IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 9, 2019, accepted November 22, 2019, date of publication December 2, 2019, date of current version December 13, 2019.
Digital Object Identifier 10.1109/ACCESS.2019.2957053

Improved Lane Detection With Multilevel Features in Branch Convolutional Neural Networks

WEI-JONG YANG, YOA-TENG CHENG, AND PAU-CHOO CHUNG, (Fellow, IEEE)
Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, Tainan 701, Taiwan
Corresponding author: Wei-Jong Yang (weijongx@hotmail.com.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 108-2218-E-006-062.

ABSTRACT Existing smart vehicles heavily depend on success of precise positioning, optical radar, visual detection and recognition to determine their road conditions and perfect routes. The visual-based approach is the simplest and most effective enabling technology to reach the goal. In performing such an overtaking maneuver, this vision-based driving assistant system must be able to precisely recognize the lane marks first. The traditional approach needs a classifier with hand-crafted features and an adjusted threshold to achieve a robust lane detection in various environment conditions. In this paper, we adopt the deep learning approach to achieve the robust lane detection. The proposed lane detection network inspired by LaneNet model, which uses semantic segmentation concepts utilizes multiple level features of the encoder and designs enhanced binary segmentation and reduced pixel embedding branches. By reduction of computation in decoders, the proposed network effectively utilizes multilevel features to precisely predict the high quality lane maps. The experiments on Tusimple and CuLane datasets verify that the proposed lane detection network achieves better accuracy performance than LaneNet and faster computation than the existed methods for real-time applications.

INDEX TERMS Lane detection, neural networks, image segmentation, pixel embedding, artificial intelligence, feature pyramid network.

I. INTRODUCTION

In recent decades, the developments of autonomous driving technologies have received a lot of attentions. The vision-based detection approach is treated as an effective tool to assist autonomous vehicles. For driving assistance, we must be able to accurately detect lane marking to guide their routes first. Thus, numerous lane detection methods have been proposed recently [1]–[27]. Generally, the lane detection methods can be divided into traditional and deep learning approaches.

The traditional lane detection approach usually needs to combine several image processing steps, including extraction of hand-crafted features, detection of joint associations and fusion lane segments to achieve a reliable lane detection. Most intuitive methods [1]–[5] utilize the cues of colors to retrieve the lane lines. Chin and Lin [6] proposed color-based segmentation for lane detection. By choosing a region of interest (ROI), we can reduce computation and calculate an adaptive threshold to detect the lane boundary [7]. A non-parametric regression [8] is also proposed to predict the color information for detecting the lanes. However, the RGB-based classifications could face degradation problems under contrast and illumination variations, the improvements of color-based lane detection are proposed in [9]–[11]. Except RGB color space, the lane marking detection can be also performed in HSI color space [12] and multi-color features [13], [5]. The usages of filtering methods can help to extract the features of line segments [14]–[16]. Then, we can extract the candidate points to reduce the computation and utilize the pre-defined rules to confirm final lane detection. Hough transform extracting line features can be applied to detect the lane feature and lane boundary [17]–[20]. The above methods based on color, contrast and edge information are easily affected by environments and lighting conditions to mortify their detection performances. To avoid unreliable

The associate editor coordinating the review of this manuscript and approving it for publication was Genny Tortora.

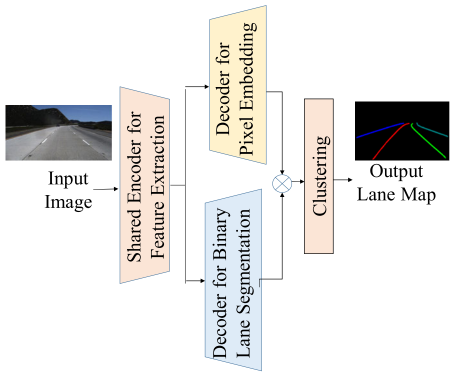
Input Image
Shared Encoder for Feature Extraction
Decoder for Pixel Embedding
Decoder for Binary Lane Segmentation
Clustering
Output Lane Map

FIGURE 1. The simplified structure of LaneNet [27].

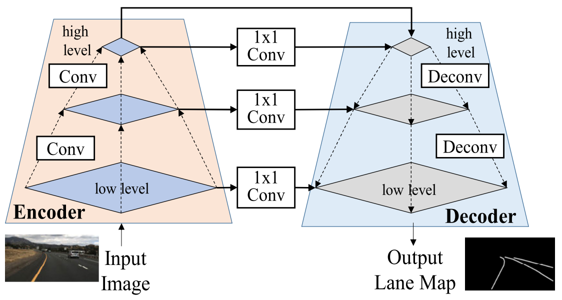

FIGURE 2. Structure of a multi-level feature network for lane detection.

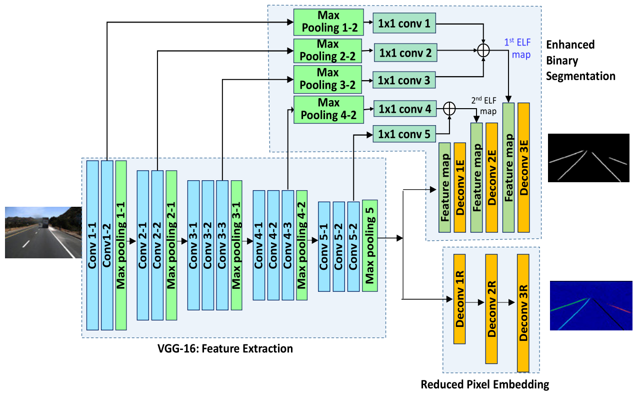

FIGURE 3. The complete network architecture of the proposed lane detection network including one encoder and two decoders.

hand-crafted features, the deep learning approach, which has been recognized as an effective detection tool, can extract the features automatically and deduce the precise lane detection successfully.

