Kim, S. H. Park, S. Lee, E. Khoshimjonov, D. Kum, J. Kim, J. S. Kim, and J. W. Choi, "LaPred: Lane-Aware Prediction of Multi-Modal Future Trajectories of Dynamic Agents," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14 636–14 645.

[12] C. Luo, X. Yang, and A. Yuille, "Self-Supervised Pillar Motion Learning for Autonomous Driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 3183–3192.

[13] H. Wang, M. S. Pathan, and S. Dev, "Stereo matching based on visual sensitive information," in *Proceedings of the 6th International Conference on Image, Vision and Computing (ICIVC)*, 2021, pp. 312–316.

[14] S. Dev, M. Hossari, M. Nicholson, K. McCabe, A. N. C. Conran, J. Tang, W. Xu, and F. Pitié, "The ALOS dataset for advert localization in outdoor scenes," in *Proc. Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2019, pp. 1–3.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] M. Jain, T. AlSkaif, and S. Dev, "Validating clustering frameworks for electric load demand profiles," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8057–8065, 2021.

[17] K. Han, Y. Wang, Q. Tian, J. Guo, and C. Xu, "GhostNet: More Features From Cheap Operations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[18] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *IEEE Computer Society*, 2013.