Lightweight Semantic Segmentation for Road-Surface Damage Recognition Based on Multiscale Learning

SEUNGBO SHIM\textsuperscript{1,2} AND GYE-CHUN CHO\textsuperscript{3}
\textsuperscript{1}Future Infrastructure Research Center, Korea Institute of Civil Engineering and Build Technology (KICT), Goyang 10223, South Korea
\textsuperscript{2}Department of Civil and Environmental Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea
\textsuperscript{3}Department of Civil and Environmental Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea
Corresponding author: Gye-Chun Cho (gyechun@kaist.edu)

This work is supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) through the Ministry of Land, Infrastructure, and Transport under Grant 20CTAP-C157234-01.

ABSTRACT With an aging society, the demand for personal mobility for disabled and aging people is increasing. As of 2017, the number of electric wheelchairs in Korea was 90,000 according to the domestic government statistics and has since increased continuously. However, people with disabilities and seniors are more likely to be involved in accidents while driving because their judgment and coordination are inferior to those of ordinary people. One of the factors that could lead to accidents is the interference in the vehicle-steering control owing to unbalanced road-surface conditions. In this paper, we introduce a lightweight semantic segmentation algorithm that can recognize the area with road-surface damage through images at high speed to prevent occurrence of such accidents. To test the algorithm, an experiment was conducted in which more than 1,500 training data and 150 validation data, including road-surface damage, were newly created. By using the data, we propose a new deep neural network composed of only encoder type, unlike the auto-encoding type consisting of encoder and decoder. To evaluate the performance of the proposed algorithm, we considered four metrics of the accuracy and two metrics of the speed. Unlike the conventional method, this deep neural network method shows improvement in all of the accuracy index, a 85.7% decrease in parameters, and an 6.1% increase in computational speed. The application of such a high-speed algorithm is expected to improve safety in personal transportation.

INDEX TERMS Road-surface damage detection, lightweight deep learning, driving safety, semantic segmentation, personal mobility.

I. INTRODUCTION

A. DEMANDS OF TRANSPORTATION SAFETY

Autonomous personal mobility vehicles (APMV) could be a means to encourage social activities among senior people in the future. Such a vehicle could be an important technology for helping people with disabilities and the elderly to live an independent life while becoming a safe and reliable means of transportation. Currently, Japan is already facing an unprecedented increase in its aging society [1]. As the number of young people decreases, the production activity population decreases significantly and the public transportation infrastructure, which is fundamentally necessary for people to survive, is lacking. This situation can be considered as a very serious problem for government institute [2]. As a solution to this problem, the production of personal mobile vehicles as a transportation alternative is increased. In recent years, the use of personal transportation, such as electric wheelchairs and electric scooters, is gradually increasing to improve the quality of life for the disabled. However, the elderly or severely disabled still have a high possibility of accidents because they cannot respond to emergency circumstances immediately and appropriately. To reduce the probability of accidents and prevent them in advance, vehicles must be equipped with various sensors and must incorporate automated driving technology [3], [4]. To ensure safety of a personal transportation vehicle from accidents, obstacle recognition technology based on various sensors, such as LiDARs and cameras, is necessary while driving [5]. Technologies that can precisely recognize obstacles, such as cars, people, bicycles,
and motorcycles, that may be on the road are typical examples. However, in the case of personal transportation used by the elderly and disabled, steering control is more affected by the road-surface condition because the wheel size is smaller than that of a general vehicle. Therefore, a small-sized vehicle for personal transportation must have a technology capable of recognizing the road-surface condition in real time [6].

Moreover, in terms of road maintenance, road-surface condition is critical for traffic safety. Irregular road-surface conditions, such as potholes and cracks, lead to vehicle collisions and other major accidents [7]. One of the most fundamental tasks is to detect and repair the pavement crack to ensure safety on roads and highways [8]. The first step to keep the traffic safe is to locate the areas with cracks during road maintenance. In early road defect detection methods, people analyzed the road images captured by using expensive equipment and vehicles. However, only a small part of the road is covered by using this method owing to the lack of government budget. In addition, the analysis requires experts in related fields, and only a few people can inspect the road-surface condition. However, with the recent emergence of deep learning in computer vision, artificial intelligence has remarkably improved the image processing technology [9]. This technology is being rapidly applied to various fields and is necessary for developing a safe personal vehicle as well as for efficient road maintenance.

