ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 6, Issue 6, November 2017

Using Deep Learning and Gaussian Mixture Models for Road Scene Segmentation

Yu-Chen Lin[a,*], Peng-Cheng Chen[b], Cheng-Hsien Wang[c],

Abstract*—Road scene segmentation play an important role in computer vision for different applications such as autonomous driving and obstacle detection. Recently, a number of deep learning schemes have demonstrated superior results on image classification and pixel-labeling segmentation tasks. However, they usually require lots of computational resources and accurate estimation of the pixel labels are not easily. In this paper, we present a novel pixel labels approach for road scene understanding including the region-based segmentation algorithm based on Gaussian mixture models (GMMs) modeling, and region merging scheme by using convolution neural networks (CNNs). Firstly, the region-based segmentation algorithm involves two parts: one to perform the K-means clustering scheme to group a dataset into a user-defined number of k-clusters in L*a*b* feature space. Next, GMMs are utilized to secondary classify for the results of k-clusters; meanwhile, the expectation maximization (EM) algorithm is applied to estimate the optimum parameters of GMMs. Hence, the labeling results of foreground image can be obtained. In order to avoid over-segmentation adjacent segments, a deep-learning method based on convolutional neural networks (CNNs) is used to merge highly overlapping regions for the previous labeling results. Thus, drivable road region and the target objects can be easily derived. This proposed approach not only promote computational efficiency through combination through region-based segmentation algorithm and region merging scheme based on deep learning, but also improve accuracy for road scene recognition. Extensive experiments demonstrate that the proposed approaches can significantly extract road region and objects markings.

***Index Terms*—Road scene segmentation, Gaussian mixture models (GMMs), convolution neural networks (CNNs), drivable road region, region merging.**

I. INTRODUCTION

Recently, drivable road understanding is necessary for the function of numerous advanced driver-assistance systems (ADAS), even autonomous driving systems. The detection of drivable road have increasingly become complex challenge because of the targeted scenarios increasingly became difficult. A number of algorithms for lane marking detection [1]-[3] by a monocular camera, such as lane departure warning and lane keeping systems, are usually used to decide where the suitable drivable region is for host vehicle and when to warn with the driver. However, the lane mark geometric features are not necessarily always present or consistency may be sheltered by front vehicles, roadside parked cars, presence of shadows neighboring vehicles or surface irregularities. In addition, the poor weather conditions, such as rain, fog, snow, are decreases the accuracy on lane detection. The very nature of these problems will makes drivable road estimation difficult.

The scene labeling problem have been developed in many different methods [4]-[11] in recent years. Y. Yu et al. [5] proposes an algorithm for detecting road scene from 3-D mobile-Laser-scanning point cloud data. In addition, some publications are used the Laser or radar sensor [6], [7] to measure the obstacle and road boundary under different environmental conditions. A radar-based road boundaries detection approach [7] is presented. This paper used a deformable template model of the road shape and a Metropolis algorithm is applied to fit the deformable template to the road edge features in the radar image. However, sensor-based road detection schemes have some disadvantages. For example, visible spectrum and infrared imaging cannot provide precise depth information and they are too sensitive to light and weather. The recognition rate is still unsatisfactory in terms of more complex road conditions such as unconstructed road, complicated weather conditions and low-light environment. In addition, the price range of such sensors is still too high for a mass produced commercial item. A hierarchical approach for labeling semantic objects and regions in scenes is presented [8]. Based on the graphical model formulation for structured prediction, the experiment demonstrates in this paper is able to easily capture the multi-class segmentation and make significant results without a probabilistic model over global configurations. Moreover, a segmentation and object recognition algorithm [9] based on ego-motion derived 3-D point clouds is proposed. C. Wojek et al. [10] presented a probabilistic 3D scene model based on a single camera which enables multi-frame tracklet inference on a scene level for multi-class object detection, object tracking, and scene labeling. D. A.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 6, Issue 6, November 2017

Chacra [11] proposed a road segmentation algorithm using texture information to classify roads in natural images. However, most methods mentioned above heavily depend on firstly segment images into local regions according to hand-coded features, such as scale-invariant feature transform (SIFT) [12], speeded up robust features (SURF) [13] and histogram of oriented gradients (HOG) [14] are applied to extract visual information from these segmented regions or combinations of adjacent segmented local regions, and label these image segments by using conditional random fields (CRF) [15], [16] or Markov random fields (MRF) [17], [18]. In fact, in real-world scenarios, it is rarely known what features are applicable for the task at hand-coded since the choice of features are very difficult. Besides, these approaches need a lot of extra memory to store the feature images and is difficult to implement on embedded system.