With designed loss functions, the deep learning approach depends on the model learned from large training dataset, which could achieve robust detection results under complex conditions. Pan *et al.* [21] proposed a spatial convolutional neural network (SCNN), which is particularly suitable for long shape structure to achieve a good lane detection. Li *et al.* [22] proposed a multitask CNN, which can detect the presence and the geometric attributes of the targets in region of interest with a recurrent neural network. They can detect lane boundaries, including those areas containing no marks. Huval *et al.* [23] extended the vehicle detection CNN for lane boundary detection, where tiny bounding boxes are needed to further connect to become lanes with a regression post-processing method.

The lane detection can be treated as a pixel-wise segmentation problem [24]–[29]. In this case, all the pixels can be detected either lane or background pixels. For pixel-wise segmentation, SegNet [24] utilizes VGG-16 [25], as the encoder, with convolution layer and max pooling to get the final features while the VGG-16 features are reconstructed by the decoder with deconvolutional layers and upsampling. For real-time applications, ENet [26] is further designed to reduce the computation of SegNet for semantic segmentation. For end-to-end lane detection, Neven *et al.* proposed LaneNet [27], as shown in Fig. 1, first performs the feature extraction by a shared encoder and uses two decoders: one for binary lane segmentation and the other for pixel embedding [30]. The final lane map can be obtained after fusion of two feature branches and clustering fused results. Generally, the encoder could be any CNN model, which is originally designed for object classification. The segmented lane detection model by using a pair of encoder and decoder performs the binary classification of the pixels with i.e., 0 for into background or 1 for lane pixels. For practical usages, LaneNet without any pre- and post-processing can identify all lanes separately for better vehicle guidance in very fast computation speed. The low resolution and inaccurate positions are the problems of the LaneNet. Keeping low computational complexity, we need to improve lane detection performance for real applications.

II. THE PROPOSED LANE DETECTION NETWORK

To achieve better accuracy performance in low computation, we propose a lane detection network based on the structure of LaneNet as shown in Fig. 1. In following subsections, we will describe the proposed network by using multiple level features based on VGG-16 backbone encoder. The designs of the decoders for the enhanced binary segmentation (EBS) and the reduced pixel embedding (RPE) branches with their associated loss functions are addressed in details.

A. PROPOSED NETWORK ARCHITECTURE

The original LaneNet uses ENet network as the backbone, which is too simple to extract decent features. To achieve high resolution lane detection, we suggest to use a more powerful multi-level backbone network and further include lower level features mixed in the decoder while LaneNet only adopts high level features. Fig. 2 shows a general structure of a multi-level feature network for lane detection.

By adopting multi-level features, Fig. 3 shows the detailed structure of the proposed lane detection network. We adopt 16-layer VGG-16 [25] as the kernel encoder to perform multi-feature extraction. However, the powerful encoder also increases system loading. To reduce the overhead, we utilize only 3-layer decoders with adding features from the encoder to perform binary lane segmentation and pixel

TABLE 1. The parameters of the proposed network architecture.

Module	Layer	Output Size	Kernel	Stride
VGG-16	Conv 1_1	224×224×64	3×3	1
VGG-16	Conv 1_2	224×224×64	3×3	1
VGG-16	Pooling 1-1	112×112×64	2×2	2
EBS	Pooling 1-2	28×28×64	8×8	8
EBS	1×1 Conv 1	28×28×2	1×1	1
VGG-16	Conv 2-1	112×112×128	3×3	1
VGG-16	Conv 2-2	112×112×128	3×3	1
VGG-16	Pooling 2-1	56×56×128	2×2	2
EBS	Pooling 2-2	28×28×128	4×4	4
EBS	1×1 Conv 2	28×28×2	1×1	1
VGG-16	Conv 3-1	56×56×256	3×3	1
VGG-16	Conv 3-2	56×56×256	3×3	1
VGG-16	Conv 3-3	56×56×256	3×3	1
VGG-16	Pooling 3-1	28×28×256	2×2	2
EBS	Pooling 3-2	28×28×256	2×2	2
EBS	1×1 Conv 3	28×28×2	1×1	1
VGG-16	Conv 4-1	28×28×512	3×3	1
VGG-16	Conv 4-2	28×28×512	3×3	1
VGG-16	Conv 4-3	28×28×512	3×3	1
VGG-16	Pooling 4-1	14×14×512	2×2	2
EBS	Pooling 4-2	14×14×512	2×2	2
EBS	1×1 Conv 4	14×14×2	1×1	1
VGG-16	Conv 5-1	14×14×512	3×3	1
VGG-16	Conv 5-2	14×14×512	3×3	1
VGG-16	Conv 5-3	14×14×512	3×3	1
VGG-16	Pooling 5	7×7×512	2×2	2
EBS	1×1 Conv 5	14×14×2	1×1	1
EBS	Deconv 1E	14×14×2	4×4	2
EBS	Deconv 2E	28×28×2	4×4	2
EBS	Deconv 3E	224×224×2	16×16	8
RPE	Deconv 1R	14×14×6	4×4	2
RPE	Deconv 2R	28×28×6	4×4	2
RPE	Deconv 3R	224×224×6	16×16	8

EBS=Enhanced Binary Segmentation RPE=Reduced Pixel Embedding

embedding branches. With indexed functional units in Fig. 3, the detailed parameters in the multi-feature VGG-16 encoder, the enhanced binary segmentation (EBS) and the reduced pixel embedding (RPE) modules are enlisted in Table 1. The convolutional layer includes ReLU and batch normalization. The EBS and RPE modules non-symmetrically comprise 3 deconvolutional (deconv) layers for computation reduction. Similar to LaneNet, we fuse EBS and RPE results and perform the feature clustering to achieve the final lane detection. The details of the EBS and RPE modules and the designs of loss functions are addressed as follows.

