Graph Attention Layer Evolves Semantic Segmentation for Road Pothole Detection: A Benchmark and Algorithms

Rui Fan, *Member, IEEE*, Hengli Wang, *Graduate Student Member, IEEE*, Yuan Wang, Ming Liu, *Senior Member, IEEE*, and Ioannis Pitas, *Fellow, IEEE*

***Abstract*— Existing road pothole detection approaches can be classified as computer vision-based or machine learning-based. The former approaches typically employ 2D image analysis/understanding or 3D point cloud modeling and segmentation algorithms to detect (*i.e.*, recognize and localize) road potholes from vision sensor data, *e.g.*, RGB images and/or depth/disparity images. The latter approaches generally address road pothole detection using convolutional neural networks (CNNs) in an end-to-end manner. However, road potholes are not necessarily ubiquitous and it is challenging to prepare a large well-annotated dataset for CNN training. In this regard, while computer vision-based methods were the mainstream research trend in the past decade, machine learning-based methods were merely discussed. Recently, we published the first stereo vision-based road pothole detection dataset and a novel disparity transformation algorithm, whereby the damaged and undamaged road areas can be highly distinguished. However, there are no benchmarks currently available for state-of-the-art (SoTA) CNNs trained using either disparity images or transformed disparity images. Therefore, in this paper, we first discuss the SoTA CNNs designed for semantic segmentation and evaluate their performance for road pothole detection with extensive experiments. Additionally, inspired by graph neural network (GNN), we propose a novel CNN layer, referred to as graph attention layer (GAL), which can be easily deployed in any existing CNN to optimize image feature representations for semantic segmentation. Our experiments compare GAL-DeepLabv3+, our best-performing implementation, with nine SoTA CNNs on three modalities of training data: RGB images, disparity images, and transformed disparity images. The experimental results suggest that our proposed GAL-DeepLabv3+ achieves the best overall pothole detection accuracy on all training data modalities. The source code, dataset, and benchmark are publicly available at mias.group/GAL-Pothole-Detection.**

Manuscript received October 27, 2020; revised May 30, 2021 and August 2, 2021; accepted August 24, 2021. Date of publication September 24, 2021; date of current version September 29, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sérgio de Faria. *(Rui Fan, Hengli Wang, and Yuan Wang contributed equally to this work.) (Corresponding author: Rui Fan.)*

Rui Fan is with the Department of Control Science and Engineering, College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China, and also with Shanghai Research Institute for Intelligent Autonomous Systems, Shanghai 201210, China (e-mail: rui.fan@ieee.org).

Hengli Wang and Ming Liu are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, SAR, China (e-mail: hwangdf@connect.ust.hk; eelium@ust.hk).

Yuan Wang is with the Industrial Research and Development Center, SmartMore, Shenzhen 518000, China (e-mail: yuan.wang@smartmore.com).

Ioannis Pitas is with the School of Informatics, University of Thessaloniki, 541 24 Thessaloniki, Greece (e-mail: pitas@csd.auth.gr).

Digital Object Identifier 10.1109/TIP.2021.3112316

***Index Terms*— Road pothole detection, machine learning, convolutional neural network, graph neural network.**

I. Introduction

A POTHOLE is a large structural road failure [1]. Its formation is due to the combined presence of water and traffic [2]. Water permeates the ground and weakens the soil under the road surface while traffic subsequently breaks the affected road surface, resulting in the removal of road surface chunks [3]. Road potholes, besides being an inconvenience, are also a safety hazard because they can severely affect driving comfort, vehicle condition, and traffic safety [3]. Therefore, frequently inspecting and repairing road potholes is a crucial road maintenance task [1]. Currently, road potholes are regularly detected and reported by certified inspectors [4]. This manual visual inspection process is tedious, dangerous, costly, and time-consuming [3]. Moreover, manual road pothole detection results are qualitative and subjective as they depend entirely on individual inspectors' experience [5]. Consequently, there is an ever-increasing need for automated road pothole detection systems, especially ones developed based on state-of-the-art (SoTA) computer vision and machine learning techniques.

In [3], we published the world's first multi-modal road pothole detection dataset, containing RGB images, subpixel disparity images, and transformed disparity images. An example of these three modalities of road vision data is shown in Fig. 1. It can be observed that the damaged road areas are highly distinguishable after disparity transformation, making road pothole detection much easier. However, there lacks a benchmark for road pothole detection based on SoTA semantic segmentation convolutional neural networks (CNNs), trained on spatial vision data, *e.g.*, disparity/depth images, other than road RGB images. Therefore, there is a strong motivation to provide a comprehensive comparison for SoTA CNNs w.r.t. different modalities of road vision data. Additionally, some semantic segmentation approaches [6], [7] combine CNNs with graph models, such as conditional random fields (CRFs), to improve the image segmentation performance. Such CRF-based approaches are nevertheless very computationally intensive, and thus they can only be deployed on the final semantic probability map. Therefore, a graph attention layer that can produce additional weights based on the relational

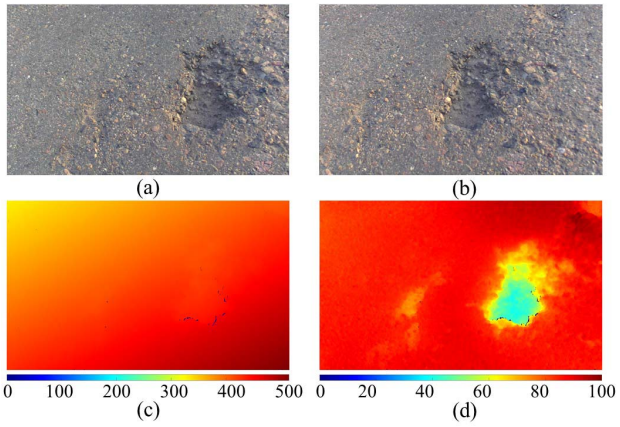


Fig. 1. An example of the multi-modal road vision data provided in [3]: (a) left road image; (b) right road image; (c) dense subpixel disparity image estimated from (a) and (b) using the stereo matching algorithm presented in [8]; (d) transformed disparity image yielded from (c) using the disparity transformation algorithm introduced in [3].

inductive bias to refine image feature representations is also a research topic that requires more attention.

We introduce a novel graph model-based semantic segmentation CNN for road pothole detection, of which the effectiveness is demonstrated with extensive experiments. The contributions of this paper are summarized as follows:

- A benchmark of all SoTA semantic segmentation CNNs trained on the three aforementioned modalities of vision data for road pothole detection;
- Graph attention layer (GAL), a novel graph layer inspired by graph neural network (GNN) [9], which can be easily deployed in any CNN to optimize image feature representations for semantic image segmentation;
- A novel CNN, referred to as GAL-DeepLabv3+, incorporating the proposed GAL in DeepLabv3+ [10]. It outperforms SoTA CNNs for road pothole detection.

The remainder of this paper is structured as follows: Section II reviews the SoTA road pothole detection approaches and semantic segmentation CNNs. Section III introduces our proposed methodology. In Section IV, we compare the performance of our proposed method with the SoTA CNNs reviewed in Section II. Section V discusses the implications and practical application of this work. Finally, Section VI summarizes the paper and provides recommendations for future work.