In order to deal with the previously mentioned problems and challenges, a number of deep network architecture [19]-[21] were proposed in the literature, such as convolutional neural network (CNN) [20], fully convolutional neural network [21] and so on. However, these procedures demand high computation complexity and a large quantity memory requirement. Therefore, to tackle the highly complicated road detection task and to improve reliability of our proposed systems, we present a new vision-based road scene semantic segmentation algorithm that combines image segmentation algorithms for region segmentation and deep learning technology for region merging. Effectiveness of the proposed pixel labels approach for road scene understanding is demonstrated by extensive verifications during different road scenes.

II. SYSTEM DESCRIPTION

The aim is to present how we have dealt with drivable road region and movement/stationary objects recognition using a single wide-angle camera mounted on a vehicle. The complete flowchart of our proposed road scene semantic segmentation approach for road scene understanding is shown in Fig. 1. In this algorithm, the camera is assumed to be calibrated beforehand. Therefore, the intrinsic and extrinsic parameters are obtained. It consists of two main parts: region segmentation and region merging. In the first part, the region-based segmentation methods, such as K-means, Gaussian mixture models (GMMs), and expectation maximization (EM) scheme, are used to derive the labeling results of foreground image. Next, the deep learning technology by using the deep convolutional neural networks (CNNs) architecture is applied to merge the before labeling results. Thus, the boundaries of road region and the target objects can be easily derived.

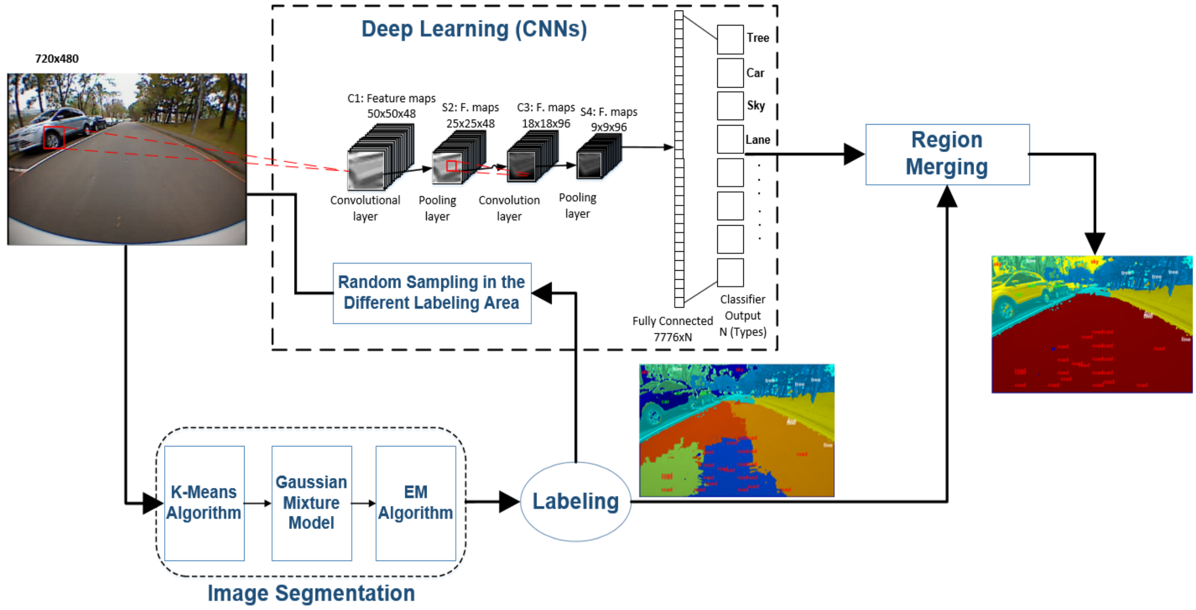


Fig 1. Architecture of the proposed road scene semantic segmentation

A. Region-Based Image Segmentation

Image region-based segmentation scheme is used to separate the variety of different features of the environment region, which features involve the intensity, color, texture and so on. For the challenge of road segmentation that is to decide for each pixel if it is road or not. In this paper, the K-means clustering scheme is perform to group a dataset into a user-defined number of k-clusters in L*a*b* feature space. Then, the Gaussian mixture models (GMMs) are used to model the original image color distribution for the results of k -clusters; meanwhile, the expectation maximization (EM) algorithm is applied to estimate the optimum parameters of Gaussian mixture models. Finally, the labeling results of foreground image can be derived.