B. ENHANCED BINARY SEGMENTATION

In the enhanced binary segmentation (EBS) branch, as shown in Fig. 3, the decoder includes 5 level feature maps from the backbone encoder to improve the detection details for binary segmentation. After max pooling and 1×1 convolutional layer, we resize and add first 3 level feature maps as the 1st enhanced low feature (1st ELF) map and add last 2 level feature maps as the 2nd enhanced low feature (2nd ELF) map to enhance the binary segmentation. The max pooling with 1×1 convolutional layer adjusts the dimensions of the feature maps to become the same size such that the maps can be directly added instead of concatenation to reduce the computation. The 1×1 convolutional layer is also acted as a selection filter, which can keep the important features and discard the insignificant ones. Thus, the proposed EBS module can achieve precise binary lane segmentation and avoid computation burden.

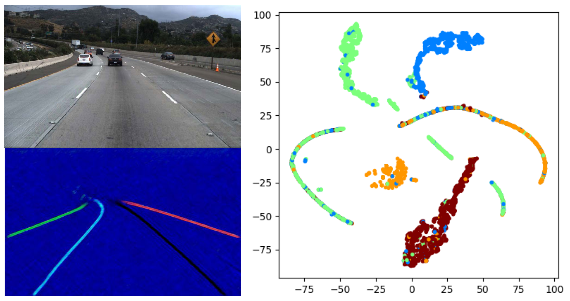

FIGURE 4. Reduced pixel embedding results and their t-SNE visualizations.

C. REDUCED PIXEL EMBEDDING

The reduced pixel embedding (RPE) branch, as shown in Fig. 3, uses the discriminative concept to classify the pixels of the whole image [27]. Fig. 4 shows the discriminated results in t-SNE visualization of multi-dimension feature map [31], which can help to observe the discriminated groups in the feature space. Furthermore, a discriminative loss function is used to evaluate the convergence of lane pixels. The loss function adjusts the location of the feature points to minimize the difference within the same group and maximize the difference between distinct groups, where each group actually corresponds to a lane.

Without multiple level features, the RPE branch can be also treated as a kind of segmentation task, but uses a slightly different loss function. Therefore, the branch architecture is similar to a semantic segmentation model. The RPR branch uses 3 deconv layers to perform pixel embedding. The feature extraction from the backbone network is used for both EBS and RPE modules, however the RPE only uses high level feature. The 3-layer EBS and RPE modules in the proposed network effectively achieve the precise lane detection and save the computation.

D. CLUSTERING

After the computation of EBS and RPE outputs, we need to merge them to get the final output. In this paper, the mean shift algorithm [32] is used to cluster the RPE results. First, we utilize the EBS result as a binary mask to take the RPE regions for clustering process. Secondly, the masked RPE results are used as the elements for clustering the binary lane region into the detected lanes. It is noted that the k-mean clustering method needs to know the number of lane lines in advance while the mean shift method does not need before the clustering process.

E. LOSS FUNCTIONS

This paper proposes a two-branch network architecture to perform lane detection. Each branch has its own loss function to be trained for different targets. The EBS branch uses the

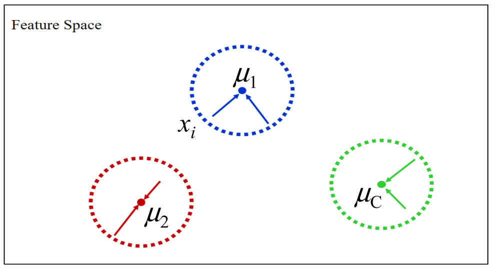

FIGURE 5. Schematic diagram of variance loss to pull the pixels to the feature centers in feature space.

cross entropy loss function as

$$L_{EBS} = -y\log(\hat{y}) - (1-y)\log(1-\hat{y}), \tag{1}$$

where y denotes the truth label and $\hat{y}$ represents predicted probability, which is obtained from the softmax as

$$\hat{y}_k = \frac{\exp(f(x)_k)}{\sum_{k=1}^{K}\exp(f(x)_k)}, \tag{2}$$

where x and $f(x)$ denote the input feature vector and the output function of the EBS branch, respectively. The cross entropy loss consults the difference between the output pixel label distribution and the actual pixel label distribution.

The RPE branch uses the loss function proposed in [28]. The loss function comprises three items, namely the variance, distance, and the regularization losses. In feature space, the variance loss is treated as the pull force showed as,

$$L_{var} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_C}\sum_{i=1}^{N_C}\max([\|\mu_c - x_i\| - \delta_v]^2, 0), \tag{3}$$

where C denotes the number of clusters in ground truth, N_c is the number of elements in cluster C, μ_c is the mean embedding position of the cluster C, x_i is embedding element, δ_v is the threshold for variance loss and max denotes the maximum operation. As shown in Fig. 5, variance loss makes the pixel points which belong to the same lane close to the group center. Hence, if the distance between the pixel point and its group center is too large, it increases the variance loss.

The distance item is treated as the push force given by

$$L_{dist} = \frac{1}{C(C-1)}\sum_{c_A=1}^{C}\sum_{c_B=1}^{C}\max([2\delta_d - \|\mu_{c_A} - \mu_{c_B}\|]^2, 0), \quad c_A \neq c_B, \tag{4}$$

where δ_d denotes the threshold for distance loss. Distance loss pushes different group centers away to keep a certain distance between them. In order to avoid the lanes being confused with each other, if a group center is too close to others, it increases the loss as shown in Fig. 6.

Finally, the regularization loss, which serves to pull all the clusters toward the origin of the feature space is stated as

$$L_{reg} = \frac{1}{C}\sum_{c=1}^{C}\|\mu_c\|. \tag{5}$$

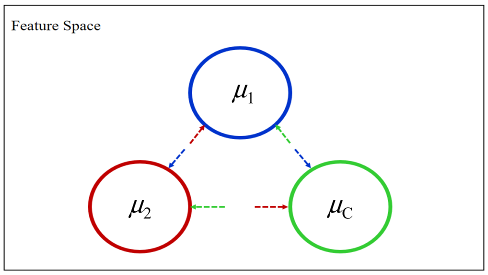

FIGURE 6. Schematic diagram of distance loss to push larger distances between any two lanes.