II. Literature Review

A. Road Pothole Detection

The existing road pothole detection methods can be categorized into two groups: explicit programming-based [3] and machine learning-based [4].

Explicit programming-based methods detect potholes via either two-dimensional (2D) image analysis/understanding or three-dimensional (3D) road point cloud modeling and segmentation [3]. As an example, a Microsoft Kinect sensor is used to capture road depth images [11], which are then segmented using image thresholding method for road pothole detection. In order to ensure the applicability of image thresholding method, the Microsoft Kinect sensor has to be mounted as perpendicularly as possible to the road surface as it requires uniformly distributed background (undamaged road areas) depth values [11]. The 3D road point cloud modeling and segmentation approaches [12] typically interpolate a road surface point cloud into an explicit mathematical model, *e.g.*, a quadratic surface. The road potholes can then be effectively detected by comparing the difference between and actual and the interpolated 3D road surfaces. Recently, [3] introduced a hybrid road pothole detection system developed based on disparity transformation and modeling. The disparity transformation algorithm can not only estimate the stereo camera's roll and pitch angles but also transform the disparity image into a quasi-inverse perspective view, where the background disparity values become very similar [13]. This algorithm does not require the depth sensor's optical axis to be perpendicular to the road surface, greatly enhancing the robustness and adaptability of depth/disparity image segmentation algorithms. Subsequently, a quadratic surface is fitted to the disparities in the undamaged road regions for accurate road pothole detection.

Machine learning-based methods generally train CNNs on well-annotated vision data for end-to-end road pothole detection [14]. Any general-purpose semantic/instance segmentation CNN can be easily applied to detect road potholes from RGB or disparity/depth images. For example, mask region-based CNN (R-CNN) is employed in [15] to detect road potholes from RGB images. In [16], DeepLabv3+ [10] is utilized to segment RGB images for road pothole detection. In [4], five single-modal and three data-fusion CNNs are compared in terms of detecting road potholes from RGB and/or transformed disparity images, where an attention aggregation framework and an adversarial domain adaptation technique are used to boost the CNN performance.

B. Semantic Segmentation Networks

A fully convolutional network (FCN) [17] is an end-to-end, pixel-to-pixel semantic segmentation network. It converts all fully connected (FC) layers to convolutions. An FCN typically consists of a downsampling path and an upsampling path. The downsampling path employs a classification network as the backbone to capture semantic/contextual information, while the upsampling path fully recovers the spatial information using skip connections. FCN-32s, FCN-16s, and FCN-8s are three main variants having different upsampling strides to provide coarse, medium-grain, and fine-grain semantic image segmentation results, respectively. In the paper, FCN-8s is used for road pothole detection.

U-Net [18] was designed based on FCN. It was extended to work with fewer training samples while yielding more accurate segmentation results. U-Net consists of a contracting path and an expansive path. The contracting path has a typical CNN architecture of convolutions, rectified linear units (ReLUs), and max pooling layers. At the same time, the expansive path combines the desired visual features and spatial information through a sequence of upconvolutions and concatenations. The skip connection between the contracting path and the

expansive path helps restore small objects' locations better. Compared with FCN, U-Net has a large number of feature channels in upsampling layers, allowing it to propagate context information to the layers with higher resolution.

SegNet [19] has an encoder-decoder architecture. The encoder network employs VGG-16 [20] to generate high-level feature maps, while the decoder network upsamples its input to produce a sparse feature map, which is then fed to a softmax classifier for pixel-wise classification. Therefore, SegNet has a trainable decoder filter bank, while an FCN does not. The network depth k determines the image downsampling and upsampling by an overall factor of $2^k \times 2^k$.

DeepLabv3+ [10] is developed based on DeepLabv1 [21], DeepLabv2 [22] and DeepLabv3 [23]. It combines the advantages of both the spatial pyramid pooling (SPP) module and the encoder-decoder architecture. Compared to DeepLabv3, it adds a simple yet efficient decoder module to refine the semantic segmentation. Additionally, it employs depth-wise separable convolution to both atrous SPP (ASPP) and decoder modules, making its encoder-decoder structure much faster.

Although ASPP can generate feature maps by concatenating multiple atrous-convolved features, the resolution of these maps is typically not dense enough for applications requiring high accuracy [22]. In this regard, DenseASPP [24] was developed to connect atrous convolutional layers (ACLs) more densely. The ACLs are organized in a cascade fashion, where the dilation rate increases layer by layer. Then, DenseASPP concatenates the output from each atrous layer with the input feature map and the outputs from lower layers. The concatenated feature map is then fed into the following layer. DenseASPP's final output is a feature map generated by multi-scale and multi-rate atrous convolutions. DenseASPP is capable of generating multi-scale features that cover a larger and denser scale range without significantly increasing the model size.

Different from the CNNs mentioned above, the pyramid attention network (PAN) [25] combines an attention mechanism and a spatial pyramid to extract accurate visual features for semantic segmentation. It consists of a feature pyramid attention (FPA) module and a global attention upsample (GAU) module. The FPA module encodes the spatial pyramid attention structure on the high-level output and combines global pooling to learn better feature representations. The GAU module provides global context as a guidance for using low-level visual features to select category localization details.

As discussed above, the recent CNNs with encoder-decoder architectures typically perform bilinear upsampling in the last decoder layer for final pixel-wise region prediction. However, simple bilinear upsampling limits the recovered pixel-wise prediction accuracy because it does not take pixel prediction correlations into account [26]. DUpsampling was therefore introduced to recover the pixel-wise prediction from low-resolution CNN outputs by exploiting the redundancy in the semantic segmentation label space. It allows the decoder to downsample the fused visual features to the lowest feature map resolution before merging them. This approach not only reduces the decoder computation cost but also decouples fused feature resolution from the final prediction.

Although most deep CNNs have achieved compelling semantic segmentation results, large networks are generally slow and computationally intensive. ESPNet [27] was designed to resolve this issue. An efficient spatial pyramid (ESP) module consists of point-wise convolutions to help reduce the computational complexity, and a spatial pyramid of dilated convolutions to resample the feature maps to learn the representations from the large effective receptive field. The ESP module's large effective field introduces gridding artifacts, which are then removed using hierarchical feature fusion. A skip-connection between the input and output is also added to improve the information flow.

Unlike the above-mentioned CNNs, gated shape CNN (GSCNN) [28] leverages a two-branch architecture. The regular branch can be any backbone CNN, while the shape branch processes shape information in parallel to the regular branch through a set of residual blocks and gated convolutional layers. Furthermore, GSCNN uses higher-level activations in the regular branch to gate the lower-level activations in the shape branch, effectively reducing noise and helping the shape branch to focus only on the relevant boundary information. This, in turn, efficiently improves the performance of the regular branch. GSCNN then employs an ASPP to combine the information from the two branches in a multi-scale fashion. The experimental results demonstrate that this architecture can produce sharper predictions around region boundaries and can significantly boost the semantic segmentation accuracy for thinner and smaller objects.

Graph models can produce useful representations for pixel relations, which greatly helps to improve the semantic segmentation performance. As discussed in Section I, the CRF-based approaches are computationally intensive, and can only be deployed on the final semantic probability map. To solve this disadvantage, we propose GAL, which is capable of producing additional weights based on the relational inductive bias to refine image feature representations in a computationally efficient manner.