K-means clustering algorithm

The K-means clustering algorithm [22], [23] is a popular unsupervised learning method that is used to primary segment the k number of disjoint cluster from the road scene images. We perform K-means on a number of k-cluster color images in L*a*b* feature space. The detailed steps of K-means clustering scheme are as follows:

1. Every image pixel value needs to be normalized by the following formula:

$$Z_{x,y} = \frac{p_{x,y} - \mu_i}{\sigma} \tag{1}$$

Where $Z_{x,y}$ is the value of standardized pixel, $p_{x,y}$ is the values of each pixel in original image, μ_i is the mean of the image pixel, and σ is the standard deviation of the image pixel.

2. Initialize the number of cluster k and center.
3. Compute the Euclidean distance between the obtained cluster centers and each pixel of an image, which formula can be expressed as:

$$D_i = \arg\min\left(\sqrt{(x-\mu_i)^2 + (y-\mu_i)^2}\right) \tag{2}$$

Where x, y is the coordinates of each pixel, $\mu_i, i = 1,...,\mathrm{k}$ is the centers of the each segmentation region, and D_i is Euclidean distance of each pixel with all regional centers.

4. Update the regional center and repeat the process until it satisfies the defined iteration number or the displacement between old and new regional center less than the given value. Thus, the image can then be segmented the k clusters.

Combine the Gaussian mixture models to perform the secondary region segmentation, which is based on the results of K-means clustering.

Gaussian mixture models (GMMs)

Gaussian mixture models (GMMs) are often used for data clustering [24], which not only accurately describe the sample categories, position in space, but also characterize these categories in the sample space of size and shape. Generally, the GMMs need to randomly choose an initialize number of cluster k and sample from its distribution. However, we utilize the result by the previous K-means clustering phase to regards as the parameter initialization of GMMs procedure. The purpose of GMMs is to maximize the likelihood function with respect to the parameters, such as the means and covariance of the components and the mixing coefficients. Assuming that each clustering region after the K-means clustering algorithm is a Gaussian normal distribution, the mean and the standard deviation of each region, and all the weighting areas are calculated to construct the Gaussian mixture models. Figure 2 illustrates the GMMs approximation distribution diagram. The blue curve means the intensity distribution of original image, and the red curves are the fitting result by eight Gaussian distribution to approximate the blue curve.

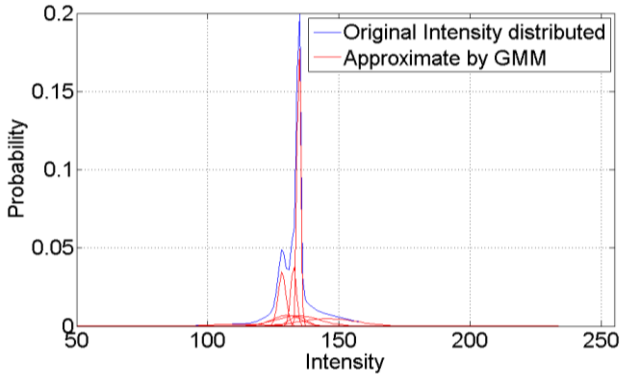


Fig 2. Approximation of a standard Gaussian density through a Gaussian mixture consisting of eight components.

The multi-variate Gaussian mixture distribution can be defined as follows:

$$P(\boldsymbol{x}) = \sum_{i=1}^{k} w_i \mathcal{N}(\boldsymbol{x} | \zeta_i, \Sigma_i) \tag{3}$$

Where

$$\mathcal{N}(\boldsymbol{x} | \zeta_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^d \, |\Sigma_i|}} \exp[-\frac{1}{2}(\boldsymbol{x}-\zeta_i)^T \Sigma_i^{-1}(\boldsymbol{x}-\zeta_i)]$$

