Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 6, No. 2, May 2021, Pp. 101-108

Moving objects semantic segmentation using SegNet with VGG encoder for autonomous driving

Wahyudi Setiawan*[1], Kori Cahyono[2]
Informatics Department, UniversitasTrunojoyo Madura, Indonesia[1]
Bappedalitbang Riau, Indonesia[2]

Article Info
Keywords:
Autonomous Driving, CamVid, SegNet, Semantic Segmentation, VGG encoder

Article history:
Received: January 11, 2021
Accepted: February 16, 2021
Published: May 31, 2021

Cite:
Setiawan, W. (2021). Moving Objects Semantic Segmentation using SegNet with VGG Encoder for Autonomous Driving. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(2). https://doi.org/10.22219/kinetik.v6i2.1203

*Corresponding author.
Wahyudi Setiawan
E-mail address:
wsetiawan@trunojoyo.ac.id

Abstract
Segmentation and recognition become the general steps to identify objects. This research discusses pixel-wise semantic segmentation based on moving objects. The data from the CamVid video which is a collection of autonomous driving images. The image data consist of 701 images accompanied by labels. The segmentation and recognition of 11 objects contained in the image (sky, building, pole, road, pavement, tree, sign-symbol, fence, car, pedestrian and bicyclist) is representing. This moving object segmentation is carried out using SegNet which is one of the Convolutional Neural Network (CNN) methods. Image segmentation on CNN generally consists of two parts: Encoder and Decoder. VGG16 and VGG19 pre-trained networks are used as encoders, while decoders are the upsampling of encoders. Network optimization uses stochastic gradient descent of Momentum (SGDM). The test produces the best recognition was road objects with an accuracy of 0.96013, IoU 0.93745, F1-Score 0.8535 using VGG19 encoder, while when using VGG16 encoder accuracy was 0.94162, IoU 0.92309, and F1-Score 0.8535.

1. Introduction

Segmentation plays an important role in the field of computer vision. Segmentation divides an image into several areas according to a particular object. From segmentation, recognition can be done for objects contained in the image. Segmentation consists of 2 types: semantic segmentation and instance segmentation. Semantic will divide the image into certain classes of objects, for example, glass objects that are different from bottle objects. Whereas instance segmentation will divide the image into classes of specific objects, each object is interpreted with a different class [1][2][3][4][5].

The implementation of semantic segmentation is not only limited to images. This can be done on video. Actually, the video consists of many images with moving objects. If the image is seen one by one, it can be seen moving objects that are all part of one video. Implementation of moving object segmentation includes autonomous driving [6][7][8], robotics [9], medical imaging [10][11][12], and agriculture [13][14].

In this study segmentation and recognition of objects in city-view are carried out. Generally, city-view segmentation and recognition are used for autonomous driving. Segmentation is performed by multiclass objects contained in the image collection. These classes include sky, building, pole, road, pavement, tree, sign-symbol, fence, car, pedestrian and bicyclist. The function of segmentation in an autonomous driver is to be able to differentiate the visible object [15].

Previous studies of moving object segmentation & recognition in autonomous driving include Yu *et al.* using the Bilateral Segment Network (BiseNet). This method consists of 2 parts spatial & context path with Xception network as a backbone. BiseNet has a Feature Fusion Module & Attention Refinement Module. The test results using the CamVid dataset show $Mean - IoU$ 0.687 with the highest recognition is road objects with an accuracy of 0.9460 [16]. Furthermore, Simon *et al.* using CNN densely. The architecture consists of 5 dense block, 2 convolutional layers, 2 transitions up, 2 transition down, 4 concatenation, 2 skip connections. Test results with the CamVid dataset show $Mean - IoU$ 0.669, global accuracy 0.915 with the highest accuracy of road objects 0.945 [17]. Siam *et al.* use the Convolution Gated Recurrent Network architecture with 7 Convolutional Layers, 3 ReLU on blocks 4,5 and 6. Next 2 pooling on layers 1 and 2. Meanwhile, the deconvolutional layer at the end of architecture network. The test results get the $Mean - IoU$ 0.483 with the highest IoU on the sky object 0.875 [18]. Visin *et al.* using the Recurrent Neural Network architecture for semantic segmentation. The test results get IoU 0.588 with the highest accuracy on the road object that is 0.98 [19].