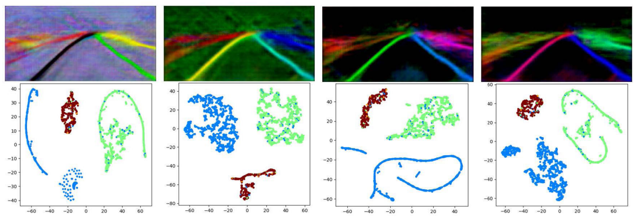

FIGURE 7. Detected results and their t-SNE visualization plots with $\gamma = 0.001$, $\alpha = 1$ and $\beta = 1, 2, 3$ and 4 (from left to right).

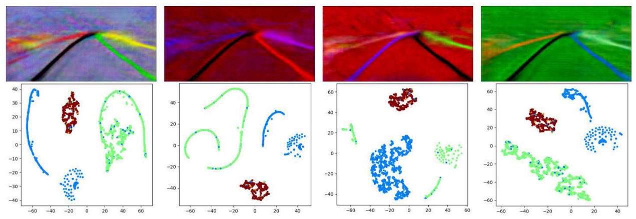

FIGURE 8. Detected results and their t-SNE visualization plots with $\gamma = 0.001$, $\beta = 1$ and $\alpha = 1, 2, 3$ and 4 (from left to right).

The variance and distance loss functions are hinged. In other words, the variance term only works when the distance between the embedding element and the group center is bigger than δ_v. This makes sure all the embedding elements which belong to the same group congregate together. On the other hand, the distance term is only activated when two cluster centers are closer than $2\delta_d$. This can maintain a distance between them. The regularization term limits the activation boundary, preventing the clusters from being too far from the origin. Finally, the total discriminative loss for the RPE module, L_{RPE} combines the variance, distance and regularization terms as

$$L_{RPE} = \alpha \cdot L_{var} + \beta \cdot L_{dist} + \gamma \cdot L_{reg}, \tag{6}$$

where α, β and γ are three selected weighting coefficients. Since the major contributions of L_{RPE} are the pull force by variance loss and the push force by distance loss, we need to determine α and β first. By setting $\gamma = 0.001$, the detection results with t-SNE visualizations of features with different α and β are shown in Fig. 7 and Fig. 8, respectively. For better grouping of features, we finally set $\gamma = 0.001$, $\alpha = 1$ and $\beta = 1$ to achieve the best performances by simulations.

III. SIMULATION RESULTS

In order to evaluate the proposed method, we conduct simulations in this section. The descriptions of dataset, evaluation criterion, data augmentation, experimental results and discussions are detailed as the following subsections.

A. DATASET

The performances of the proposed scheme were evaluated by using two well-known training and testing datasets, Tusimple [33] and CuLane [34], which contain images recorded in front of the car. In particular, Tusimple contains the images acquired in USA, while CuLane comprises the images captured in China.

Tusimple dataset is collected under nice weather conditions and clean highway at day times. Every image contains 2-5 lanes. The dataset comprises 3,626 images for training, and 2,782 images for testing. This dataset provides an algorithm to transform the point labels to segmentation labels, where the different lanes have different label values. In our experiments, because this dataset does not contain validation data, we set 20% of the training data for validation while the remaining 80% is used for training, then use cross validation technique to train all the training data after five iterations. Tusimple dataset contains an annotation file of ground truth.

CuLane dataset is captured by six vehicles in Beijing with various traffic conditions of highways, regular roads, side streets in day and night. The dataset contains 133,235 frames, where 88,880 frames are used for training, 9,675 for validation and 34,680 for testing. CuLane dataset also contains an annotation file of ground truth. The details of configurations of datasets are shown in TABLE 2.

TABLE 2. The adjusted configurations of datasets.

Dataset / Data Type	Tusimple	CuLane
Training	2901	88880
Validation	725	9675
Testing	2782	34680

B. EVALUATION CRITERIA

The performance of the proposed and the existed methods, are evaluated in terms of F1 metric. F1-score is a common criterion for evaluating the performance of a classification algorithm defined as

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}, \tag{7}$$

where TP, FN, and FP denote true positive, false negative and false positive rates, respectively. Of course, we can evaluate it by recall and precision rates respectively given as

$$Recall\ rate = TP/(TP + FN), \tag{8}$$

$$Precision\ rate = TP/(TP + FP), \tag{9}$$

In CuLane dataset, F1-score is calculated in lane-wise fashion, which computes the intersection over union (IoU) with lane prediction and ground truth. If the IoU is higher than a threshold, the lane prediction belongs to true positive. As shown in Fig. 9, green line denotes the ground truth of the image, while blue and red lines indicate true positive and false positive, respectively.

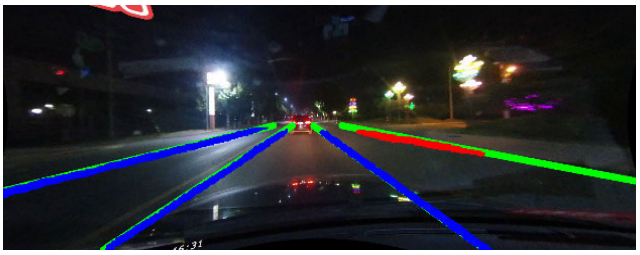

FIGURE 9. The IoU criterion for performance assessment: Green line denotes ground truth, blue line is TP and red line is FP.

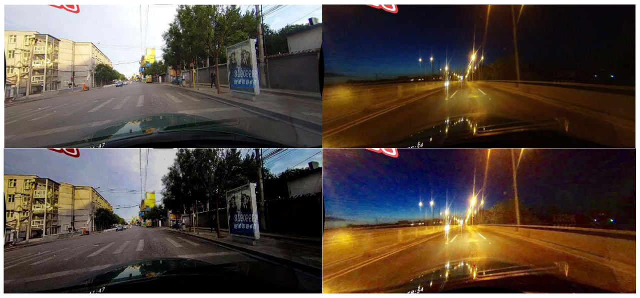

FIGURE 10. Data augmentation. Upper are original images, bottom are images with random brightness.