III. Methodology

The architecture of our introduced semantic segmentation network for road pothole detection is shown in Fig. 2. An initial feature is learned from the input image by the backbone CNN. It is then fed into our proposed GAL to produce a refined feature, which is concatenated with the input feature and sent to the following ASPP module.

A. Graph Attention Layer

A graph is commonly defined as a three-tuple $\mathcal{G}(\mathbf{u}, \mathcal{V}, \mathcal{E})$, where $\mathbf{u}$ is a global attribute, $\mathcal{V} = \{\mathbf{v}_k\}_{k=1:N^v}$ is the vertex set, and $\mathcal{E} = \{(\mathbf{e}_k, r_k, s_k)\}_{k=1:N^e}$ is the edge set (r_k and s_k represents the index of the receiver and sender vertex, respectively) [9].

As illustrated in Fig. 3, our proposed GAL consists of two main components: a feature generator, which generates the representations of both vertex and edge features, and

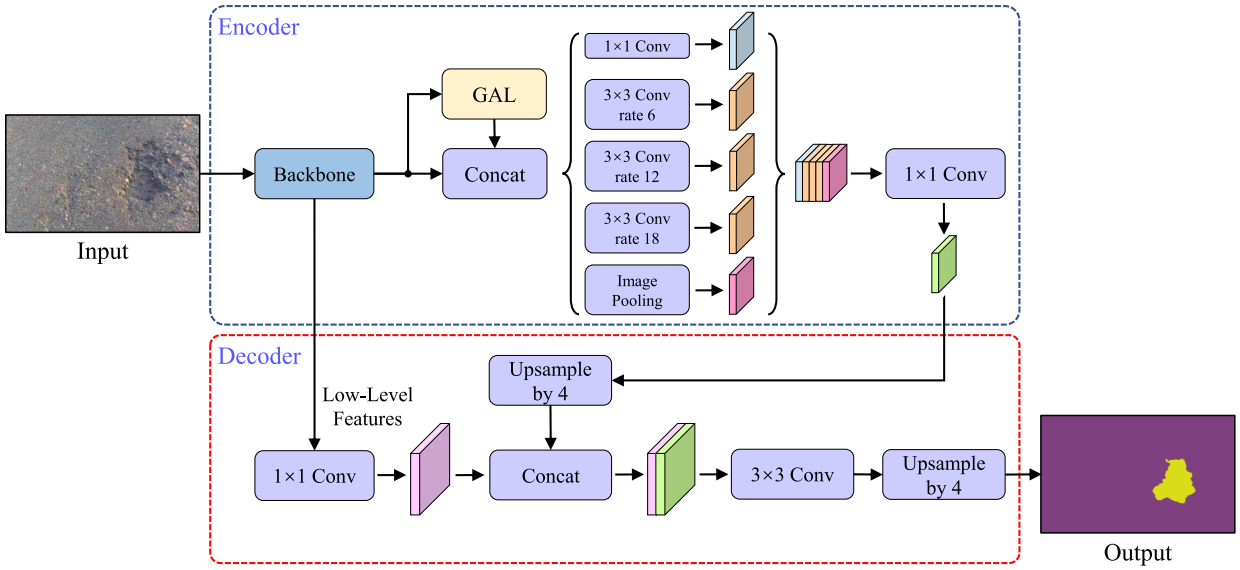


Fig. 2. The overview of our proposed GAL-DeepLabv3+. GAL takes the feature from the backbone as the input and it outputs the refined feature, which is then concatenated with the input feature and sent to the following ASPP module and the decoder.

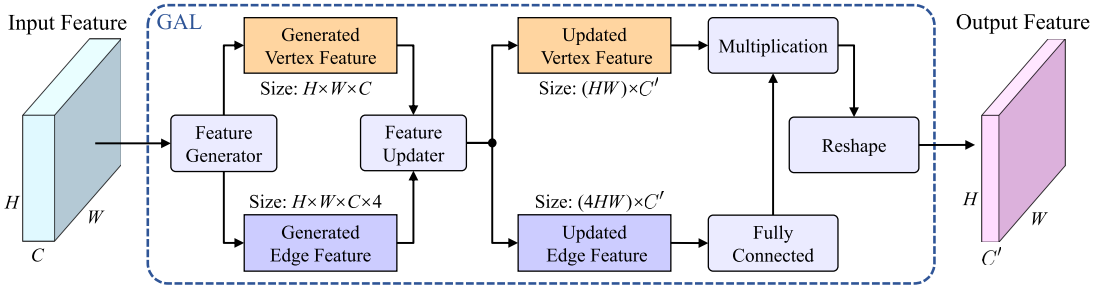


Fig. 3. An illustration of our proposed GAL, which utilizes a feature generator and a feature updater to optimize the input feature of size $H \times W \times C$ and output the refined feature of size $H \times W \times C'$.

a feature updater which updates these two types of feature representations based on our proposed GNN block. Then, we implement our GAL in DeepLabv3+ [10] and refer to it as GAL-DeepLabv3+, as illustrated in Fig. 2. The remaining subsections detail the feature generator and updater of our GAL, as well as our GAL-DeepLabv3+, separately.

1) GAL Feature Generator: As shown in Fig. 3, the input of our GAL is a tensor (feature representation) T of size $H \times W \times C$. T is first converted to a graph $\mathcal{G}(\mathbf{u}, \mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{\mathbf{v}_k\}_{k=1:HWC}$ and $\mathcal{E} = \{(\mathbf{e}_k, r_k, s_k)\}_{k=1:4HWC}$ (only four closest neighbors are considered for each vertex). These are then considered as the vertex and edge features, respectively. An illustration of the edge feature generation is shown in Fig. 4, where it can be seen that there exist two special cases: corners and boundaries. When a given vertex is at the graph corner, the vertexes at another two corners are considered to be its neighbors. Moreover, when a given vertex is on the graph boundary, the vertex itself will be considered as one of its four neighbors. The generated vertex and edge feature will then be updated using a simplified GNN block.

2) GAL Feature Updater: A general full GNN block is illustrated in Fig. 5. It can be seen that it consists of three

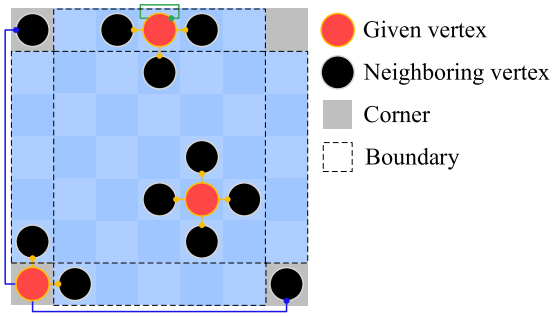


Fig. 4. An illustration of the edge feature generation, where only four closest neighbors are considered for each vertex. Moreover, when the vertex locates at a corner or on a boundary, the corresponding special neighbors are marked with blue and green lines, respectively.

sub-blocks: an edge block, a vertex block, and a global block. Each full GNN block also contains three update functions ϕ^e, ϕ^v, and ϕ^u of the following forms [9]:

$$\mathbf{e}'_k = \phi^e(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}), \quad (1)$$

$$\mathbf{v}'_i = \phi^v(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}), \quad (2)$$

$$\mathbf{u}' = \phi^u(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u}), \quad (3)$$

and three aggregation functions $\rho^{e\to v}$, $\rho^{e\to u}$, and $\rho^{v\to u}$ of the following forms [9]:

$$\bar{\mathbf{e}}'_i = \rho^{e\to v}(\mathscr{E}'_i), \tag{4}$$

$$\bar{\mathbf{e}}' = \rho^{e\to u}(\mathscr{E}'), \tag{5}$$

$$\bar{\mathbf{v}}' = \rho^{v\to u}(\mathscr{V}'), \tag{6}$$

where $\mathscr{E}'_i = \{(\mathbf{e}'_k, r_k, s_k)\}_{r_k=i,k=1:N^e}$, $\mathscr{V}' = \{\mathbf{v}'_i\}_{i=1:N^v}$, and $\mathscr{E}' = \bigcup_i \mathscr{E}'_i = \{(\mathbf{e}'_k, r_k, s_k)\}_{k=1:N^e}$.