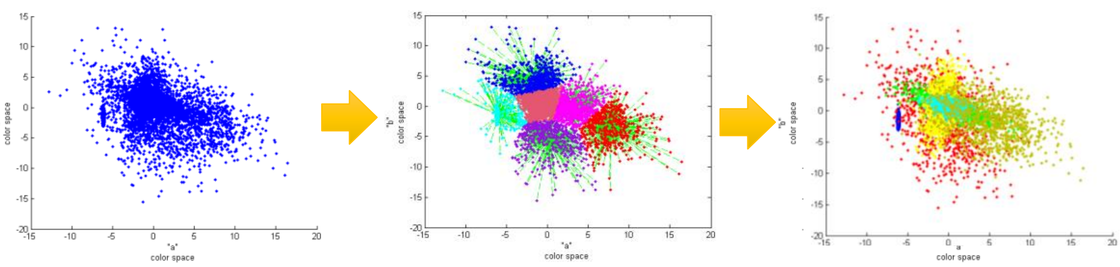
Here $\boldsymbol{x}$ is a continuous-valued data vector, $\boldsymbol{x} = \{x_1, ..., x_n\}$ (i.e. measurement or features); w_i, $i = 1, ..., k$ are the corresponding weight and the mixture weights satisfy the constraint that $\sum_{i=1}^{k} w_i = 1$. $\mathcal{N}(\boldsymbol{x} | \zeta_i, \Sigma_i)$, $i = 1, \ldots, k$ are the component Gaussian densities, where ζ_i and Σ_i represent the mean vector and covariance matrix of the i^{th} Gaussian, respectively. The complete Gaussian mixture model is parameterized by using the mean vectors, covariance matrices and mixture weights from all component densities. For the dimension of feature vectors (i.e. regional dimensions of data), the Gaussian probability density function can be utilized to describe each pixel of the probability density function. However, in the image area segmentation algorithms, it is indispensable to estimate the optimal GMMs parameters, such as regional center position, the number of regional members. Thus, the parameters of GMMs can be estimated using expectation maximization (EM) technique.

Expectation-maximization (EM) algorithm

The image segmentation area can be obtained by using Gaussian mixture models. The Gaussian mixture models needs to calculate the members of each pixel colors in the area and a probability value, then to find the probability of each pixel maximum (maximum likelihood) which belongs to a zone. Although the calculated probability value represents the probability of a pixel matching the region in the region in line with the number of chances, it is not exactly to calculate the pixel belonging to a region in the image. Therefore, the expectation-maximization (EM) algorithm [24], [25] is an iterative optimization technique which repeatedly employed to estimate the parameters in each region of GMMs.

In (3), the parameters of ζ_i and Σ_i for each cluster are taken as the initial values of the EM algorithm, and then the maximum value expectation algorithm is used to estimate a set of new parameter values. When the probability of the new parameter value converges to the defined value or the number of iterations reaches the upper limit, the calculation process will stop. The maximum value of the algorithm is divided into two parts. Firstly, the part of the estimation step (E-step) is to evaluate the responsibilities of the latent variable for given parameter values; the other part of the maximization step (M-step) aims to determine the probability of maximizing the parameters and updates the parameters of our model based on responsibilities of the latent variable.

Figure 3 shows the resulting segments using K-means and GMMs with EM approaches for Lab color space. Firstly, we convert the original RGB image into the Lab color space and extract the color space of a and b form. Then, K-means clustering algorithm is applied to obtain the result of means ζ_i, covariance matrices Σ_i and mixing coefficients w_i, considered as initialization parameters for the next stage of GMMs-EM. Therefore, the accurate clustering results of foreground image can be derived.



Original pixel distribution **K-Means** **GMMs-EM**

Fig 3. Clustering results using K-Means and GMMs-EM

B. Deep Learning for Region Merging

A convolutional neural network (CNNs) is a type of artificial neural network which have wide applications in image and video recognition, recommender systems and natural language processing. Thus we employ a CNNs

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 6, Issue 6, November 2017

consisting of a stack of multiple training stages to extract the features hierarchically. Traditional neural networks have the full connectivity between nodes thus they cannot scale well to higher resolution images because of the large dimensionality. However, the CNNs is a form of sparse connectivity and weightings sharing, which reduces the total number of free parameters of the neural network, and makes them computationally applicable to model images. The typical CNNs architecture may include various distinct layers such as convolutional layer, pooling layer and fully connected layers.