Research on semantic segmentation has been widely developed, there are many things that can be contributed. Performance Measure in previous research needs to be improved, especially for segmentation the objects. In this

research, semantic segmentation was carried out using SegNet. The novelty of this study uses VGG16 and VGG19 as an encoder. The encoder format on SegNet matches the convolutional layer in VGG [20]. In addition, the technique for reduce imbalancing classes uses weighting the median frequency. The experiment data from CamVid datasets.

2. Research Method

2.1 Semantic segmentation with SegNet

Semantic Segmentation is image segmentation based on pixel-intensity value. The result can be differentiate each image objects [21]. In general, semantic segmentation architecture consists of encoder and decoder. The encoder is a pre-trained classification network such as VGG or ResNet. The decoder is a feature projection with lower resolution from the encoder stage to pixel space with higher resolution to get the classification results. SegNet is made with efficiency architecture for pixel-wise. The encoder include convolution and max-pooling layer. If use VGG-16, there are 13 convolutional layers. The fully connected layer is not used. Max-pooling indices (location) in the encoder is stored see Figure 1. In the decoder section, there are upsampling and convolution layer. Upsampling is done by max-pooling indices in the encoder section.

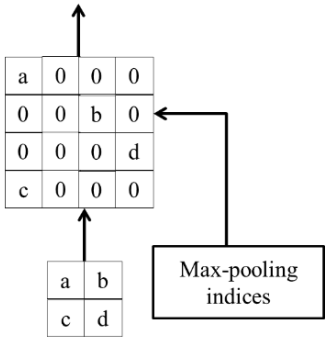


Figure 1. Max-Pooling Indices

Semantic segmentation for anonymous driving requires the ability of segmentation models such as road & building, shapes like car & pedestrian, as well as understanding spatial-relationship between classes such as road & sidewalk. SegNet has an encoder network. It is associated with a decoder network followed by the final pixel-wise labels classification layer. The architecture can be illustrated in Figure 2. The encoder network consists of 13 convolutional layers with a VGG16 network designed for object classification [22]. The Fully Connected Layer is removed to maintain the maps feature in the deepest encoder output. This reduces the number of parameters in the SegNet encoder significantly (from 134 million to 14.7 million) [20]. The same thing was done when VGG19 used as an encoder. Meanwhile VGG19 consist of 16 convolutional layer [22].

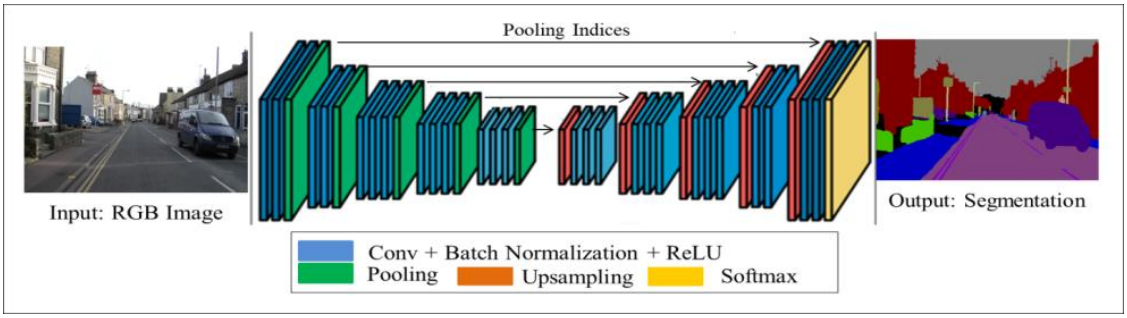
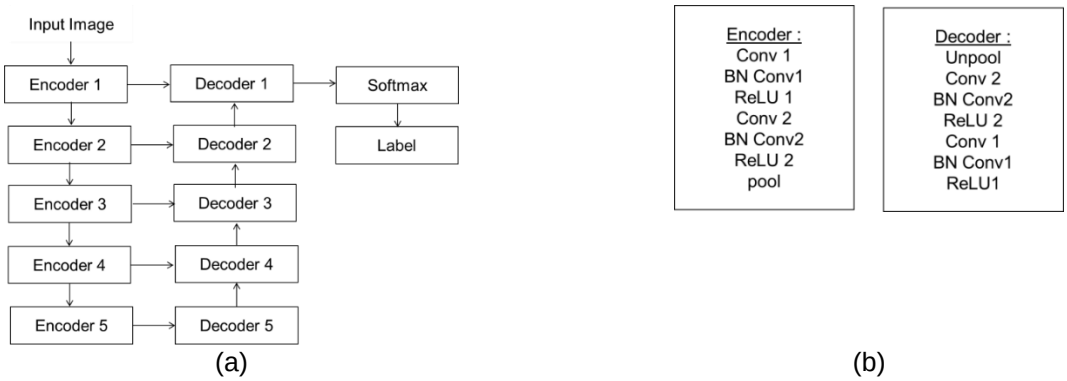