In Tusimple dataset, they also suggest a different evaluation criterion by sampling the detected lanes with lane points in fixed y-coordinate and comparing the distance between ground truth and prediction x-coordinate of the lane point. If the distance is lower than the threshold, the test point is treated as correct. Finally, it calculates the accuracy rate of the correct points over total points as:

$$Accuracy\ rate = \frac{number\ of\ correct\ points}{number\ of\ total\ points}. \tag{10}$$

C. DATA AUGMENTATION

Data augmentation is used for making the model robust in different lightness situations. In training step, every input image is added a random brightness. It makes the model adapt the same image in different illumination. Especially in the evening, the model can minimize the influence of the low luminescence, and maintain good recognition quality. Fig.10 shows the data augmentation with random brightness

D. TRAINING SET UP AND DETAILS

The loss function parameters are assigned the setting given in [1]. It combines the pixel embedding loss and the binary segmentation loss to give a total loss for training process. In addition, the training process is performed in accordance with the following loss function,

$$L_{Total} = L_{EBS} + L_{RPE}. \tag{11}$$

The training process resizes the images to 512x256, and sets 4 images for a batch as an input, and tests for validation data very 2,000 iterations. In addition, the training model is optimized using Adam with a 5e-4 initial learning rate.

TABLE 3. Running speeds of all lane detection networks.

Method	Speed (fps)
VGG-16+SCNN [21]	8.4
MRFNet$^+$ [38]	13
ResNet-50$^+$ [39]	28.6
ResNet-101$^+$ [39]	18.8
LaneNet [27]	62.5
FCN-32s$^+$ [36]	45.3
DenseNet$^+$ [37]	43.6
FCN-8s$^+$ [36]	43.1
Ours	**42.8**

E. EXPERIMENTS

In this subsection, we compare the proposed model to other methods, including two-branch models such as LaneNet, FCN-32s [36], FCN-8s [36] as well as the famous backbones such as DenseNet [37], MRFNet [38], ResNet-50 [39], and ResNet-101 [39]. For pixel-wise segmentation networks, we can duplicate another decoder for pixel imbedding. For backbone networks, we can introduce two decoders, which are symmetrically constructed by reversely changing the convolution and pooling units to deconvolution and upsampling ones, respectively. For fair comparisons, we trained the networks using the same loss functions as the proposed network with the same coefficients as $\alpha = 1$, $\beta = 1$, $\gamma = 0.001$. $\delta_v = 0.5$ and $\delta_d = 3$. Table 3 shows the computation speeds of all the lane detection networks, where the superscript, "+" denotes the networks, which originally are not re-designed for lane detection segmentation and will be evaluated for the comparisons. All the methods were tested in Tensorflow with GPU GTX 1080Ti. LaneNet with 62.6 fps has the fastest computation while VGG-16+SCNN [21] with 8.4 fps is the slowest one. The proposed network achieves the reasonable speed, which is good enough for real-time applications. Generally, the more complex backbone CNNs have the lower computation speeds.

1) TUSIMPLE DATASET

In TuSimple dataset, we first test the networks, which satisfy the real-time requirements, i.e., their speeds are larger than 30 fps. Marked by superscript, "$*$", we also show the results of the top five methods enlisted in TuSimple Benchmark Lane Detection Challenge for references. As shown in Table 4, VGG-16+SCNN [21], which won the 1st place on the Challenge. It performs better than the proposed method with about 2.7% accuracy performance. However, its speed cannot detect the results in real time. Hsu *et. al.* [35] uses a similar concept and retains the low-level feature to restore details. They adopted symmetric decoder and encoder structure with task-specific layers to integrate all information and need more computation. Tables 4 and 5 show that the proposed method performs best for accurate and F1 performances among all two-branch methods on Tusimple dataset.

As shown as Fig. 11, the proposed network performs best. FCN-8s has excellent result in general, however, the pixels which are close to vanishing point and image boundary are not accurate enough. Actually, FCN-8s [36], which also retains the feature maps from the last three maxpooling layer and adds to three deconv layers, is very close to the proposed model. FCN-8s lacks the information from the first two level feature maps such that it cannot achieve fine details by convolution computing. DenseNet [37] uses more features that make the performance higher than FCN-8s.

TABLE 4. Accurate performance of the methods on TuSimple.

Method	Accurate Rate
VGG-16+SCNN* [21]	96.53
Y. Hsu* [35]	96.50
D. Neven* [27] (Lanenet+H-net)	96.40
Xxxxcvxxxx*	96.14
TF Placeholder	95.96
Ours	**93.8**
DenseNet$^+$ [37]	93.5
FCN-8s$^+$ [36]	91.2
LaneNet [27]	90.8
FCN-32s$^+$ [36]	90.1

TABLE 5. F1-score performance of two-branch methods on TuSimple.

Method	F1-score (pixel-wise)
Ours	**67.5**
DenseNet$^+$ [37]	65.8
FCN-8s$^+$ [36]	60.5
LaneNet [27]	49.8
FCN-32s$^+$ [36]	47.4

TABLE 6. F1-score performance on CuLane dataset.

Method	F1-score (pixel-wise)
VGG-16+SCNN [21]	71.6
MRFNet$^+$ [38]	66.7
ResNet-50$^+$ [39]	66.7
ResNet-101$^+$ [39]	70.8
Ours	**66.5**

2) CULANE DATASET

Since the propose network is better than FCN-8s [36], FCN-32s [36], LaneNet [27], and DenseNet [37] on Tusimple dataset, we ignored them for comparisons. We only evaluate the other complex networks, such as VGG-16+SCNN [21], MRFNet [38], ResNet-50 [39], and ResNet-101 [39] on CuLane database. As shown in Table 6, VGG-16+SCNN still obtains the best F1-score in lane-wise detection. The SCNN combines all feature maps obtained from different convolutional directions to collect most information from the image for prediction. Comparing to ResNet-50, the proposed method has lower 0.2% in F1-score, but achieves faster speed more than 14.2 frames per second (fps), which is exhibited in Table 5.