B. ROAD-DAMAGE DETECTION USING ARTIFICIAL INTELLIGENCE

The methods for recognizing road damage through artificial intelligence can be divided into unsupervised and supervised learning. In the case of unsupervised learning, the algorithm proposed by Akagic et al. first applied the Otus technique to the grey-level image [10]. The algorithm showed exceptional performance in low signal-to-ratio situations through unsupervised learning. Li et al. proposed an artificial intelligence algorithm using a multiscale fusion method that found cracks on the road [11]. The algorithm used a Gaussian blurring technique and a windowed minimal-intensity-path method to extract candidate groups of cracks in an image and found the crack regions by combining candidate groups obtained from multiple scale images. Amhaz et al. developed the algorithm with the assumption that the cracked area in the image would be a path and would have a smaller loss value than the path in the background area [12]. Specifically, the minimal path-selection algorithm was used, and then the skeleton and thickness information of the cracks were estimated.

In terms of supervised learning, Shi et al. proposed a crack-detection algorithm based on the random structure forest method [13]. This algorithm was characterized by efficient operation in crack images with uneven or complex shapes. In addition, the 118 urban road crack images used in the paper were released as crack forest dataset (CFD), contributing to the development of many other studies. Unlike the classical method introduced earlier, the supervised learning method using deep learning has been actively applied in these days. Among them, the method of recognizing an object can be mainly classified into an object detection-based method and a pixel-level segmentation method.

There are many types of object detection algorithms, among which the Faster region-based convolution neural network (Faster R-CNN), SSD, and YOLO are most commonly used [14]–[16]. Li et al. proposed a road-surface-damage object-recognition algorithm using the Faster R-CNN method [17] to distinguish six types of road-surface damages and simultaneously recognize a damaged area in the image. A total of 5,966 images were used and the average accuracy was 96.3%, showing excellent performance. Maeda et al. compared the performance of backbone network using MobileNet and Inception v2 and by utilizing SSD as the basic algorithm for high-speed computation [18]–[20]. For this experimental comparison, 9,000 images were taken with a smart phone and 15,000 objects were marked. The comparison showed that MobileNet utilizes 30.6 ms but did not work properly in detecting long and curved shapes such as cracks because this object detection algorithm is suitable for recognizing objects such as vehicles or people [21].

The pixel-level segmentation method uses CNNs as the main operation method, similar to the aforementioned object detection methods; however, the most significant difference is that every pixel is classified. Fan et al. introduced an algorithm by dividing the input image into patches and performs multiple CNN and max-pooling operations [22]. Finally, 1 × 1 – 5 × 5 binary images were obtained as output after passing through the fully connected layer in three steps. A total of 96 images were used for training and 60 images were used for validation. As a result, the accuracy was over 90%. Jenkins et al. proposed an algorithm that uses U-Net based on the auto-encoder method [23], [24]. It consists of two stages: an encoder and a decoder. In the encoder stage, the algorithm mainly played the role of generating feature maps. In the decoder stage, the feature maps were used to estimate the probability distribution of each pixel unit for segmentation. The CFD was used, in particular, 80 and 20 images were used for training and validation, respectively. The result showed 92.46% accuracy, 82.82% reproducibility, and 87.38% F1-Score. Similarly, Zou et al. proposed an algorithm called DeepCrack [25], a deep neural network, that was designed so that the weight was updated for each scale by using a skip layer between the encoder and decoder stages. As such, the performance was greatly improved compared to that of other algorithms. Bang et al. proposed a deep neural network algorithm of the auto-encoder type using several images collected through a black box camera [26]. By using these images, the crack area was marked on 427 and 100 images for training and validation, respectively. By using the transfer learning technique, the results of the segmentation algorithm showed an average accuracy of 77.68%.

According to previous studies, we conclude that the auto-encoder type of a neural network structure, such as U-Net and DeepCrack, has been used to increase the accuracy of pixel-level segmentation. In this article, however,
we propose a new type of neural network structure. The structure of the pixel-level segmentation currently being used consists of encoder and decoder stages that perform up-sampling or de-convolution operations, respectively. Such a structure has high accuracy in dividing an object in units of pixels, but its computational speed decreases as the neural network deepens. To address this disadvantage, we propose a lightweight deep neural network structure that performs pixel-level segmentation using only the encoder stage but shows better performance than the auto-encoder method.

