A semantic segmentation method using model uncertainty

Shuya Isobe*, Shuichi Arai

Tokyo City University Graduate School of Engineering
1-28-1 Tamazutsumi, Setagaya, Tokyo, Japan 158-8557

*Corresponding Author: isobe15@ipl.cs.tcu.ac.jp

Abstract

This paper proposed a semantic segmentation method using model uncertainty. Recently, many architectures achieved good segmentation performance but the certainty of the segmentation result was not considered. Certainty is needed for practical use of semantic segmentation in scene understanding because it is necessary to recognize accurately whether the object is known or not. Bayesian SegNet makes it possible to produce an uncertainty of the segmentation results using a measure of model uncertainty. However, the uncertainty is not used for segmentation itself. Then, our method rejects the uncertain region and classifies it as an unknown object using the model uncertainty and we aim the improvement of certainty. We implement encoder-decoder type CNN architecture and train with several optimizers such as SGD with momentum, Adam, AdaGrad, and so on. We use CamVid dataset for the training and testing and evaluate our segmentation results using 5 common measures. We achieved improvement of certainty by our method as shown in the evaluation results. Furthermore, optimizer comparison clarifies the best optimizer for our segmentation architecture.

Keywords: Semantic Segmentation, CNN, Certainty.

1. Introduction

Semantic segmentation originating from TextonForest [1] and TextonBoost [2] is a pixel-level labelling and makes it possible to recognize each object in per-class and divide the object region as the shape of the object contour. Recently, convolutional neural networks (CNNs) [3] are widely used for segmentation models. However, many architectures have tried to directly adopt deep convolutional architectures to pixel-wise labelling, so the segmentation results appear coarse. To solve this problem, various

architectures were proposed [4]–[9]. SegNet [9] inspired by the architecture Ranzato et al. proposed [10] is a deep convolutional encoder-decoder network. This architecture improves boundary delineation and reduces trainable parameters enabling end-to-end training. As a result, inference time is shortened so as to inference in real time. It has led to expectations for autonomous driving. Moreover, the accuracy of prediction has improved considerably but this accuracy ignores the regions of unknown objects. It becomes a problem on a practical level so that the it is necessary to recognize accurately whether the object is known or not. Therefore, models should consider the certainty of the segmentation result. Bayesian SegNet [11] which is an architecture that extends SegNet makes it possible to produce an uncertainty of segmentation results with a measure of model uncertainty from the sampling of the posterior distribution of the model using Dropout [12]. However, the uncertainty is not used for segmentation itself. Therefore, we apply the uncertainty to segmentation and aim the improvement of certainty by rejecting the uncertain region by our method. In a certain pixel, if the uncertainty is large, the model considers it as an unknown object and the pixel is not classified into any predefined classes. If the uncertainty is small, the model considers it as the object belonging to a class which has the highest probability among predefined classes. We apply this method on CamVid road scene dataset [13]. Furthermore, we train our network using several optimizers for finding the best optimizer for our segmentation architecture.

2. Network Architecture

In this section, we present our network architecture. Fig.1 shows the our network architecture. Our network is a deep convolutional encoder-decoder network like SegNet [9] and Bayesian SegNet [11]. Encoder network is

composed of 10 convolutional layers, 4 pooling layers, and 2 dropout layers. Decoder network corresponding encoder network is 10 convolutional layers, 4 upsampling layers, and 2 dropout layers. The decoder performs upsampling to restore resolution of input feature maps using the memorized max-pooling indices from the corresponding encoder feature maps. The upsampling expands the resolution of input feature maps using subsampled values and spatial information in max-pooling. This step can obtain high-resolution feature maps but they are sparse. Therefore the decoder obtains the dense high-resolution feature maps by convolution of the sparse feature maps. This is often called deconvolution. The decoder finally outputs the dense feature maps which are the same resolution as the input RGB images. The end of our network is softmax classifier. Rectified Linear Unit (ReLU) and Batch normalization [15] are applied to each convolutional layer.

Bayesian SegNet is composed of 26 convolutional layers, 10 pooling layers, and 6 dropout layers. On the other hand, we implement the smaller network than Bayesian SegNet because we consider that the more convolutional layers are deep, the more difficult it is to restore abstract features in the decoder. Therefore our network composed of 20 convolutional layers, 8 pooling layers, and 4 dropout layers.

Fig. 1. Our network architecture.