Figure 2. SegNet Architecture [20]

2.2 Arrangement of Segnet VGG layers

Segnet architecture needs to be specified in the composition of the layers. For segNet using VGG16 it has 91 layers ranging from image input to labels. The architecture consists of image input, convolutional layer encoder blocks (each block has 7, 7, 10, 10 and 10 layers), convolutional layer decoder blocks (consisting of 10, 10, 10, 7, and 7 layers), softmax, and label. Whereas Segnet VGG19 has 109 layers consisting of image input, convolutional layer encoder blocks (consisting of 7, 7, 13, 13 and 13 layers), convolutional layer decoder blocks (consisting of 13, 13, 13, 7, and 7 layers), softmax, and label.

Each encoder and decoder block can have 7, 10 and 13 layers. For 7 layers consist of convolutional, Batch Normalization and ReLU 2 layers each plus pooling. For 10 and 13 layers consist of 3 and 4 convolutional layers, Batch Normalization, and ReLU activation as well as the last layer plus pooling. Whereas the decoder consists of unpooling, followed by convolutional, batch normalization and ReLU with the vise versa convolutional layer arrangement of the encoder. The number of each convolutional layer, batch normalization and ReLU adjusts to the number of decoder layers. In Figure 3 (a) the graph layer is shown, while Figure 3 (b) is a 7 layer arrangement in the SegNet VGG encoder and decoder.



(a) (b)

Figure 3. (a) Graph Layer (b) Block Encoder and Decoder from SegNet VGG

SegNet uses VGG16 Encoder has 91 layers (input layer, 31 first layers on VGG16 + 13 BN layer, 44 layer decoder, softmax, and label). Whereas SegNEt with VGG19 encoder has 109 layers (input layer, 37 first layers on VGG19 + 16 BN layer, 53 layer decoder, softmax, and label). The complete SegNet layer details are shown in Table 1.

Table 1. Number of Layers in SegNet

Layer	Input	Encoder		Decoder	Softmax + Label	Total of Layers	Type of Encoder
		Conv1 until maxpooling5	Batch Normalization				
Number of Layers	1	31	13	44	2	91	VGG16
	1	37	16	53	2	109	VGG19

2.3 Dataset

The dataset used comes from The Cambridge-driving Labeled Video Database (CamVid). The data consists of images that are 701 video chunks. In another part, there has been a groundtruth segmentation of pixel-wise masks based images. In the video, there are 32 semantic classes. For this study segmentation and recognition of 11 objects were carried out [15]. An example of CamVid data is in Figure 4.