Convolutional layer

The convolutional layer aims to extract local patterns. In CNNs, feature filters are initialized as random templates and will be learned to be edge, color, and specific patterns' detectors, which are similar to the basis of sparse coding and hidden neuron's weightings of an auto-encoder. In convolutional layer, the filter comprises weightings and bias. A kernel will convolve the input image as a sliding window moving along the x-axes and y-axes. In a specific location, an inner product is calculated using the filter matrix and the corresponding elements of the input image, and the output feature map's element is inner product add bias. Based on convolving the input image with different filter (equal to weightings), several feature maps can be generated as follows

$$y_n^l = f^l\left(x_m \otimes w_{m,n}^l + b_n^l\right) \tag{4}$$

where $l = 1,...,K$; K is the number of convolutional layers, n is the output number of feature maps, x_m is the input image, f^l is nonlinear activation function which is defined as $f(x) = 1/1+e^{-x}$. The filter $w_{m,n}^l$ is randomly initialized at first and is then trained with a well-known back-propagation neural network.

Pooling layer

The convolutional layer role is to extract local features of a training image. When performing the classifying procedure, the precise positions of the features are detrimental to the performance, because different training instances in the same label have different precise positions. In order to solve this problem, a reasonable method, pooling in a CNNs, is subsampling the features to decrease feature maps' resolution.

For the pooling layer, it has the same number of planes as the preceding convolution layer. A subsampling plane divides its 2-D input into non-overlapping blocks of size 2×2 pixels. For each block, the mean of four pixels can be calculated. Clearly, each sub-sampling plane reduces its input size by half, along each dimension. A feature map in a sub-sampling layer is connected to one or more planes in the next convolution layer. It's performed on the corresponding feature map in convolution layer as:

$$y_k^{l+1} = f^{l+1}\left(pool\left(y_n^l\right) \otimes w_{n,k}^l + b_k^l\right) \tag{5}$$

Where $pool$ means the average pooling method, $w_{n,k}^l$ is the weightings of pooling layer, and b_k^l is bias of pooling layer. The input size is reduced by half, along each dimension in the pooling plane. Usually, after the convolution layer and pooling layer, the feature maps are vectorization, and input to one or more fully-connected layers.

- *Fully-connected layer*

Define a convolutional layer and a pooling layer as a stage, thus the CNNs model usually has several stages. After having passed through several stages, original images are converted to lots of low-resolution feature maps. These small-size feature maps are concatenated into a vector. Such a vector plays the same role as hand-coded features and fed it to a hidden layer, the feature vector is fully connected to the hidden layer is called fully-connected layer which defined as:

$$y_p^{out} = f^F\left(y_k^{l+1} \times w_{k,p}^F + b_p^F\right) \tag{6}$$

Where y_k^{l+1} is a vector which is concatenated from last layer, $w_{k,p}^F$ is the weight of fully-connected layer, b_p^F is bias of fully-connected layer, and y_n^{out} is the output of CNNs, same as classifier results?

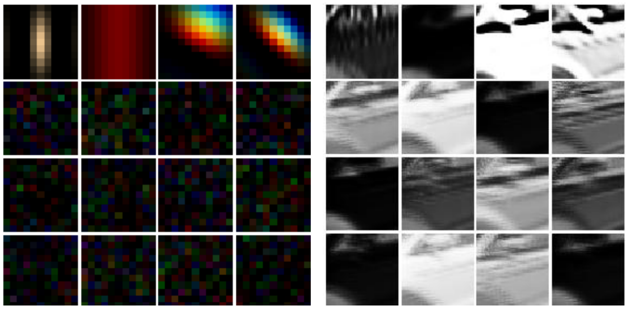
The CNNs feature detector can be supplied with raw pixels to automatically learn low-level and midlevel features of stacked structures, alleviating the need for hand-engineered features and improving the recognition accuracy. The CNNs is perform end-to-end feature learning and are trained with back-propagation algorithm.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 6, Issue 6, November 2017

C. CNNs Training

The input image is firstly segmented into local regions using GMMs-EM according to its pixel-level information. Then the training image samples are randomly cropped from each segmented local region. The cropped images are fed to train convolutional neural network. The label of each segmented local region is finally determined by averaging the predictions which is made by the network's final layer on its k cropped samples. The CNNs model include two convolutional layer, two pooling layer and fully-connected layer. The first convolution layers of the network are composed of a bank of filters of size 11×11 followed by sigmoid units and 2×2 average-pooling operations. The second convolution layers takes as input the output of first convolution layer, and the filters of size 8×8 followed by sigmoid units and 2×2 max-pooling operations. Finally, the pooling layer's result is concatenated into a 7776 vector, then connected with n-class hidden units, also called fully-connected layer. The input training images ($60\times60\times3$) are shown in Fig. 4, and then connect it to first convolution layer with forty-eight filter size of 11×11 without padding, so the size of 50×50 of forty-eight feature map are obtained as shown in Fig. 5.