In summary, our approach has three contributions. The first is to regenerate a new set of training data to recognize road-surface damage. Second, unlike the auto-encoder type, it suggested a new method to reduce calculation time using only the encoder type. Third, developing an algorithm with high accuracy is possible even with the encoder type of improving the neural network structure. To describe those contributions in detail, we first explain the set of images acquired for multiscale supervised learning. Second, the structure of the auto-encoder type and the algorithm method using only the encoder step are described. Third, we provide an analysis of the experimental results including our approach. Finally, we discuss and summarize this paper.

II. IMAGE DATA OF ROAD-SURFACE DAMAGE
A. VARIOUS IMAGES OF ROAD-SURFACE DAMAGE
The damage caused to the road surface could be of various forms, such as horizontal and vertical cracks, tortoise cracks, potholes, rutting, and labeling. The goal of the proposed algorithm is dependent on the developer’s intention of which types of road damage to recognize. Jo et al. designed an algorithm to recognize only potholes, and the algorithm by Chun et al. was designed to detect both cracks and potholes [27], [28]. Eisenbach et al. classified road damage into five types, such as cracks, potholes, and patches, according to the regulations by the Road and Transportation Research Association, and developed a deep neural network to distinguish the type [29]. Finally, Maeda et al. captured and collected images of road-surface damage by using a smartphone and classified them into eight types [19]. A considerable amount of image data were collected and classified to recognize not only linear and fatigue cracks but also the shape of road markers. In this study, parts of the image data collected by Maeda et al. are utilized because the goal of this study was to develop an algorithm that acquires images from mobile devices and exploit them to recognize road-surface damage. Therefore, images acquired in a similar manner must be used considering the final field of application.

B. DATA REGENERATION FOR SEGMENTATION
The image of the road-surface damage considered in this study is shown in Fig. 1. Among the image data provided by Maeda et al., images containing longitudinal linear crack, lateral linear crack, alligator crack, rutting, bump, pothole, and separation were selected. Thus, a total of 1,650 images were secured, which are divided to two groups: 1,500 images for training and 150 images for validation. We used the images that are already captured but have different ground truths, as shown in Fig. 2; this dataset was created by the authors. Instead of the existing eight types of classification criteria, binary classification was used to determine whether roads are damaged. Therefore, only the nondamaged and damaged areas were distinguished using these image data.

Next, the damaged area was marked through segmentation instead of using bounding boxes. Regarding the accuracy of detecting the damaged areas, the segmentation method extracts the damaged area better than the method of using a bounding box. If the road-surface damage is indicated in the form of a bounding box, the damaged area in the image is included with the nondamaged area when taking a picture by using a black box camera for a vehicle. Actually, the bounding-box image includes a larger area than that indicated through the segmentation method, as shown in Fig. 2. Certainly, the segmentation method is advantageous in detecting the exact damage region from the image. Therefore, the labeling is performed using the LEAR Image
Annotation Tool [30]. One of our contributions is that the proposed algorithm is developed using newly created image data of road-surface damage, which are the re-generated ground truth.

C. DATA CONFIGURATION FOR MULTISCALE LEARNING
The multiscale grid-cell method was applied for training, and the image data were newly configured. In general, supervised learning is performed by using one ground-truth image as one input image. However, the supervised-learning method used in this study uses four ground-truth grid images as one input image. This expansion method creates a grid image by dividing the ground-truth image into cells of sizes $2 \times 2$, $4 \times 4$, $8 \times 8$, and $16 \times 16$. When the labelled region of the ground-truth image that belongs to the area of each cell occupies more than half of the area of a cell, the cell is regarded as an area comprising road-surface damage. Considering that the size of the input image used in this study is $576 \times 576$, the sizes of the grid image are $36 \times 36$, $72 \times 72$, $144 \times 144$, and $288 \times 288$. Therefore, we prepared and configured the image dataset for multiscale training by using four grid images for one input image, as shown in Fig. 3.