According to [8], a collection of random variables $\mathscr{X} = \{x_{\mathbf{v}_1}, \ldots, x_{\mathbf{v}_{N^v}}\}$ depending entirely on their local neighbors in the graph are considered to be in a Markov random field (MRF). Pairwise MRF (pMRF) models are generally used to represent the vertex relations and infer the vertex posterior beliefs [29]. A pMRF over a graph is associated with a set of vertex potentials as well as edge potentials [30]. The overall distribution is the normalized product of all vertex and edge potentials [29]:

$$P(\mathscr{X}) = \frac{1}{Z}\prod_{\mathbf{v}_i\in\mathscr{V}} \varphi(x_{\mathbf{v}_i}) \prod_{(\mathbf{e},r,s)\in\mathscr{E}} \psi(x_{\mathbf{v}_r}, x_{\mathbf{v}_s}), \tag{7}$$

where Z is a normalizer, φ represents the vertex potential of $\mathbf{v}_i$, and ψ denotes the edge compatibility between the sender $\mathbf{v}_s$ and the receiver $\mathbf{v}_r$. Belief propagation (BP) is commonly used to approximate the posterior belief of a given vertex. The message $m^{(t)}_{rs}(x_{\mathbf{v}_r})$ sent from $\mathbf{v}_s$ to $\mathbf{v}_r$ in the t-th iteration is [29]:

$$m^{(t)}_{rs}(x_{\mathbf{v}_r}) = \varphi(x_{\mathbf{v}_s})\psi(x_{\mathbf{v}_r}, x_{\mathbf{v}_s}) \prod_{k\in\mathscr{N}(\mathbf{v}_s)\setminus\mathbf{v}_r} m^{(t-1)}_{sk}(x_{\mathbf{v}_s}), \tag{8}$$

where $\mathscr{N}(\mathbf{v}_s)$ is the neighborhood system of $\mathbf{v}_s$. The posterior belief of a vertex $x_{\mathbf{v}_i}$ is proportional to the product of the factor and the messages from the variables, namely

$$P^{(t)}(x_{\mathbf{v}_i}) \propto \varphi_i(x_{\mathbf{v}_i}) \prod_{k\in\mathscr{N}(\mathbf{v}_i)} m^{(t)}_{ik}(x_{\mathbf{v}_i}). \tag{9}$$

It can be found from (8) and (9) that the posterior belief of $x_{\mathbf{v}_i}$ is only related to its vertex potential and the edge compatibility between $\mathbf{v}_i$ and its neighbors. Therefore, the global attribute $\mathbf{u}$ in Fig. 5 can be omitted. The simplified GNN block is shown in Fig. 6. In this paper, each vertex is considered to have relations with its four closest neighboring vertexes. (1) and (2) can, therefore, be rewritten as

$$\mathbf{e}'_k = \phi^e(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}), \tag{10}$$

and

$$\mathbf{v}'_i = \phi^v(\bar{\mathbf{e}}'_i, \mathbf{v}_i), \tag{11}$$

where the multi-layer perceptron is used for ϕ. Our feature updater then produces an updated vertex feature $\mathbf{R_v}$ of size $(HW)\times C'$ and an updated edge feature $\mathbf{R_e}$ of size $(4HW)\times C'$. These updated features are then processed by an FC layer, a multiplication and a reshaping operator, as shown in Fig. 3, to generate an updated tensor (feature representation) T' of size $H\times W\times C'$, which can be considered as a refinement of the input tensor T. Considering the balance between the performance improvement and the memory cost, we set the output feature channels to half of the input feature channels, *i.e.*, $C' = \frac{1}{2}C$.

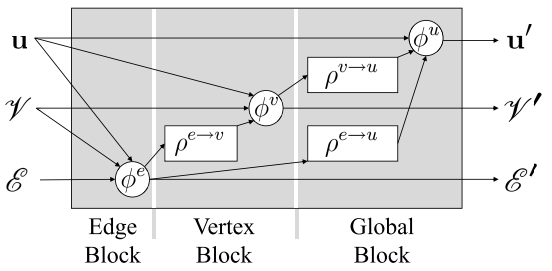


Fig. 5. An illustration of a full GNN block, which consists of an edge block, a vertex block and a global block.

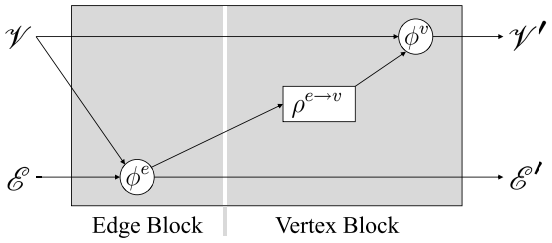


Fig. 6. An illustration of our proposed simplified GNN block, which only consists of an edge block and a vertex block.

B. GAL-DeepLabv3+

We implement the developed GAL in DeepLabv3+ [10] and build a new architecture referred to as GAL-DeepLabv3+, as shown in Fig. 2. We adopt several residual blocks [31] as the backbone, and the output feature size from Block5 (the last block) of the adopted backbone is $\frac{H}{16}\times\frac{W}{16}\times C_5$. Then, our GAL takes the feature from Block5 of the backbone as input and outputs the refined feature, which is then concatenated with the input feature and sent to the following ASPP module and the decoder, separately.

IV. Experimental Results and Discussion

A. Dataset and Experimental Setup

We use our recently published road pothole detection dataset [3] to compare the performance of the nine SoTA CNNs mentioned in Section II-B and our introduced GAL-DeepLabv3+. This dataset contains 55 samples of RGB images (RGB), subpixel disparity images (Disp), and transformed disparity images (T-Disp), which correspond to 14 potholes. The image resolution is 800 × 1312 pixels. The 13th and 14th potholes have only one sample each, and therefore they are only used for CNN cross-validation. The remaining 53 samples are divided into 12 sets, which correspond to 12 different potholes. The sample numbers in these 12 sets are: 13, 9, 5, 4, 3, 2, 3, 3, 2, 2, 2, and 5.