As shown in Fig. 12, the detected results on CuLane datast also show similar conclusions as Tusimple dataset. However, the proposed network performs well in the regions, which are close to vanishing point and image boundary. Furthermore, some occulated area, where the lanes are covered by scooters, vehicles, or sunlight reflection, can influence the results of most lane detection networks. For the occluded lane restoration, the proposed method is better than FCN-8s. In summary, the proposed method utilizes more low-level

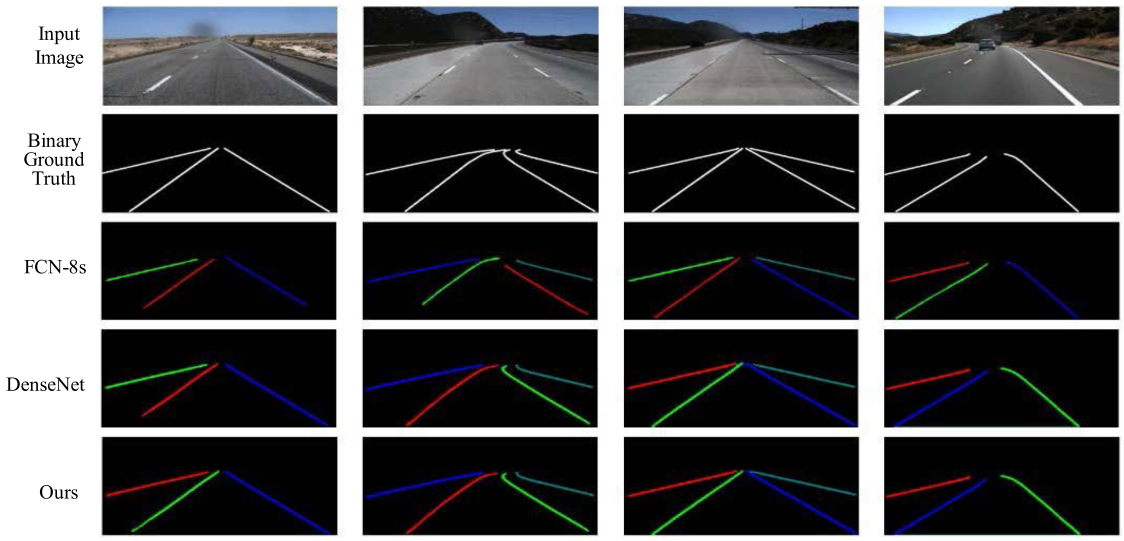

FIGURE 11. Detected results on Tusimple dataset with other two-branch segmentation models.

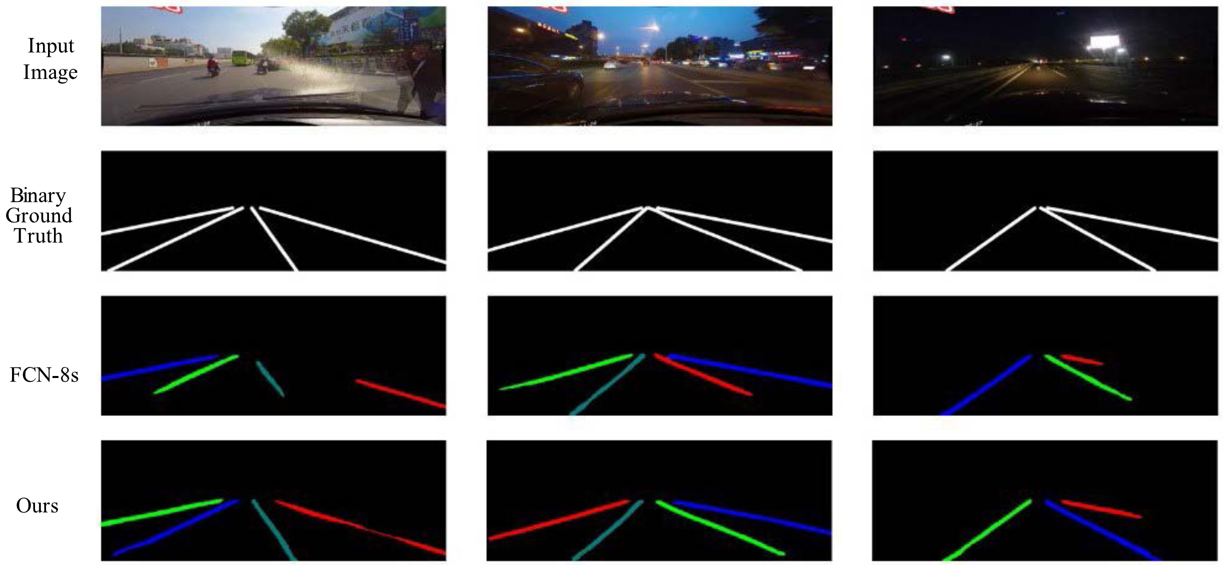

FIGURE 12. Detected results on CuLane dataset with other two-branch segmentation models.

feature maps to keep more accurate spatial information. The proposed network, which can help for restoring the occluded lanes, is robust to actual road conditions to achieve complete lane segmentation for real-time applications.

3) EVALUATION OF ENHANCED LEVEL FEATURE MAPS

In order to evaluate the effectiveness of multiple level features suggested in the proposed method, we tried to remove the first enhanced low feature (1^{st} ELF) map in Fig. 3 and tested it by simulations. As shown in Fig. 13, the proposed model without the 1^{st} ELF map lost some entire lanes in the testing images. Most of missed lanes are caused by the condition of light intensity or occlusion by other objects. It is noted that missing a whole lane is a serious problem for lane detection. It will greatly affect the safety problem for the driver. The low level features not only help to restore the details but also help to detect the occluded lanes. Furthermore, the additions of the 1st and 2^{nd} ELF maps do not increase the calculation too much for including low-level features. The proposal network with the speed over 42 fps acquires a good balance between processing speed and detection accuracy.