D. IMAGE AUGMENTATION
To improve the training effect through data augmentation, the images can be changed in three ways. First, the color brightness of the image is changed by considering cloudy weather when driving a vehicle. Second, a random blur is generated in the case of unclear images that can be captured when the camera is out of focus during high-speed driving. Finally, global contrast normalization is utilized to minimize the effect of lighting changes outside the vehicle. Hence, image data are augmented and are exploited so that the algorithm can work in various environments.

III. TWO TYPES OF ROAD-SURFACE-DAMAGE DETECTION
A. COMPARISON BETWEEN TWO TYPES OF SEGMENTATION ALGORITHMS
In this section, we compare the structures of segmentation for the auto-encoder and encoder methods. As shown in Fig. 5, the auto-encoder method is divided into the encoder and decoder stages to construct a deep neural network. In addition, Softmax is used as the activation function to finalize segmentation at the last neural network at the pixel level. Such a neural network structure often undergoes a process of...
reducing the size of the input image using the max-pooling or convolution operation at the encoder stage. In the decoder stage, the image size is enlarged using up-sampling or deconvolution. Then, the output image obtained at the last that is equal to size of the input image marks the area of road-surface damage.

However, the disadvantage of the auto-encoder structure is its decreasing computational speed with increasing number of neural network layers. To resolve this issue, we propose a method for simplifying the structure of the deep neural network. In our method, only the encoder step of the auto-encoder structure is used to improve the operation speed, as shown in Fig. 6. This method has the advantage that the amount of computation is only half completed when the deep neural network structure of the auto-encoder is symmetric. In addition, we introduced the classification stage in place of the decoder stage to enhance the accuracy. In the classification stage, several simple neural network layers are added to create multiple suboutputs, which are then combined to give the final output. The neural network in the classification stage comprises four suboutputs and is developed from the final neural networks of each block. These four suboutputs are indicated as suboutputs 1, 2, 3, and 4 in Fig. 6. Each suboutput has information on the areas of road-surface damage, and these suboutputs are summarized to determine whether the road surface is damaged.

In this study, we focused on the comparison of the decoder and classification stages when the same encoder stage is used in the deep neural network. For the sake of the comparison, four evaluation metrics were used that were introduced by Long et al. [31] to evaluate the accuracy of semantic segmentation algorithm. According to Long et al. [31], \( n_{ij} \) represents the number of all pixels of class \( i \) predicted to belong to class \( j \). \( n_{cl} \) indicates the number of classes, and \( t_i \) indicates the number of all pixels belonging to class \( i \). Equations (1) and (2) calculate the pixel and mean accuracies, respectively. Equations (3) and (4) show the mean Intersection over Union (IoU) and frequency weighted IoU, respectively, with respect to the area accuracy. Second, the average time required to process about 150 images used for verification was calculated to measure the calculation speed. In this calculation, the time to process the first image is excluded. Moreover, the number of parameters used to construct a deep neural network was calculated to compare the computing loads of the GPU.

\[
\sum_i t_i = \sum_j n_{ij} \quad (1) \\
\frac{1}{n_{cl}} \sum_i t_i \\
\frac{1}{n_{cl}} \sum_i t_i + \sum_j n_{ji} - n_{ii} \quad (3) \\
\frac{1}{n_{cl}} \sum_i t_i + \sum_j n_{ji} - n_{ii} \quad (4)
\]
B. IMPLEMENTATION AND TRAINING CONDITION

Three deep neural network models were used as the structures at the encoder stage: ResNet50, DetNet59, and DenseNet121 [32]–[34]. These models are commonly down-sampled in five stages, and thus when an input image of 576 × 576 is used, the size of the output feature is 18 × 18. To obtain the size of the final output image as 36 × 36, we utilized only up to four levels of these deep neural networks for reducing the computational load. Finally, the encoder stage comprises four blocks to develop a feature map. In the decoder stage, the original image size is restored by performing several operations, such as up-sampling, 3 × 3 convolution, batch-normalization, and ReLU activation four times. Then, the weight is updated by matching with the ground truth at the final layer. At the classification stage, a 1 × 1 convolution, batch-normalization, and Softmax operations are applied in the last neural network of each block. The suboutputs from four blocks are then matched to four grid images created for multiscale training, corresponding to image size. Then, the weight is updated from each block.