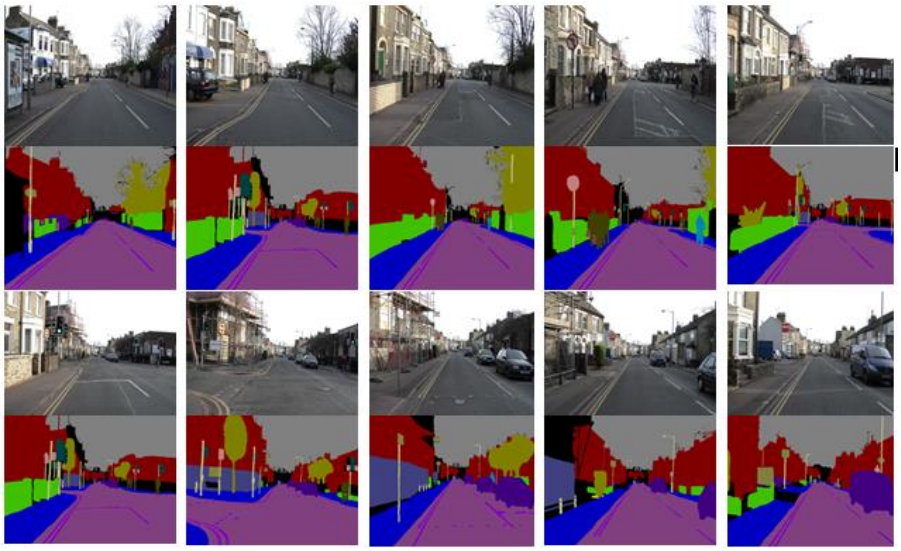


Figure 4. Example Image in CamVid Video

Furthermore, to do semantic segmentation, the image must have a label on each pixel. At the time of labeling, the color is determined to differentiate each object. For the CamVid dataset, the RGB color is initialized in Table 2.

Table 2. RGB Color of the CamVid Object Class

Class	RGB	Class	RGB	Class	RGB Color	Class	RGB
Sky	128 128 128	Road	128 64 128	SignSymbol	192 128 128	Pedestrian	64 64 0
Building	128 0 0	Pavement	60 40 222	Fence	64 64 128	Bicyclist	0 128 192
Pole	192 192 192	Tree	128 128 0	Car	64 0 128		

2.4 Measurement of results

Measurements for semantic segmentation include pixel accuracy, Intersection-over-Union (IoU) and F1-score. Pixel accuracy is the percentage of pixels in an image that is properly classified. But pixel accuracy can produce measurements that are less specific if the objects contained in the image are not balanced (imbalance). For this reason, IoU, which is a jaccard index, was introduced. IoU is the overlap area between predicted results and groundtruth divided by the union area between predicted results and groundtruth. Metric range between 0-1 (0-100)%. 0 indicates no overlap, 1 indicates perfectly overlapping segmentation. Mean-IoU is the average IoU per class. Besides IoU, another metric is F1-score or Dice-Coefficient [21].

2.5 Research stages

Image input on CamVid data is quite large, 720 × 960. Image reduction is needed to speed up time and memory usage. The image reduction to 360 × 460. Next, devide the image into 2 parts: training and testing with a percentage of 60:40, so that it becomes 421 images for training and 280 images for testing [20]. Then create a VGG16 or VGG19 encoder segnet network as explained in section 2.2. Furthermore, the class normalization with weights to improve training data using the weight median frequency [23]. The training options by optimization used the Stochastic Gradient Descent with Momentum (SGDM), epoch 100, and learning-rate 1e-3.

3. Results and Discussion

3.1 Result of SegNet with VGG encoder

Before testing the network the CamVid dataset is analyzed. In ideal conditions, each class has the same number of pixels so the training can run well. However, the pixel distribution can be seen in Figure 5, showing that the pixel distribution is uneven. This can cause losses during the training process because the results of testing can only support a large number of pixel of classes.

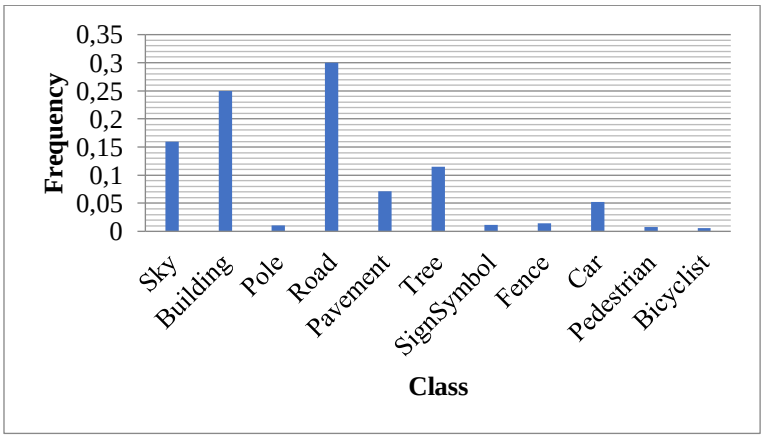


Figure 5. Frequency of Class in Camvid Dataset

Frequency distribution in the classes is not evenly distributed, there is a data gap between the class that has a large number of pixels (Road, Sky, and Building) with a class that has a small number of pixel (Pedestrian and biclyclist). This can be minimized by giving a weighting to each class. Weighting is done by median frequency balancing. Formula of Image frequency and class weight contained on Equation 1 and Equation 2 [20][23]. Weighting results per class are shown in Table 3.