In our experiments, we employ the 12-fold cross-validation strategy [32] to quantify the CNNs' performances, *i.e.*, the performance of each CNN is evaluated 12 times. Each time, a CNN was trained using RGB, Disp, and T-Disp,

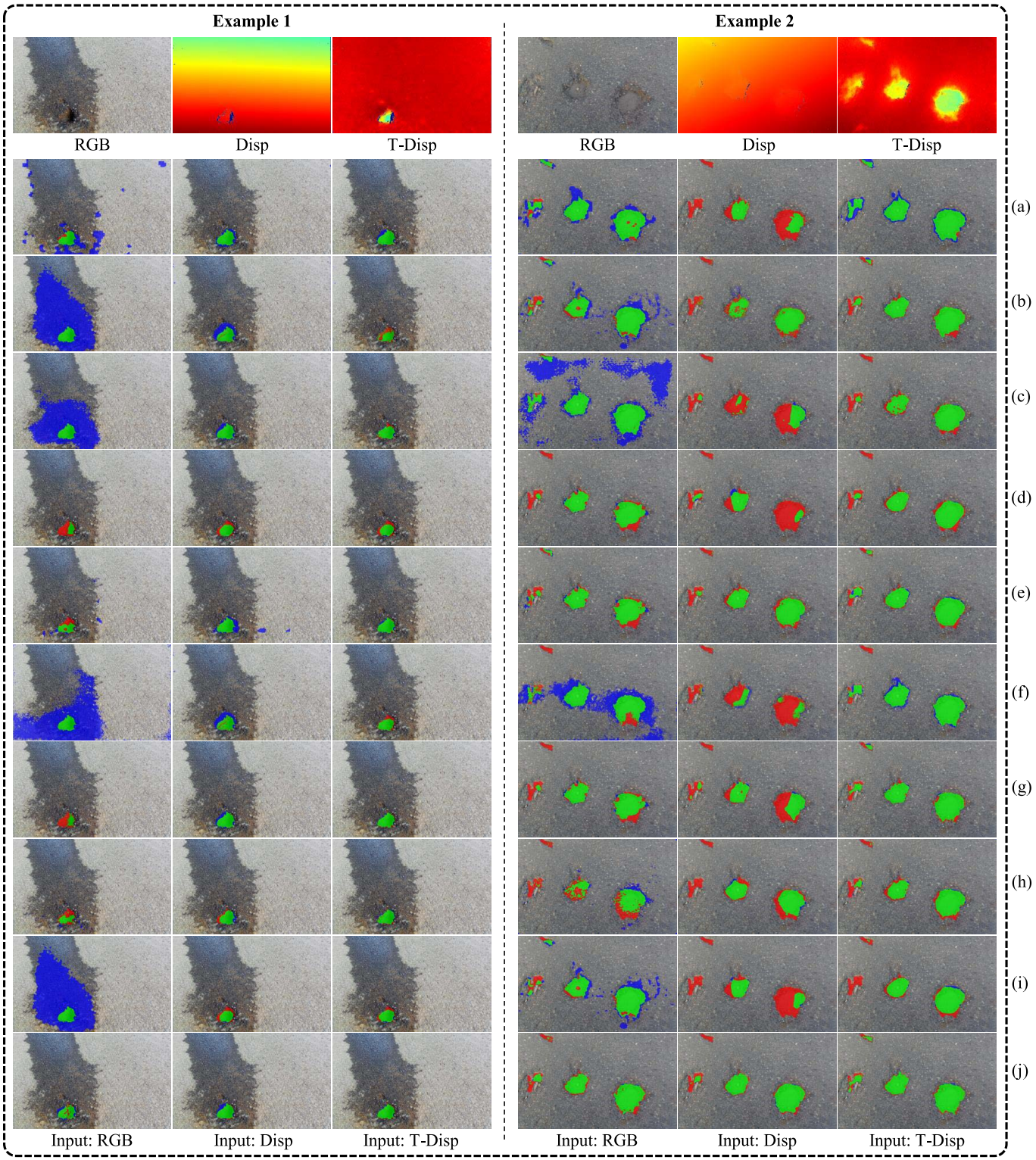


Fig. 7. Examples of the experimental results of the nine SoTA CNNs and GAL-DeepLabv3+: (a) FCN [17]; (b) U-Net [18]; (c) DenseASPP [24]; (d) DUpsampling [26]; (e) GSCNN [28]; (f) SegNet [19]; (g) DeepLabv3+ [10]; (h) PAN [25]; (i) ESPNet [27]; (j) our developed GAL-DeepLabv3+, where the true-positive, false-positive, and false-negative pixels are shown in green, blue and red, respectively.

separately. To quantify the CNNs' performances, we compute the pixel-level precision (Pre), recall (Rec), accuracy (Acc), F-score (Fsc), and intersection over union (IoU). We compute the mean value across the 12 sets for each metric, denoted as mPre, mRec, mAcc, mFsc and mIoU.

Additionally, stochastic gradient descent with momentum (SGDM) optimizer [33] is used for CNN training. The maximum epoch for the experiments on RGB, Disp, and T-Disp is set to 150, 100, and 100, respectively. Each network is trained on two NVIDIA GeForce RTX 2080Ti GPUs.