4) EVALUATION OF DATA AUGMENTATION

In training processing, we also tested different proportion of data augmentation is set. Fig.14 shows two images obtained from CuLane dataset. The left images are clean street in daytime while the right ones are regular road at night. With data augmentation (lower images), we can correctly detect the missing lane in the night image. The results show that the lighting data augmentation can improve the cases which are in night or poor lighting conditions. To test different ratios of data augmentation from 0% to 100% with random

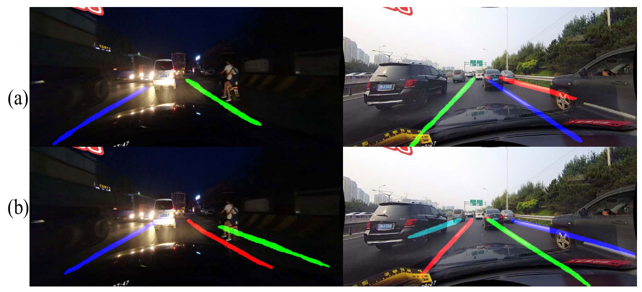

FIGURE 13. The detected results achieved by the proposed method: (a) without low-level feature (upper); (b) with low-level feature adding (lower).

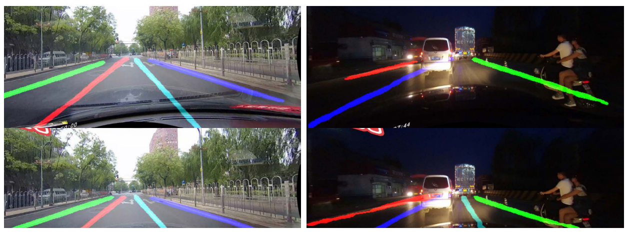

FIGURE 14. Data augmentation: original day (left) and night (right) images at top and their augmented images with 100% random lighting variations at bottom.

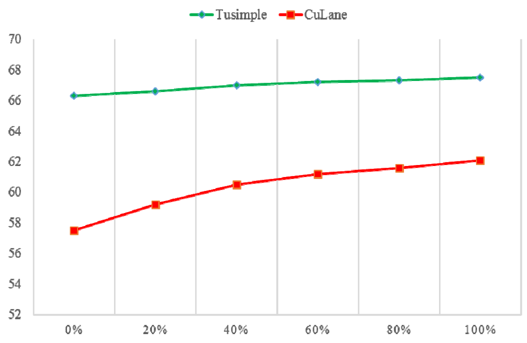

FIGURE 15. Performance curves with different data augmentation proportion for Tusimple and CuLane datasets.

brightness, the augmentation makes the model become robust to recognize lanes in night scene or poor lighting condition. Fig. 15 shows the performance improvements of Tusimple and CuLane datasets are 1.2% and 4.6%, respectively. The main reason is that Tusimple data are collected in good weather and clean highway while CuLane data are captured in complex traffic and night conditions. Therefore, the improvement of data augmentation in CuLane is more than Tusimple.

IV. CONCLUSION

This paper presents a real-time lane detection model based on LaneNet with one encoder and two decoders. By using the multiple level feature concept, the proposed lane detection network uses 16-layer VGG-16 as the encoder for feature extraction and adopts 3-layer deconv decoders to perform binary segmentation and pixel embedding branches. The proposed network achieves better performances than LaneNet with a reasonable increase of computation. The 5 low level features are added into the enhanced binary segmentation (EBS) branch, in which we use max pooling layers and 1×1 convolutional units to adjust the sizes of feature maps, which can directly added to 2 deconv layers. Simulations show that the multiple level features can help to increase the accuracy performances of lane detection and reach the real-time requirements. Compared to other well-known methods, the proposed network under real-time requirement shows the superiority in maintaining both speed and accuracy. The proposed lane detection network has better ability of restoring details and recognizing the occluded lanes due to the usages of multiple level features.

REFERENCES

[1] T.-Y. Sun, S.-J. Tsai, and V. Chan, "HSI color model based lane-marking detection," in *Proc. Intell. Transp. Syst. Conf.*, Sep. 2016, pp. 1168–1172.

[2] H.-Y. Cheng, B.-S. Jeng, P.-T. Tseng, and K.-C. Fan, "Lane detection with moving vehicles in the traffic scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 571–582, Dec. 2006.

[3] H. Jung, J. Min, and J. Kim, "An efficient lane detection algorithm for lane departure detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Gold Coast, QLD, Australia, Jun. 2013, pp. 976–981.

[4] Z. Teng, J.-H. Kim, and D.-J. Kang, "Real-time lane detection by using multiple cues," in *Proc. Int. Conf. Control Autom. Syst*, Oct. 2010, pp. 2334–2337.

[5] L. N. P. Boggavarapu, R. S. Vaddi, H. D. Vankayalapati, and J. K. Munagala, "A robust multi-color lane marking detection approach for Indian scenario," *Int. J. Adv. Comput*, vol. 2, no. 5, pp. 71–75, 2011.

[6] K.-Y. Chin and S.-F. Lin, "Lane detection using color-based segmentation," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2015, pp. 706–711.

[7] D. Ding, C. Lee, and K.-Y. Lee, "An adaptive road ROI determination algorithm for lane detection," in *Proc. TENCON*, Oct. 2013, pp. 1–4.

[8] P. Chanawangsa and C. W. Chen, "A new color-based lane detection via Gaussian radial basis function networks," in *Proc. Int. Conf. Connected Vehicles Expo (ICCVE)*, Dec. 2012, pp. 166–171.

[9] Y. He, H. Wang, and B. Zhang, "Color-based road detection in urban traffic scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 309–318, Dec. 2003.