A holdout-type training method was used, and the initial values were all set to those used in [35]. ADAM was used as the optimization function and the learning rate was set to 0.001, beta-1 0.9, and beta-2 0.999 [36]. To finish the training quickly, an early stopping method was used to determine whether the loss value is terminated according to the trend in [37]. Windows 10-based Keras was used as the training environment, and the development PC specifications were i7-7700, 32GB RAM, and NVIDIA-RTX 2080TI.

Regarding the training, binary cross entropy was used as the loss function of the auto-encoder method. In contrast, in the encoder-type method, the loss function is calculated as shown in (5) and (6). In these equations, \( i \) is the pixel position, \( s \) is stage of the suboutput that consists of four blocks, \( N(s) \) is the number of the pixels obtained from the suboutput, and \( y_{s,i} \) is the grid image to match with the ground-truth, the pixel values of which are 0 or 1. Further, \( P(y_{s,i}) \) represents the suboutput values that show the prediction probability at pixel position \( i \) and stage \( s \). Thus, it ranges from 0 to 1 continuously. The binary cross-entropy is used as a function for each suboutput, which are finally summarized.

\[
\text{loss} = \sum_{s=1}^{4} CE(s) 
\]

\[
CE(s) = -\frac{1}{N(s)} \sum_{i=1}^{N(s)} y_{s,i} \log(P(y_{s,i})) + (1 - y_{s,i}) \log(1 - P(y_{s,i}))
\]

For inference, the method sums up the four suboutputs from each block to a single output. Although the sizes of the suboutputs created for each block differ, all of the sizes must be changed to 576 × 576, the same size as the input image. Therefore, we resized the suboutputs by using the linear interpolation method. All of the suboutput channel was two, and the resized suboutputs were added to the corresponding channel to obtain the final result in the shape of 576 × 576 × 2.

The pixel values of this final output image range from 0 to 4 as float numbers; however, these values are normalized for 0 to 1 distribution. Finally, the road-surface damage is considered to exist at a position where the probability value is greater than 0.5 in the second channel.

IV. EXPERIMENTS

A. COMPARISON OF TRAINING RESULTS

Training was performed for comparing the performances of the auto-encoder-and encoder-type methods. As mentioned earlier, the image dataset is categorized to two parts: for training and verification. This section provides the analysis of only the loss value of the verification data for comparison. First, the validation loss values are shown in Fig. 7(a); three deep neural networks were trained in the auto-encoder method. According to the graph, the training process of all the deep neural network models was completed when the epoch reached around 40 and the loss value converged to less than approximately 0.1. The lowest loss value of 0.074 was obtained for ResNet50. Fig. 7(b) shows the validation loss value obtained during the training of three deep neural network models based on the encoder-type method. This method also showed similar results; that is, the training process was

![Validation Loss according to the Networks](image-url)

FIGURE 7. Validation loss graph when using the (a) auto-encoder-type method and (b) encoder-type method.
TABLE 1. Performance evaluation results.

| Index            | Auto-encoder type | Encoder type |
|------------------|-------------------|--------------|
|                  | DenseNet121 | DetNet59 | ResNet50 | Average | DenseNet121 | DetNet59 | ResNet50 | Average | ProposedNet |
| Pixel Accuracy   | 97.834     | 97.435   | 97.555   | 97.607   | 97.472     | 97.304   | 97.290   | 97.355   | 97.902     |
| Mean Accuracy    | 88.356     | 87.293   | 88.073   | 87.907   | 92.803     | 90.425   | 91.489   | 91.572   | 90.580     |
| Mean IoU         | 64.809     | 64.350   | 65.677   | 64.349   | 65.001     | 62.264   | 62.464   | 63.376   | 69.178     |
| Frequency Weighted IoU | 96.628 | 96.616   | 96.700   | 96.648   | 96.611     | 96.715   | 96.710   | 96.745   | 96.958     |

TABLE 2. Computational time results.

| Index      | Auto-encoder type | Encoder type |
|------------|-------------------|--------------|
|            | DenseNet121 | DetNet59 | ResNet50 | Average | DenseNet121 | DetNet59 | ResNet50 | Average | ProposedNet |
| Parameter  | 10.613     | 11.195   | 11.880   | 11.836   | 11.377     | 13.446   | 13.593   | 13.789   | 13.513     |
| Elapsed Time | 40.052 | 38.126   | 36.174   | 38.117   | 35.280     | 33.808   | 31.073   | 33.387   | 33.971     |

B. PERFORMANCE COMPARISON

From among the previous training results, we selected the model with the lowest validation loss for evaluation, and the results are shown in Table 1. First, we compared the performance indexes of all three deep neural network models for each method type. For the auto-encoder-type method, the pixel and mean accuracies showed the highest values when DenseNet121 was used. In addition, the mean IoU and frequency weight IoU were the highest when ResNet50 was used. These results show that the auto-encoder method has varying performance results according to the model used. In contrast, the encoder type method showed the highest performance for both methods.