$$if = \frac{pc}{ipc} \tag{1}$$

$$cw = \frac{median(if)}{if} \quad (2)$$

$if = imageFrequency; pc = pixelcount; ipc = imagepixelcount; cw = classweight;$

Table 3. Class Weight on CamVid Dataset

Objects	PixelCount	ImagePixelCount	imageFrequency	ClassWeight
Sky	1.9161e + 07	1.2079e + 08	1.59E-01	3.19E-01
Building	2.932e + 07	1.2079e + 08	2.43E-01	2.0E-01
Pole	1.1975e + 06	1.2079e + 08	9.91E-03	5.10E + 00
Road	3.5212e + 07	1.2113e + 08	2.91E-01	1.74E-01
Pavement	8.401e + 06	1.1802e + 08	7.12E-02	7.10E-01
Tree	1.3547e + 07	1.1197e + 08	1.21E-01	4.18E-01
SignSymbol	1.3049e + 06	1.1716e + 08	1.11E-02	4.54E + 00
Fence	1.7308e + 06	6.2899e + 07	2.75E-02	1.84E + 00
Car	6.1084e + 06	1.2079e + 08	5.06E-02	1.00E + 00
Pedestrian	8.5029e + 05	1.1111e + 08	7.65E-03	6.61E + 00
Bicyclist	6.4745e + 05	6.5491e + 07	9.89E-03	5.12E + 00

Furthermore, training is conducted on the segnet network and testing on one of the test images, for example, image 0016E5-00540.png. The display of the test results shows the difference in pixels from the results of segmentation with groundtruth. Green and magenta colors show this difference, as shown in Figure 6.

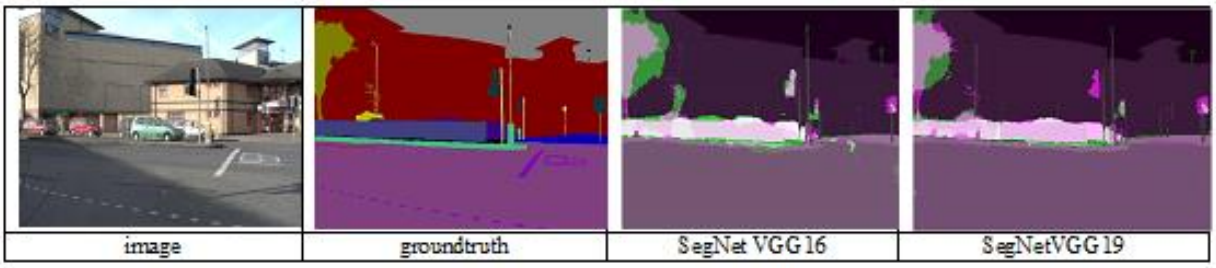


Figure 6. Image, Groundtruth, Differ Groundtruth with SegNet (Green & Magenta Regions)

For the measurement, it used Intersection over Union (IoU) as the standard of semantic segmentation. It gives similarity between predicted segmentation images and groundtruth. The result shown in Table 4. This table is the IoU of 0016E5-00540.png images that use SegNet with VGG19 and VGG16 encoders.

Table 4. IoU of the 0016E5-00540.png Test Image

	Segnet Encoder	
Objects	VGG19	VGG16
Sky	0.95696	0.93935
Building	0.88168	0.8526
Pole	0.22521	0.18236
Road	0.97915	0.97809
Pavement	0.5342	0.57232
Tree	0.56757	0.50384
SignSymbol	0.16843	0.40089
Fence	0.54981	0.57319
Car	0.096539	0.23671
Pedestrian	0	0
Bicyclist	0	0

Next, all testing data will be tested. At this stage, a minibatch-size (MB) can be used to reduce memory usage. In this testing used the size of MB = 4. Size can be increased and lowered according to the type of GPU that is owned. The results of testing using all testing data are shown in Table 5. Each class displayed F1-score, IOU and Pixel Accuracy

results. The test shows that the best segmentation in the Road class with IoU 0.93745 and Sky with IoU 0.91035 uses VGG19 encoder. As for VGG16, the best segmentation is Road with IoU 0.92309 and Sky with IoU 0.91187.