[10] M. A. Sotelo, F. J. Rodriguez, L. Magdalena, L. M. Bergasa, and L. Boquete, "A color vision-based lane tracking system for autonomous driving on unmarked roads," *Auto. Robots*, vol. 16, no. 1, pp. 95–116, Jan. 2004.

[11] C. Lee and J.-H. Moon, "Robust lane detection and tracking for real-time applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 4043–4048, Dec. 2018.

[12] T. T. Tran, C. S. Bae, Y. N. Kim, H. M. Cho, and S. B. Cho, "An adaptive method for lane marking detection based on HSI color model," in *Advanced Intelligent Computing Theories and Applications*. Berlin, Germany: Springer-Verlag, 2010, pp. 304–311.

[13] G. Kaur and D. Kumar, "Lane detection techniques: A review," *Int. J. Comput. Appl.*, vol. 112, no. 10, pp. 4–8, Feb. 2015.

[14] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 9, pp. 891–906, Sep. 1991.

[15] T. Gao and H. Aghajan, "Self lane assignment using egocentric smart mobile camera for intelligent gps navigation," in *Proc. IEEE Comput. Vis. Pattern Recognit. Workshops*, Jun. 2009, pp. 57–62.

[16] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010.

[17] R. Duda and P. Hart, "Use of Hough transform to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[18] C. Tu, B. J. van Wyk, Y. Hamam, K. Djouan, and S. Du, "Vehicle position monitoring using Hough transform," *IERI Procedia*, vol. 4, pp. 316–322, Jan. 2013.

[19] D. Hanwell and M. Mirmehdi, "Detection of lane departure on high-speed roads," in *Proc. ICPRAM*, 2012, pp. 529–536.

[20] S. Srivastava, M. Lumb, and R. Singal, "Improved lane detection using hybrid median filter and modified Hough transform," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 4, no. 1, pp. 30–37, Jan. 2014.

[21] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial CNN for traffic scene understanding," Dec. 2017, *arXiv:1712.06080*. [Online]. Available: https://arxiv.org/abs/1712.06080

[22] J. Li, X. Mei, D. V. Prokhorov, and D. Tao, "Deep neural network for structural prediction and lane detection in traffic scene," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 690–730, Mar. 2017.

[23] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, F. Mujica, A. Coates, and A. Y. Ng, "An empirical evaluation of deep learning on highway driving," Apr. 2015, *arXiv:1504.01716*. [Online]. Available: https://arxiv.org/abs/1504.01716

[24] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," Nov. 2015, *arXiv:1511.00561*. [Online]. Available: https://arxiv.org/abs/1511.00561

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[26] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," Jun. 2016, *arXiv:1606.02147*. [Online]. Available: https://arxiv.org/abs/1606.02147

[27] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool, "Towards end-to-end lane detection: An instance segmentation approach," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2018, pp. 286–291.

[28] W. Wang, J. Shen, R. Yang, and F. Porikli, "Saliency-aware video object segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 20–33, Jan. 2018.

[29] W. Wang, J. Shen, and L. Shao, "Consistent video saliency using local gradient flow optimization and global refinement," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4185–4196, Nov. 2015.

[30] B. De Brabandere, D. Neven, and L. Van Gool, "Semantic instance segmentation with a discriminative loss function," Aug. 2017, *arXiv:1708.02551v1*. [Online]. Available: https://arxiv.org/abs/1708.02551v1

[31] L. van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[32] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.

[33] *TuSimple Lane Challange*. Accessed: 2017. [Online]. Available: http://benchmark.tusimple.ai/

[34] *CuLane Dataset*. Accessed: Feb. 2018. [Online]. Available: https://xingangpan.github.io/projects/CULane.html

[35] Y.-C. Hsu, Z. Xu, Z. Kira, and J. Huang, "Learning to cluster for proposal-free instance segmentation," in *Proc. Joint Int. Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[36] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[37] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[38] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang, "Deep learning Markov random field for semantic segmentation," Jun. 2016, *arXiv:1606.07230*. [Online]. Available: https://arxiv.org/abs/1606.07230

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 770–778.

WEI-JONG YANG was born in Hsinchu, Taiwan, in 1990. He received the B.S. degree in computer science from Tunghai University, Taiwan, in 2012, and the M.S. degree in computer science and information engineering from the National University of Tainan, Taiwan, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, Taiwan. His current research interests include video coding and processing, pattern recognition, machine learning, and deep learning systems.

YOA-TENG CHENG was born in Chiayi, Taiwan, in 1995. He received the B.S. degree in electrical engineering from National Chi Nan University, Taiwan, in 2017, and the M.S. degree from the Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, Taiwan. His current research interests include video detection, pattern recognition, and deep learning systems.

PAU-CHOO CHUNG (S'89–M'91–SM'02–F'08) received the Ph.D. degree in electrical engineering from Texas Tech University, Lubbock, TX, USA, in 1991. She was with the Department of Electrical Engineering, National Cheng Kung University (NCKU), Tainan, Taiwan, in 1991, and became a Full Professor, in 1996. She applies most of her research results to healthcare and medical applications. Her research interests include image/video analysis and pattern recognition, and biosignal analysis. She was a member of the Board of Governors of CAS Society, from 2007 to 2009 and from 2010 to 2012. She is a member of the Phi Tau Phi Honor Society. She is currently an ADCOM Member of the IEEE CIS and the Chair of CIS Distinguished Lecturer Program. She is also an Associate Editor of the IEEE TRANSACTION ON NEURAL NETWORKS, an Editor of *Journal of Information Science and Engineering*, a Guest Editor of *Journal of High Speed Network* and the IEEE TRANSACTION ON CIRCUITS AND SYSTEMS-IN PART, and the Secretary General of Biomedical Engineering Society of China. She is one of the Co-Founders of Medical Image Standard Association (MISA), Taiwan, and is currently on the Board of Directors of MISA.