3. Segmentation method

In this section, we propose the method of segmentation by inference with the model uncertainty. We use dropout to perform probabilistic inference. Dropout is a way to prevent over-fitting by dropping units randomly. The dropping units randomly has the same meaning as the learning with various models. We use this property as a way of obtaining samples from the posterior distribution of models from Gal and Ghahramani proposed methods [16], [17] like Bayesian SegNet [11]. This method enables to perform probabilistic inference over our segmentation model. We also perform minimizing the cross entropy loss objective function to encourage the model to learn a distribution of weights. We train the model and sample the posterior distribution over the weights at test time using dropout to obtain the posterior distribution of softmax class probabilities. We take the mean of distribution for classification prediction and use the variance to model uncertainty for each class based on Bayesian SegNet [11]. We perform segmentation using the model uncertainty. In a certain pixel, if the variance is larger than the empirically determined threshold, the model considers it as an unknown object and the pixel is not classified into any predefined classes. If the variance is smaller than the threshold, the model considers it as the object belonging to a class which has the highest probability among predefined classes. To our knowledge, there has been no methods to reject the uncertain region like our method for semantic segmentation. Our method is different from some methods which require a large training dataset to let the model learn the regions of unknown objects as a background class.

4. Training

We train our network with the CamVid [13] dataset for inference using our method at test time. We implement our network using the Caffe [19] which is common deep learning framework. CamVid is a scene understanding dataset with road scene images. We use 367 training and 233 testing only RGB images in this dataset and resize these images to 360×480 pixels to keep the condition with the Bayesian SegNet [11]. Classes are 11 objects in the road scene such as sky, building, pole, road, pavement, and so on. We train our network end-to-end using 6 optimizers: stochastic gradient descent (SGD) with momentum [20], AdaGrad [21], AdaDelta [22], Adam [23], NesterovAG [24], and RMSProp [25]. In addition, dropout ratio is 0.5 and we performed training until the cross-entropy training

loss converged on such conditions.

5. Evaluations

In this section, we evaluate improvement of certainty by our proposed segmentation method. Additionally, we compare the performance of several optimizers. Fig. 2 shows the segmentation results obtained from classification by our model. In Bayesian SegNet, the segmentation results are obtained from the mean of distribution like Fig. 2 showed. These results show that all pixels are classified into one of the predefined classes. However, the unlabeled regions exist in ground truth labels. This means that the regions of unknown objects are definitely misclassified as one of the predefined classes. In model uncertainty in Fig. 2, darker colors indicate a larger value of the variance and it means more uncertain predictions. We can see that our method prevents labelling in higher variance pixels as black unlabeled pixels. It means success to the rejection of the uncertain region.

Toward practical use of the semantic segmentation, it is hard to define all objects in the scene as classes, so the segmentation results should have the unlabeled regions having the same meaning as the regions of unknown objects. Our method can infer the region which is not classified into any predefined classes as the unlabeled region. Hence, it is necessary to evaluate the unlabeled regions in ground truth labels. However, the conventional evaluation method ignores the unlabeled regions in ground truth labels and considers only the labeled regions. Therefore, we newly consider the case of misclassification in the unlabeled regions in ground truth labels.

Fig. 2. Segmentation results on the CamVid dataset.

In calculating classification accuracies, we use 5 common measures of evaluation: Global accuracy (Global acc) in Eq. 1 measures the percentage of pixels correctly classified by the division of the total number of pixels of true positive prediction (TP) and the total number of pixels of ground truth (GT).

$$Global\ acc = \frac{TP}{GT} \quad (1)$$

Class accuracy (Class acc) in Eq. 2 measures the percentage of pixels correctly classified in a class i .

$$Class\ acc_i = \frac{TP_i}{GT_i} \quad (2)$$

Class average accuracy (Class avg) in Eq. 3 is the mean of the predictive accuracy over all classes by the division of the sum total of class accuracy in all classes and the number of classes (C).

$$Class\ avg = \frac{\sum_i^C Class\ acc_i}{C} \quad (3)$$

Intersection over union (IoU) in Eq. 4 is a measure which imposes the penalty of false positive predictions (FP) on the class accuracy in a class i .

$$IoU_i = \frac{TP_i}{GT_i + FP_i} \quad (4)$$

Mean intersection over union (Mean IoU) in Eq. 5 is the mean of intersection over union in all classes.

$$Mean\ IoU = \frac{\sum_i^C IoU_i}{C} \quad (5)$$

First, we evaluate the segmentation results only in the labeled regions in ground truth labels like the conventional evaluation method to verify whether or not our method reduces the case of the mismatch of the predicted class label and the ground truth class label in a pixel. The results of the classification accuracy using 3 measures of evaluation are shown in Table 1. Table 1 shows that our method results exceeded conventional results obtained from the mean of distribution in all 3 accuracy scores. These evaluation results mean that we succeeded in reducing wrong predictions in this case.

Table 1. Evaluation results of our method in labeled regions.

	Global Acc	Class avg	Mean IoU
Conventional method	87.8	74.7	58.2
Our method	91.6	80.1	66.2

Table 2. Evaluation results of our method in unlabeled regions.