Fig 4. Training data



(a) (b)

Fig 5. Examples of filter and feature maps. (a)Some convolutional filter of size $11\times11\times3$ learned by the 1^{st} convolution layer; (b) 1^{st} convolution layer's feature maps

Next, pooling layer are applied to the 2D feature maps, reducing the size of these 2D feature maps to 25×25. Through the same convolution procedure, ninety-six feature map size of 50×50 are obtained.

In this paper, the backpropagation procedure as a learning algorithm to compute a gradient descent with respect to weights for CNNs model. The main concept of back-propagation is to minimize the cross-entropy loss function, and to learning an optimal weighting parameters in CNNs model. The cross entropy loss function is defined as:

$$E = \frac{1}{s}\sum_{c}\left[T_n \ln y_n^l + (1-T_n)\ln\left(1-y_n^l\right)\right] \tag{7}$$

where s is the number of class, c is number of CNNs' output, T_n is the ground truth label, y_n^l is CNNs' output, and n is the nth neuron in fully-connected layer. The weighting parameters w are updated by gradient descent method to comprise a small step in the direction of the negative gradient, so that:

$$\hat{w}_{i,j}^l = w_{i,j}^l + \Delta w_{i,j},\ \Delta w = -\eta\frac{\partial E}{\partial w_{i,j}} \tag{8}$$

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 6, Issue 6, November 2017

Where Δw is momentum variable, η is learning rate, and $\partial E / \partial w_{i,j}$ is the derivative which is computed by back-propagation error. After each such update, the gradient is re-evaluated for the new weighting vector and the process repeated. The gradient-based learning technique is used to train all the weightings for CNNs so that the minimum loss function can be derived.

III. EXPERIMENT RESULTS

According our proposed systems configuration of Fig. 1. The input image sequences is real road scene, which is grabbed from the vision sensors with the 720×480 resolution and 8 bits in each color channel (RGB) as shown in Fig. 6. The image has 5 categories, namely car, road, lane-line, tree and sky (background).

Fig 6. Original road scene images

First, segmented the local image region of input image by using K-means and GMMs scheme, the initial parameter for GMMs was from the K-means method, such as the number of clustering or the centre of clustering and so on. After the initialization parameters, and the first image segmentation result is obtained by GMMs as shown in Fig. 7.

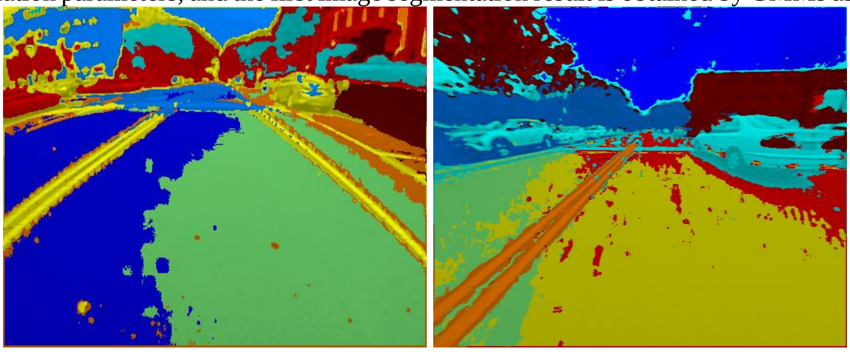


Fig 7. Image segmentation results by using GMMs-EM

The labeling results of 1000 testing images are shown in Table. 1. We can observe that can obtain best overall accuracy value of 98.1% for road scene semantic segmentation when we consider the Filter size of the 1st layer is 11, filter size of the 2nd layer is 8, filter number of the 1st layer is 48 and filter number of the 2nd layer is 96.

Table 1: Road scene semantic segmentation performances of the different filter size and filter number

Filter size of the 1st layer	Filter size of the 2nd layer	Filter number of the 1st layer	Filter number of the 2nd layer	Learning rate	Accuracy (%)
9	5	48	64	0.0001	95.4
9	5	48	96	0.0001	96.7
11	5	48	64	0.0001	96.9
11	5	48	96	0.0001	97.3
11	8	48	64	0.0001	97.9
11	8	48	96	0.0001	98.1
17	8	48	64	0.0001	93.9

Figure 8 shows the road scene semantic segmentation results. The label of each local region segmented using the image segmentation algorithms are determined by the average probability scores of its cropped samples given by

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 6, Issue 6, November 2017

the convolutional neural network. According to the area of each segmented local region, to select the number of randomly cropped image. All randomly cropped image as input the trained convolutional neural networks, and the probability score is given by convolutional neural networks result. Then, the local segmented image will be merged, which has the same label. As shown in Fig. 8, the drivable road region and the target objects can significantly be identified and good segmentation of the entire image.