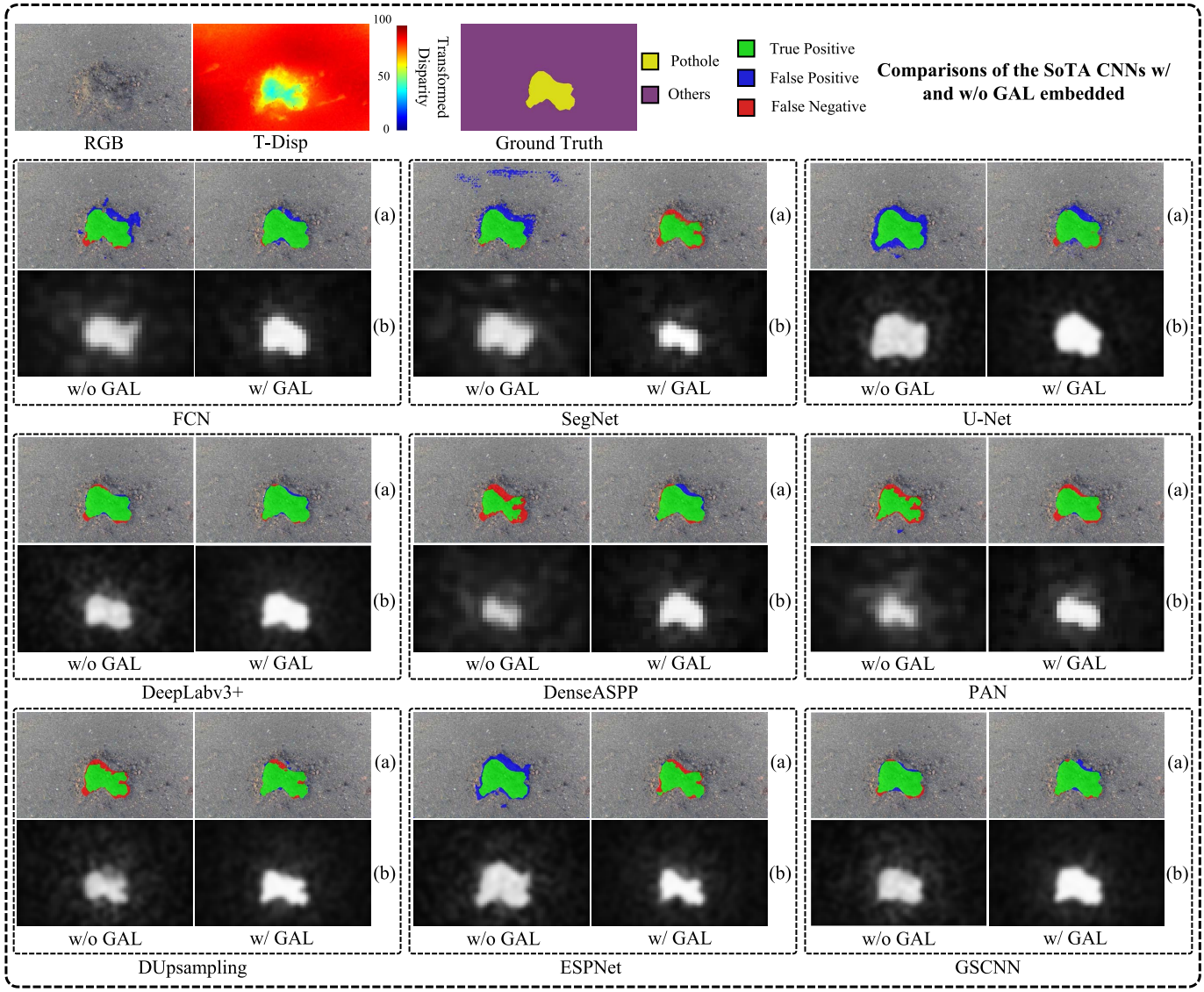


Fig. 8. An example of the experimental results of the nine SoTA CNNs without and with GAL embedded: (a) pothole detection results; (b) the corresponding mean activation maps of the features after the last layers of the encoders.

We also leverage common training data augmentation techniques, such as random flip, rotation, and translation, to further improve the CNNs' robustness and accuracy.

Next, we conduct ablation studies in Section IV-B to demonstrate the effectiveness of our GAL. Then, a road pothole detection benchmark that provides a detailed performance comparison between the nine SoTA CNNs and our GAL-DeepLabv3+ on the three modalities of training data is presented in Section IV-C. To further understand how our GAL improves the overall performance for road pothole detection, we implement it in all SoTA CNNs and analyze the feature variation with and without our GAL, as discussed in Section IV-D.

TABLE I

ABLATION STUDIES: (A) SHOWS THE BASELINE RESULTS; (B) SHOWS THE RESULTS OF THE BASELINE WITH GAL EMBEDDED; AND (C) SHOWS THE RESULTS OF THE BASELINE WITH RESNET-101 USED AS THE BACKBONE. THE BEST RESULTS ARE SHOWN IN BOLD TYPE

Setups		Evaluation Metrics		
Backbone	GAL	mAcc (%)	mFsc (%)	mIoU (%)
(A) ResNet-50	–	98.453	81.479	69.011
(B) ResNet-50	✓	**98.669**	**85.636**	**75.008**
(C) ResNet-101	–	98.581	85.167	74.228

B. Ablation Study

We adopt DeepLabv3+ [10] as the baseline to conduct ablation studies because it outperforms all other SoTA CNNs. It inputs T-Disp because this modality of vision data is most informative [4]. Table I shows the results of the ablation study, where (A) shows the baseline performance, (B) shows the performance of the proposed approach, and (C) shows the performance of the baseline with ResNet-101 as the backbone (abbreviated as ResNet101-DeepLabv3+). Compared

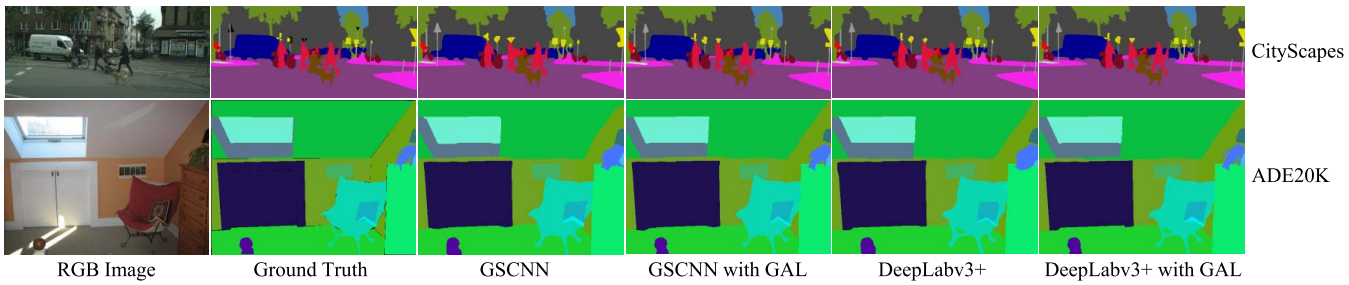


Fig. 9. Examples of the experimental results of GSCNN [28] and DeepLabv3+ [10] with and without GAL embedded.

TABLE II

THE ROAD POTHOLE DETECTION BENCHMARK CONDUCTED WITH NINE SOTA CNNS AND OUR PROPOSED GAL-DEEPLABV3+ ON THREE MODALITIES OF TRAINING DATA. THE BEST RESULTS ARE SHOWN IN BOLD TYPE

Approach	Input	mPre (%)	mRec (%)	mAcc (%)	mFsc (%)	mIoU (%)
FCN [17]	RGB	57.029	76.324	94.881	63.670	47.647
	Disp	85.465	49.456	97.126	60.587	43.812
	T-Disp	73.970	80.874	97.386	76.940	62.698
SegNet [19]	RGB	35.092	74.886	89.356	46.268	30.709
	Disp	79.007	41.299	96.578	51.603	35.211
	T-Disp	70.463	79.980	97.216	74.065	58.982
U-Net [18]	RGB	70.618	63.459	95.925	62.967	47.317
	Disp	75.872	57.253	97.062	63.886	47.205
	T-Disp	82.334	67.034	97.833	73.573	58.438
DeepLabv3+ [10]	RGB	78.802	66.107	97.399	70.938	55.305
	Disp	66.993	69.968	96.798	67.314	50.966
	T-Disp	**89.819**	75.154	98.453	81.479	69.011
DenseASPP [24]	RGB	41.818	63.212	91.113	48.929	33.191
	Disp	68.513	63.959	96.777	64.249	47.729
	T-Disp	88.914	65.637	98.018	74.398	59.824
PAN [25]	RGB	73.157	47.839	96.640	57.417	42.207
	Disp	74.522	45.762	96.352	53.290	37.146
	T-Disp	87.359	66.592	97.944	72.657	59.972
DUpsampling [26]	RGB	75.051	66.281	97.230	69.472	53.515
	Disp	80.446	57.076	97.311	66.132	49.618
	T-Disp	86.160	72.365	98.204	77.931	64.225
ESPNet [27]	RGB	74.547	63.694	96.133	64.850	49.327
	Disp	85.953	51.716	97.354	63.846	47.245
	T-Disp	87.190	67.644	98.038	75.358	61.013
GSCNN [28]	RGB	81.135	58.818	97.713	66.571	52.712
	Disp	86.275	54.612	97.407	65.481	48.992
	T-Disp	81.165	80.171	98.289	80.389	67.635
GAL-DeepLabv3+ (Ours)	RGB	83.900	69.803	97.740	74.166	60.097
	Disp	79.847	68.710	97.473	72.428	57.812
	T-Disp	89.713	**82.205**	**98.669**	**85.636**	**75.008**

to ResNet50-DeepLabv3+, our developed GAL-DeepLabv3+ presents a much better performance. One exciting fact is that our developed GAL-DeepLabv3+ with ResNet-50 as the backbone performs even slightly better than ResNet101-DeepLabv3+. This demonstrates the effectiveness of our proposed GAL.