	Global acc
Conventional method	0.0
Our method	31.1

Secondly, we also evaluate the segmentation results only in the unlabeled regions in ground truth labels to verify whether or not our method reduces the case of the misclassification in an unlabeled pixel of the ground truth label. We regard the unlabeled pixels as the unknown class in this evaluation and if a classification result in a pixel is an unknown class, this result is evaluated as a true positive. It is not necessary to evaluate using except global accuracy because the number of correct classes is only one unknown class. Table 2 shows the result of the classification accuracy. We succeeded in reducing misclassifications in this case because the accuracy was improved from 0 % to 31.1 % by making it possible to reject the uncertain regions using our method. The accuracy of 31.1 % is far from the high score. However as we have shown in the end of Section 3, our model has avoided learning of the regions of unknown objects so this score is appropriate. From these evaluation results, we succeeded in the improvement of certainty by our method. Note that we used RMSProp for the training in these evaluation of our segmentation method.

Thirdly, we compare the performance of several optimizers. Fig. 3 shows the comparison results of the maximum accuracy score of each optimizer in training term. We can see that Adam shows better accuracy score than other methods. RMSProp is close to the score of Adam. In addition, the comparison results of the standard deviation of each accuracy in Fig. 4 shows that RMSProp is the most stable optimizer. Altogether, RMSProp is the best optimizer for our segmentation architecture from these comparison results.

Finally, we challenge the Pascal VOC12 [26] dataset. This dataset consists of segmenting a few salient object classes from a widely varying background class. It is unlike the segmentation for scene understanding but widely used for benchmarks of the object recognition. We applied our segmentation method to pre-trained model provided by SegNet. Fig. 5 shows the segmentation results by our

method. We can see that more frequently occurring classes have higher certainty but other classes have lower certainty and most of the regions are classified as unknown objects.

Fig.3. Optimizer comparison of maximum accuracy.

Fig. 4. Optimizer comparison of standard deviation.

Fig. 5. Segmentation results on the Pascal VOC12 dataset.

6. Conclusions

This paper proposed a new semantic segmentation method and the necessity of certainty for practical use of semantic segmentation in scene understanding. We aim the improvement of certainty using model uncertainty on the road scene for semantic segmentation. We implement a encoder-decoder type CNN architecture and use CamVid dataset for the training and testing of our network. Our segmentation results showed that our method can reject the uncertain region where the variance of the posterior distribution of the model is large. We succeeded in reducing wrong predictions and achieved improvement of certainty by our method as shown in the evaluation results. Furthermore, optimizer comparison made it clear that RMSProp is the best optimizer for our segmentation architecture.

References

- [1] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," CVPR, 2008.
- [2] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Texonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," IJCV, 2009.
- [3] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," ECCV, 2014.
- [4] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," ICCV, 2015.
- [5] S. Hong, H. Noh, and B. Han, "Decoupled deep neural network for semi-supervised semantic segmentation," NIPS, 2015.
- [6] C. L. Zitnick and P. Dollar, "Edge boxes: Locating object proposals from edges," ECCV, 2014.
- [7] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille, "Weakly and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation," ICCV, 2015.
- [8] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr, "Conditional random fields as recurrent neural networks," ICCV, 2015.
- [9] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," CVPR, 2015.
- [10] M. Ranzato, F. J. Huang, Y. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," CVPR, 2007.
- [11] V. Badrinarayanan, A. Handa, and R. Cipolla, "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding," arXiv preprint arXiv:1511.02680, 2016.
- [12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," JMLR, 2014.
- [13] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," Pattern Recognition Letters, 30(2):8897, 2009.
- [14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," CVPR 2015.
- [15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," ICML, 2015.
- [16] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," ICLR, 2016.
- [17] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," ICML, 2016.
- [18] A. Graves, "Practical variational inference for neural networks," NIPS, 2011.
- [19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," ACM MULTIMEDIA, 2014.
- [20] Amari, Shunichi, "A theory of adaptive pattern classifiers." IEEE Transactions on Electronic Computers 3 (1967): 299-307.
- [21] Duchi, John, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." Journal of Machine Learning Research 12: Jul (2011): 2121-2159.
- [22] Zeiler, Matthew D. "ADADELTA: an adaptive learning rate method." arXiv preprint arXiv:1212.5701, 2012.
- [23] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv: 1412.6980, 2014.
- [24] Y. Bengio, N. Boulanger, and R. Pascanu, "Advances in optimizing recurrent networks." In Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP), 2013.
- [25] Tijmen Tieleman; G. Hinton (2012). Lecture 6.5 - rmsprop, COURSE: Neural Networks for Machine Learning.
- [26] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303-338, 2010.