Table 5. Segment VGG Encoder Segmentation Trial Results

	VGG19			VGG16		
Class	F1-score	IoU	Pixel Accuracy	F1-score	IoU	Pixel Accuracy
Sky	0.91043	0.91035	0.94262	0.91174	0.91187	0.94424
Building	0.70453	0.81619	0.86508	0.70609	0.81751	0.87094
Pole	0.70064	0.31375	0.70088	0.6785	0.27505	0.72128
Road	0.8535	0.93745	0.96013	0.79409	0.92309	0.94162
Pavement	0.8047	0.76287	0.86992	0.75078	0.73145	0.83115
Tree	0.75074	0.77598	0.85439	0.74243	0.77698	0.86566
SignSymbol	0.62729	0.47004	0.74109	0.60527	0.44315	0.72011
Fence	0.55136	0.42921	0.83315	0.60739	0.55829	0.80717
Car	0.76649	0.78554	0.9057	0.70275	0.70513	0.90558
Pedestrian	0.66364	0.4678	0.78806	0.6317	0.42395	0.76554
Bicyclist	0.68819	0.66508	0.84546	0.58764	0.54061	0.69311

Furthermore, an average measurement of 11 classes was made in the test dataset. The measurement results are in Table 6.

Table 6. Average Segmentation Results of Segnet VGG Encoders

SegNet Encoder	Global Accuracy	Mean Accuracy	MeanIoU	WeightedIoU	The mean F1-Score
VGG19	0.90221	0.84604	0.66675	0.83834	0.74041
VGG16	0.89544	0.82422	0.6461	0.82837	0.71254

Measuring for semantic segmentation results can be done with $global - accuracy$, $mean - accuracy$, $mean - IoU$, $weighted - IoU$, and $mean - F1 - score$. Global accuracy is accuracy with conformity between results and overall groundtruth. This accuracy does not take into account the accuracy of the class. The test was conducted on 11 classes out of 32 classes. Whereas mean accuracy is the average accuracy of 11 classes that were tested. Mean-IoU is the $average - IoU$ of the 11 class. $Weighted - IOU$ is the weighted average of IoU. This metric is used when the image has a disproportionate size class. This is done to reduce the impact of errors on classes with small pixels. The $mean - F1 - Score$ is the $F1score$ average of 11 classes on CamVid. The difference between the original image and groundtruth/label.

Generally, data in the real world is always presented in the form of imbalance classes. For this reason, a technique is needed to reduce this imbalance. One technique used is the median frequency balancing, which gives weight to each class so that the data distribution is better [23]. If the research focuses on data with large classes, it certainly does not suitable for the real problems. But if have to focus also on classes with a limited number of pixels it will certainly be a challenge to do balanced data. Median frequency balancing was done for training data only. After the training data was weighted, pixel spread for small object classes can be improved.

The comparison result with imbalance classes show that class with large pixel values can be recognized properly. The result shows in Table 7. The SegNet with VGG19 encoder has $meanIoU$ 0.6670, Accuration of 0.9601 for road object, and accuration of 0.2751 for pole object. While the SegNet with VGG16 encoder has $meanIoU$ 0.6460, Accuration of 0.9416 for road object, and accuration of 0.3138 for pole object.

For the small object, it cannot be recognized properly. The Imbalance classes method doesn't recommended for small objects segmentation. The results of the study still need to be increased in the $meanIoU$ for semantic segmentation. Especially in classes with small object, so they are not underpresented.