C. Road Pothole Detection Benchmark

This subsection presents a road pothole detection benchmark with quantitative and qualitative comparisons among nine SoTA CNNs and GAL-DeepLabv3+ trained on the three modalities of vision data. Some examples of the experimental results are shown in Fig. 7. It can be observed that the CNNs

trained on RGB can be easily misled by noise, such as a stain on the road (see Fig. 7, Example 1). The CNNs trained on Disp perform slightly better but still produce many false-negative predictions (see Fig. 7, Example 2). By comparison, the CNNs trained on T-Disp perform much more robustly. This is due to that the disparity transformation algorithm makes the damaged road regions become highly distinguishable [3]. Furthermore, Fig. 7 shows that our developed GAL-DeepLabv3+ outperforms all other SoTA CNNs on all three modalities of vision data.

Additionally, the quantitative comparisons are given in Table II, where it can be seen that the mIoU increases by ∼11-28%, while the mFsc goes up by ∼8-28% when the CNNs are trained on T-Disp rather than RGB. These results further validate the effectiveness of the disparity transformation algorithm, which converts road disparity information into a more informative format. Furthermore, GAL-DeepLabv3+ outperforms all other SoTA CNNs by ∼6-17% on the mIoU and by ∼4-13% on the mFsc, when trained on T-Disp. This demonstrates that GAL can effectively improve the road pothole detection performance.

D. Further Discussion on GAL

To further understand how GAL improves the CNN's overall performance for road pothole detection, we implement it in each of the nine SoTA CNNs. It should be noted here that we implement GAL after the encoder's last layer for each CNN. The quantitative and qualitative comparisons are given in Table III and Fig. 8, respectively. It can be observed that the CNNs with GAL embedded generally perform better than themselves without GAL embedded.

To explore how GAL refines the feature representations, we visualize the mean activation maps of the features output from the encoders' last layers with and without GAL embedded, as shown in Fig. 8(b). These maps suggest that GAL can help CNNs concentrate more on the target (road pothole) areas. We believe this is because GAL can be considered as a weight modulation operator, which can effectively augment the activation values in the target areas and reduce the activation values in the background areas. A typical convolutional layer can be formulated as follows:

$$y(x) = \sum_{n=1}^{N} w_n \cdot x(n). \tag{12}$$

Now, with GAL embedded, we can formulate this process as follows:

$$y(x) = \sum_{n=1}^{N} w_n \cdot \left(\bar{x}(n) \cdot \Delta w_e\right), \tag{13}$$

where Δw_e is obtained from the updated edge features, and $\bar{x}(n)$ is the updated vertex features from $x(n)$. Based on (12) and (13), we can conclude that GAL can be considered as an effective and efficient weight modulation operator, which can greatly refine the feature representations, thus improving the CNN's overall performance for road pothole detection.

TABLE III

THE EXPERIMENTAL RESULTS OF THE SOTA CNNS WITH AND WITHOUT GAL EMBEDDED. THE BEST RESULTS FOR EACH CNN ARE SHOWN IN BOLD TYPE

Approach	mAcc (%)	mFsc (%)	mIoU (%)
FCN [17]	97.386	76.940	62.698
GAL-FCN	**98.016**	**80.358**	**67.205**
SegNet [19]	97.216	74.065	58.982
GAL-SegNet	**97.880**	**79.275**	**65.703**
U–Net [18]	97.833	73.573	58.438
GAL-U–Net	**97.958**	**77.306**	**63.040**
DeepLabv3+ [10]	98.453	81.479	69.011
GAL-DeepLabv3+	**98.669**	**85.636**	**75.008**
DenseASPP [24]	98.018	74.398	59.824
GAL-DenseASPP	**98.177**	**79.856**	**66.505**
PAN [25]	97.944	72.657	59.972
GAL-PAN	**98.102**	**79.194**	**65.592**
DUpsampling [26]	98.204	77.931	64.225
GAL-DUpsampling	**98.349**	**81.426**	**68.713**
ESPNet [27]	98.038	75.358	61.013
GAL-ESPNet	**98.165**	**79.747**	**66.354**
GSCNN [28]	98.289	80.389	67.635
GAL-GSCNN	**98.486**	**84.329**	**72.954**

TABLE IV

THE EXPERIMENTAL RESULTS OF GSCNN [28] AND DEEPLABV3+ [10] WITH AND WITHOUT OUR GAL EMBEDDED ON THE CITYSCAPES [34] AND ADE20K [35] DATASETS. THE BEST RESULTS FOR EACH CNN ARE SHOWN IN BOLD TYPE

Approach	Cityscapes [34]		ADE20K [35]	
	mFsc (%)	mIoU (%)	mFsc (%)	mIoU (%)
GSCNN [28]	86.383	76.829	59.727	42.556
GAL-GSCNN	**88.920**	**80.186**	**62.962**	**45.934**
DeepLabv3+ [10]	87.198	77.398	60.685	43.528
GAL-DeepLabv3+	**89.802**	**81.537**	**64.120**	**47.291**

V. DISCUSSION

Potholes are a common type of road distress. The detection of other categories of road distresses, such as cracks, typically requires different kinds of computer vision algorithms. For example, the SoTA road crack detection algorithms [36]–[38] commonly leverage image classification CNNs instead of semantic segmentation CNNs to identify whether an image patch contains cracks because road cracks cannot be easily identified from depth/disparity images, and the semantic segmentation CNN is challenging to retain highly accurate semantic content for such tiny objects. Furthermore, although different types of road distress can be recognized and classified with SoTA object detection algorithms, such as YOLO-v3 [39] utilized in [40], such road distress detection results can only be at instance level instead of pixel level. The measurement of a

road pothole's volume typically requires pixel-level predictions instead of a region of interest.

In addition to road pothole detection, our introduced GAL can also be embedded in CNNs to solve other semantic image segmentation problems. To demonstrate its feasibility in other challenging multi-class scene understanding applications, we train GSCNN [28] and DeepLabv3+ [10] both with and without GAL embedded on the Cityscapes [34] and ADE20K [35] datasets. The Cityscapes dataset [34] was created for semantic urban scene understanding, while the ADE20K dataset [35] (including diverse scenarios) was created for general scene parsing. The qualitative and quantitative results are given in Fig. 9 and Table IV, respectively. It can be observed that both CNNs with GAL embedded can produce more accurate results, where the mIoU increases by $\sim$4% and the mFsc increases by $\sim$3%. These results suggest the generalizability of our proposed GAL for other challenging semantic segmentation tasks. Therefore, we believe our proposed GAL can be easily incorporated into any existing CNN to achieve SoTA semantic scene understanding performance.