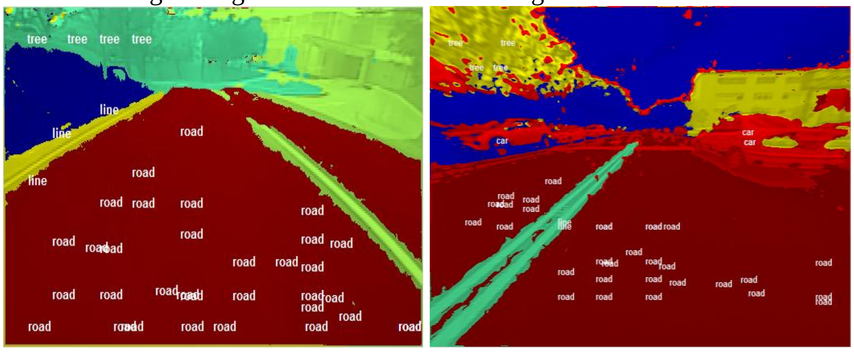


Fig 8. Road scene semantic segmentation results

IV. CONCLUSIONS

This paper presents a novel pixel labels approach for road scene understanding (drivable road region and objects detection), including the region-based segmentation algorithm based on GMMs-EM, and region merging scheme by using convolution neural networks (CNNs). A 5-layer convolutional neural networks is trained with a huge amount of data of data to emulate the road scene. The boundaries of road region and the target objects can be easily decided. This proposed approach not only doesn't need to extract features from the objects via extra image recognize method, but also improve accuracy for road scene recognition. Extensive experiments demonstrate that the proposed approaches can significantly extract road region, boundaries and other objects. The future work will be research into the new deep network methods to enhance moving objects recognition, segmentation accuracy and quantitative analysis and other detection algorithms comparison.

ACKNOWLEDGMENT

This research was sponsored by Ministry of Science and Technology, Taiwan, R.O.C., under the contract No.: 106-2628-E-035 -001 -MY3.

REFERENCES

[1] D. O. Cualain, C. Hughes, M. Glavin, and E. Jones, "Automotive standards-grade lane departure warning system," IET Intelligent Transport Systems, vol. 6, no. 1, pp. 44-57, March 2012.

[2] C. F. Wu1, C. J. Lin, , H. Y. Lin, and H. Chung, "Adjacent lane detection and lateral vehicle distance measurement using vision-based neuro-fuzzy approaches," Journal of Applied Research and Technology, vol. 11, no. 2, pp. 251-258, April 2013.

[3] N. Mechat, N. Saadia, N. K. M'Sirdi, and N. Djelal, "Lane detection and tracking by monocular vision system in road vehicle," IEEE International Cong. Image and Signal Processing, pp. 1276-1282, February 2012.

[4] A. Laddha, "Road detection and semantic segmentation without strong human supervision," Robotics Institute Carnegie Mellon University, Master Thesis, July 2016.

[5] Y. Yu, H. Guan, C. Wang, and C. Wen, "Bag of contextual-visual words for road scene object detection from mobile laser scanning data," IEEE Trans. Intelligent Transportation Systems, vol. 17, no. 12, pp. 3391-3406, April 2016.

[6] M. Cho, J. An, W. Jang, and E.Kim, "Histogram-model based road boundary estimation by using laser scanner," IEEE International. Conf. Control, Automation and Systems, pp. 1653-1655, October 2016.

[7] K. Kaliyaperumal, S. Lakshmanan, and K. Kluge, "An algorithm for detecting roads and obstacles in radar images," IEEE Trans. Vehicular Technology, vol. 50, no. 1, pp. 170-182, January 2001.

[8] D. Munoz, J. Bagnell, and M. Hebert, "Stacked hierarchical labeling," in Proc. Eur. Conf. Computer Vision, pp. 57-70, July 2010.

[9] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, “Segmentation and recognition using structure from motion point clouds,” in Proc. Eur. Conf. Computer Vision, pp. 44–57, January 2008.