Table 7. Comparison with Previous Studies

Method	MeanIoU	Highest Accuration (Road Class)	Lowest Accuration (Pole Class)
Visin et al.[19]	0.5880	0.9800	0.3560
Proposed Method1 (Segnet VGG16)	0.6460	0.9416	0.3138
Proposed Method 2 (SegNet VGG19)	0.6670	0.9601	0.2751
Jegou et al.[17]	0.6690	0.9450	0.3780
ChangqianYu et al.[16]	0.6870	0.9460	0.3190

3.2 Discussion

Segnet is an architecture that consists of 2 parts: encoder and decoder. The encoder on the Segnet used the VGG architecture without a fully connected layer to speed up computation and decrease memory [20]. In fact, for the VGG encoder, a batch normalization layer is added to each convolutional layer. Data normalization is needed so that the ReLU activation process can converge more quickly and of course speed up computation [24]. Furthermore, the decoder is the reverse process of the encoder. The decoder process starts with unpooling, the final convolutional layer block process, and then the initial convolutional layer block process. At the end of the SegNet architecture, there is softmax activation and output label segmentation results.

Segmentation is carried out on 11 of the 32 objects contained in each image in the dataset. The distribution of the pixel-numbers was imbalance classes. It can cause segmentation failure. For this reason, the imbalance classes reduction was carried out. The reduction using weight of the median frequency. The weighting process is carried out in 3 stages: First, looking for the frequency of each pixel. This frequency is obtained by dividing the number of pixels of a certain class by the number of pixels in the entire image. Second, order the frequency of each class in ascending order to get the median frequency. Third, calculate the weight of each class by dividing the median frequency by the image frequency value of each class [23].

The use of VGG19 is better than VGG16 encoder for large object segmentation. The VGG19 convolutional layers are more than VGG16. It has an additional convolutional layer in the last 3 blocks before max-pooling. Meanwhile there is more an update of weight and bias on the network that affects the segmentation results. For the small objects segmentation, it need other treatment for segmentation succed.

We just experiment one scenario for computation. The Convolutional neural network architectures applied SGDM optimization and setting variables value for network including epoch, learning rate, and minibatch-size. Epoch is an iteration with backpropagation. Learning stops when the max-epoch is reached. Minibatch-size is the amount of data processed simultaneously by the GPU, while learning rate is the major parameter that controls the speed of training. The challenge with deep learning projects is big data and high computation require that the devices used also have specification standards. GPU Ge Force GTX 1060 was used for this research. If the computer specification better, it can be tested in the higher value of minibatch-size and epoch with the aim of improving the segmentation results.

4. Conclusion

The application of SegNet for semantic segmentation has been applied to the CamVid dataset using 11 classes for autonomous driving. The results obtained can be increased by the Mean-IoU value, especially in techniques for completing imbalanced data. Training data can also be enlarged, especially for classes with small pixel representations. For further research, it can use the DeepLab semantic segmentation method from Google in the hope that mean-IoU can be better. Besides that, it can be tested on other autonomous driving datasets such as KITTI [25] and Mapillary dataset [26].

References

[1] J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440. https://doi.org/10.1109/cvpr.2015.7298965

[2] Y. Li, J. Dai, and X. Ji, “Fully Convolutional Instance-aware Semantic Segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2359–2367. https://doi.org/10.1109/cvpr.2017.472

[3] R. Girshick, J. Donahue, T. Darrell, U. C. Berkeley, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2–9. https://doi.org/10.1109/CVPR.2014.81

[4] J. Dai, K. He, and J. Sun, “Instance-aware Semantic Segmentation via Multi-task Network Cascades,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3150–3158. https://doi.org/10.1109/cvpr.2014.81

[5] A. Khoreva, R. Benenson, J. Hosang, and M. Hein, “Simple Does It : Weakly Supervised Instance and Semantic Segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 876–885. https://doi.org/10.1109/CVPR.2017.181

[6] M. Yahiaoui *et al.*, “FisheyeMODNet: Moving Object detection on Surround-view Cameras for Autonomous Driving,” 2019, pp. 1–4.https://doi.org/10.21427/v1ar-t994

[7] H. Rashed, M. Ramzy, V. Vaquero, A. El Sallab, G. Sistu, and S. Yogamani, “FuseMODNet : Real-Time Camera and LiDAR based Moving Object Detection for robust low-light Autonomous Driving,” in *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 1–10. https://doi.org/10.1109/ICCVW.2019.00293

[8] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, and Q. Xu, “nuScenes : A multimodal dataset for autonomous driving,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, no. March, pp. 1–16. https://doi.org/10.1109/CVPR42600.2020.01164