VI. Conclusion and Future Work

In this paper, we provided a comprehensive study on road pothole detection including building up a benchmark, developing a novel layer based on GNN, and proposing an effective and efficient CNN for road pothole detection. Experiments verify that our proposed GAL can effectively refine the feature representations and thus improve the overall semantic segmentation performance. Moreover, the transformed disparity images can make road potholes highly distinguishable and benefit all CNNs for road pothole detection. Compared with the SoTA CNNs, our proposed GAL-DeepLabv3+ achieves superior performance and produces more robust and accurate results. We believe that the provided benchmark and our proposed models are helpful to stimulate further research in this area. Furthermore, our proposed techniques can also be employed to solve other general semantic segmentation/understanding problems. In the future, we will continue to explore graph-based architectures that can optimize the feature representations effectively and efficiently for semantic segmentation.

References

[1] S. Mathavan, K. Kamal, and M. Rahman, "A review of three-dimensional imaging technologies for pavement distress detection and measurements," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2353–2362, Oct. 2015.

[2] R. Fan, U. Ozgunalp, Y. Wang, M. Liu, and I. Pitas, "Rethinking road surface 3-D reconstruction and pothole detection: From perspective transformation to disparity map segmentation," *IEEE Trans. Cybern.*, early access, Mar. 24, 2021, doi: 10.1109/TCYB.2021.3060461.

[3] R. Fan, U. Ozgunalp, B. Hosking, M. Liu, and I. Pitas, "Pothole detection based on disparity transformation and road surface modeling," *IEEE Trans. Image Process.*, vol. 29, no. 1, pp. 897–908, Aug. 2019.

[4] R. Fan, H. Wang, M. J. Bocus, and M. Liu, "We learn better road pothole detection: From attention aggregation to adversarial domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 285–300.

[5] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Adv. Eng. Inform.*, vol. 29, no. 2, pp. 196–210, 2015.

[6] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with Gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 109–117.

[7] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1529–1537.

[8] R. Fan, X. Ai, and N. Dahnoun, "Road surface 3D reconstruction based on dense subpixel disparity map estimation," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3025–3035, Jun. 2018.

[9] P. W. Battaglia *et al.*, "Relational inductive biases, deep learning, and graph networks," 2018, *arXiv:1806.01261*. [Online]. Available: https://arxiv.org/abs/1806.01261

[10] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[11] M. R. Jahanshahi, F. Jazizadeh, S. F. Masri, and B. Becerik-Gerber, "Unsupervised approach for autonomous pavement-defect detection and quantification using an inexpensive depth sensor," *J. Comput. Civil Eng.*, vol. 27, no. 6, pp. 743–754, 2013.

[12] U. Ozgunalp, "Vision based lane detection for intelligent vehicles," Ph.D. dissertation, Dept. Elect. Electron. Eng., Univ. Bristol, Bristol, U.K., 2016.

[13] H. Wang, R. Fan, Y. Sun, and M. Liu, "Dynamic fusion module evolves drivable area and road anomaly detection: A benchmark and algorithms," *IEEE Trans. Cybern.*, early access, Mar. 24, 2021, doi: 10.1109/TCYB.2021.3064089.

[14] R. Fan and M. Liu, "Road damage detection based on unsupervised disparity map segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4906–4911, Nov. 2020.

[15] A. Dhiman and R. Klette, "Pothole detection using computer vision and learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3536–3550, Aug. 2020.

[16] H. Wu *et al.*, "Road pothole extraction and safety evaluation by integration of point cloud and images derived from mobile mapping sensors," *Adv. Eng. Informat.*, vol. 42, Oct. 2019, Art. no. 100936.

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

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[19] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[21] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[22] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[23] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, pp. 1–14, 2017.

[24] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3684–3692.

[25] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2018, pp. 1–13.

[26] Z. Tian, T. He, C. Shen, and Y. Yan, "Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3126–3135.

[27] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 552–568.

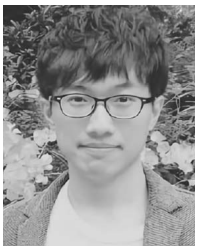
[28] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-SCNN: Gated shape CNNs for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5229–5238.

[29] W. Gatterbauer, "The linearization of belief propagation on pairwise Markov random fields," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 3747–3753.
[30] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.
[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[32] S. Geisser, "The predictive sample reuse method with applications," *J. Amer. Stat. Assoc.*, vol. 70, no. 350, pp. 320–328, 1975.
[33] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[34] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.
[35] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 633–641.
[36] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3708–3712.
[37] H. Oliveira and P. L. Correia, "Automatic road crack detection and characterization," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 155–168, Mar. 2012.
[38] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3434–3445, Dec. 2016.
[39] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *CoRR*, pp. 1–6, 2018.
[40] Y. Du, N. Pan, Z. Xu, F. Deng, Y. Shen, and H. Kang, "Pavement distress detection and classification based on YOLO network," *Int. J. Pavement Eng.*, pp. 1–14, Jan. 2020, doi: 10.1080/10298436.2020.1714047.

Rui Fan (Member, IEEE) received the B.Eng. degree in automation from Harbin Institute of Technology in 2015 and the Ph.D. degree in electrical and electronic engineering from the University of Bristol in 2018.

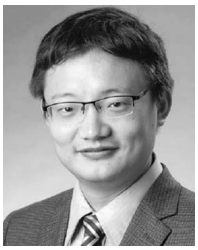
He worked as a Research Associate with The Hong Kong University of Science and Technology from 2018 to 2020 and a Postdoctoral Fellow at the University of California at San Diego, San Diego, from 2020 to 2021. He is currently a Full Research Professor with Tongji University and Shanghai Research Institute for Intelligent Autonomous Systems. His research interests include computer vision, machine learning, and robotics.

Hengli Wang (Graduate Student Member, IEEE) received the B.E. degree in mechatronics engineering from Zhejiang University, Hangzhou, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electronic and Computer Engineering, Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong, SAR, China. His research interests include computer vision, robot navigation, and deep learning.



Yuan Wang received the B.Sc. degree in telecommunication engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2016, and the M.Sc. degree in telecommunication engineering from The Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2017.

From 2018 to 2019, he worked as a Research Assistant with the Robotics and Multiperception Laboratory, HKUST. Since 2020, he has been working with SmartMore, Shenzhen, China. His research interests include semantic segmentation, 2D/3D object detection, and graph neural networks.



Ming Liu (Senior Member, IEEE) received the B.A. degree in automation from Tongji University, Shanghai, China, in 2005, and the Ph.D. degree from the Department of Mechanical and Process Engineering, ETH Zürich, in 2013, supervised by Prof. R. Siegwart.

He is currently an Associate Professor with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. His research interests include dynamic environmental modeling, 3D mapping, machine learning, and visual control.

Ioannis Pitas (Fellow, IEEE) received the Diploma and Ph.D. degrees in electrical engineering from the Aristotle University of Thessaloniki (AUTH), Thessaloniki, Greece.

Since 1994, he has been a Professor with the Department of Informatics, AUTH, where he is the Director of the Artificial Intelligence and Information Analysis Laboratory. He served as a Visiting Professor at several universities. He leads the big European H2020 Research and Development Project MULTIDRONE. He is an AUTH Principal Investigator in H2020 Research and Development Projects Aerial Core and AI4Media. He is the Chair of the Autonomous Systems Initiative. He is the Head of the EC funded AI doctoral School of Horizon2020 EU-funded Research and Development Project AI4Media (1 of the 4 in Europe). His current interests are in the areas of computer vision, machine learning, autonomous systems, and image/video processing. He is an IEEE Distinguished Lecturer and a Fellow of EURASIP.