[10] C. Wojek, S. Roth, K. Schindler, and B. Schiele, “Monocular 3D scene modeling and inference: understanding multi-object traffic scenes,” in Proc. European Conf. Computer Vision, pp. 467-481, January 2010.

[11] D. A. Chacra, and J. Zelek, “Road segmentation in street view images using texture information,” International Conf. Computer and Robot Vision, pp. 424-431, June 2016.

[12] J. Chu, Y. Zhang, G. Zhao, and L. Wang, “A novel scene descriptor and outdoor scene recognition method,” IEEE International Conf. Intelligent Systems and Knowledge Engineering, pp. 374-381, November 2015.

[13] J. Duan, and M. Viktor, “Real time road edges detection and road signs recognition,” IEEE International Conf. Control Automation and Information Sciences, pp. 107-112, October 2015.

[14] L. Zhang, E. Wu, “A road segmentation and road type identification approach based on new-type histogram calculation,” IEEE International Cong. Image and Signal Processing, pp. 1-5, October 2009.

[15] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning hierarchical features for scene labeling,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1915-1929, August 2013.

[16] L. Ladicky, C. Russell, P. Kohli, and P. Torr, “Associative hierarchical CRFs for object class image segmentation,” IET Intelligent Transport Systems, vol. 5, no. 1,pp. 739-746, October 2009.

[17] C. Z. Guo, S. Mita, and D. McAllester, “Robust road detection and tracking in challenging scenarios based on Markov random fields with unsupervised learning,” IEEE Trans. Intelligent Transportation Systems, vol. 13, no. 3, pp. 1338-1354, September 2012.

[18] D. A. Chacra, and J. Zelek, “Road segmentation in street view images using texture information,” International Conf. Computer and Robot Vision, pp. 424-431, June 2016.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient based learning applied to document recognition,” in Proc. the IEEE, vol. 86, no. 11, pp. 2278-2324, November 1998.

[20] D. Strigl, K. Kofler, and S. Podlipnig, “Performance and scalability of GPU-based convolutional neural networks,” 18th Euro micro International Conf. Parallel, Distributed and Network-Based Processing, pp. 317-324, February 2010.

[21] J. Long, E. Shelhamer, T. Darrell, “Fully convolutional networks for semantic segmentation,” IEEE Int. Pattern Analysis and Machine Intelligence, pp. 640-651, April 2015.

[22] J. A. Hartigan, and M. A. Wong, “A K-means clustering algorithm,” J. Royal Statistical Society, Series C., vol. 28, no. 1, pp. 100-108, 1979.

[23] C. Adam, and N. Y. Andrew, “Learning feature representations with K-means,” Neural Networks: Tricks of the Trade, Springer, pp. 561-580, 2012.

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, “Gaussian mixture models and K-means clustering,” Numerical Recipes: The Art of Scientific Computing 3rd. New York: Cambridge University Press, 2007.

[25] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” J. Royal Statistical Society, Series C., vol. 39, no. 1, pp. 1-38, 1977.

AUTHOR BIOGRAPHY

Yu-Chen Lin, was born in Taipei, Taiwan, R.O.C., in 1981. He received the Ph.D. degree in electrical engineering from the National Chung Hsing University, Taichung, in 2009. He was a Researcher with the department of safety sensing and control, intelligent mobility technology division, mechanical and systems research laboratories, Industrial Technology Research Institute, Hsinchu, Taiwan, from 2009 to 2014. He was an Assistant Professor in the department of automatic control engineering at Feng Chia University, Taichung, Taiwan, from 2014 to 2017. He is currently an Associate Professor with the department of automatic control engineering, Feng Chia University, Taichung, Taiwan. His research interests include computer vision, image processing, deep learning, optimal control, intelligent control, robust control, vehicle active safety and vehicle control.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 6, Issue 6, November 2017

Peng-Cheng Chen, was born in Taipei, Taiwan, R.O.C., in 1994. He received the B.S. degree in automatic control engineering from the Feng Chia University, Taichung, in 2016. He is currently working toward the M.A. degree in electrical engineering at the Feng Chia University, Taichung, Taiwan. His research interests include computer vision, deep learning, reinforcement learning and embedded system.

Cheng-Hsien Wang, was born in Taichung, Taiwan, R.O.C., in 1993. He received the B.S. degree in automatic control engineering from the Feng Chia University, Taichung, Taiwan, in 2015 and the M.S. degree in automatic control engineering from Feng Chia University, Taichung, Taiwan, in 2017. His research interests include image processing, computer vision, deep learning and intelligent control.