[9] G. Gordon, “Social behaviour as an emergent property of embodied curiosity : a robotics perspective,” *Philos. Trans. B*, vol. 374, pp. 1–7, 2019. https://doi.org/10.1098/rstb.2018.0029

[10] W. Lumchanow and S. Udomsiri, “Image classification of malaria using hybrid algorithms : convolutional neural network and method to find appropriate K for K-Nearest neighbor,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 1, pp. 382–388, 2019. https://doi.org/10.11591/ijeecs.v16.i1.pp382-388

[11] W. Setiawan, M. I. Utoyo, and R. Rulaningtyas, “Transfer learning with multiple pre-trained network for fundus classification,” *TELKOMNIKA*, vol. 18, no. 3, pp. 1382–1388, 2020. https://doi.org/10.12928/telkomnika.v18i3.14868

[12] M. Siam, V. Sepehr, M. Jagersand, and N. Ray, “Convolutional Gated Recurrent Networks for Video Segmentation,” in *IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 1–5. https://doi.org/10.1109/ICIP.2017.8296851

Cite: Cite: Setiawan, W. (2021). Moving Objects Semantic Segmentation using SegNet with VGG Encoder for Autonomous Driving. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(2). https://doi.org/10.22219/kinetik.v6i2.1203

[13] N. Fatihahsahidan, A. K. Juha, N. Mohammad, and Z. Ibrahim, "Flower and leaf recognition for plant identification using convolutional neural network," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 2, pp. 737–743, 2019. https://doi.org/10.11591/ijeecs.v16.i2.pp737-743
[14] M. Syarief and W. Setiawan, "Convolutional neural network for maize leaf disease image classification," *TELKOMNIKA*, vol. 18, no. 3, pp. 1376–1381, 2020. https://doi.org/10.12928/telkomnika.v18i3.14840
[15] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video : A high-definition ground truth database," *Pattern Recognit. Lett.*, pp. 1–10, 2008. https://doi.org/10.1016/j.patrec.2008.04.005
[16] C. Yu, J. Wang, C. Peng, C. Gao, and N. Sang, "BiSeNet : Bilateral Segmentation Network for Real-time Semantic Segmentation," in *European COnference on Computer Vision*, 2018, pp. 1–17. https://doi.org/10.1007/978-3-030-01261-8_20
[17] J. Simon, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The One Hundred Layers Tiramisu : Fully Convolutional DenseNets for Semantic Segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 11–19. https://doi.org/10.1109/cvprw.2017.156
[18] M. Siam, S. Elkerdawy, and M. Jagersand, "Deep Semantic Segmentation for Automated Driving : Taxonomy , Roadmap and Challenges," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–8. https://doi.org/10.1109/ITSC.2017.8317714
[19] F. Visin, M. Ciccone, and A. Romero, "ReSeg : A Recurrent Neural Network-based Model for Semantic Segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 41–48. https://doi.org/10.1109/CVPRW.2016.60
[20] V. Badrinarayanan, A. Kendall, R. Cipolla, and S. Member, "SegNet : A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 1–14, 2016. https://doi.org/10.1109/TPAMI.2016.2644615
[21] E. Fernandez-moral, R. Martins, D. Wolf, and P. Rives, "A new metric for evaluating semantic segmentation : leveraging global and contour accuracy," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1–8. https://doi.org/10.1109/IVS.2018.8500497
[22] M. Kampffmeyer, A. Salberg, and R. Jenssen, "Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 1–9. https://doi.org/10.1109/CVPRW.2016.90
[23] D. Eigen and R. Fergus, "Predicting Depth , Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1–9. https://doi.org/10.1109/ICCV.2015.304
[24] R. Dong, X. Pan, and F. Li, "DenseU-Net-Based Semantic Segmentation of Small Objects in Urban Remote Sensing Images," *IEEE Access*, vol. 7, no. June, pp. 65347–65356, 2019. https://doi.org/10.1109/ACCESS.2019.2917952
[25] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets Robotics : The KITTI Dataset," *Int. J. Rob. Res.*, vol. 32, no. 11, pp. 1–6, 2011. https://doi.org/10.1177/0278364913491297
[26] G. Neuhold, T. Ollmann, S. R. Bul, and P. Kontschieder, "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1–10. https://doi.org/10.1109/ICCV.2017.534