C. COMPUTATIONAL TIME COMPARISON

Finally, we compare the computational time and parameters of the three models. The experimental results using the validation data are shown in Table 2. The parameters were measured in millions and the elapsed time was measured in milliseconds. The parameters of the auto-encoder-type method include the encoder and decoder stages, and those of the encoder-type method comprise the encoder and classification stages. When the average values of the experiments were compared, the auto-encoder type was found to have high performance in terms of the pixel accuracy and mean IoU. In contrast, the encoder type showed the highest performance in terms of mean accuracy and frequency weight IoU.

D. OUR APPROACH FOR IMPROVEMENT

The previous experimental results clearly verified that the computational time is improved when the encoder type is used. However, the performance index is not significantly improved compared to that of the auto-encoder type. To overcome this drawback, we propose a new deep neural network called ProposedNet. DenseNet121, which has the highest performance index in the encoder type was selected as the basic structure and then modified to improve the performance index, as detailed in Table 3.

ProposedNet is a modified version of DenseNet121 in three aspects. First, the first max pooling operation is removed, and a transition layer and a dense block are added to the rear end. As the size of the input image is reduced by half after max pooling, without this operation, the final layer would be 72 × 72. Instead, the addition of transition layers (3) and a dense block (4) increases the number of reduced features. Second, the growth rate is changed from 32 to 16. This helps improve the computational time. According to the results of the previous experiment, DenseNet121 was noted to be the slowest among the three deep neural network models. To improve its computational time, the size of parameters of DenseNet121 must be reduced. Thus, the parameters of the overall deep neural network must be reduced by changing the growth rate in half.

Third, the parameters used in the dense block are modified, and the 1 × 1 convolution operation is replaced with a 3 × 3 operation. Hence, it is possible to retain the connectivity with nearby pixels for each convolution operation in the dense block. However, the number of iterations in the dense block is changed from (6, 12, 24) to (3, 5, 10, 7) because increasing the size of the kernel slows down the computation speed. Therefore, the effect of increasing the size of the kernel compensates for the iteration reduction, while the nearby pixel connectivity is retained. In addition, the operation used and the computational time decreases by 12.4% on average when using the auto-encoder-type method compared to using the encoder-type method. In addition, the parameters of DenseNet121 are the smallest and the calculation speed of ResNet50 is the fastest for both method types.
TABLE 3. Comparison between the two deep neural network models.

| Layers          | DenseNet121 | ProposedNet |
|-----------------|-------------|-------------|
|                  | Output size | Operational Parameter | Output size | Operational Parameter |
| Convolution     | 244x244     | 7x7 kernel, 2 stride | 244x244 | 7x7 kernel, 2 stride |
| Pooling         | 144x144     | 3x3 max pool, 2 stride | -       | -                   |
| Dense block (1) | 144x144     | 1x1 kernel | 244x244 | 3x3 kernel and 3x3 kernel, 2 stride |
| Transition layers (1) | 72x72 | 2x2 average pool, 2 stride | 144x144 | 2x2 max pool, 2 stride |
| Dense block (2) | 72x72       | [1x1 kernel and 3x3 kernel]x6 | 144x144 | [3x3 kernel and 3x3 kernel]x3 |
| Transition layers (2) | 72x72 | 1x1 kernel | 144x144 | 3x3 kernel |
| Dense block (3) | 36x36       | 2x2 average pool, 2 stride | 72x72 | 2x2 max pool, 2 stride |
| Transition layers (3) | - | - | 36x36 | 2x2 max pool, 2 stride |
| Dense block (4) | -           | -           | 36x36 | [3x3 kernel and 3x3 kernel]x7 |
| growth rate     | 32          | 16          |

in the transition layers is modified in a similar way: the average pool is replaced with max pooling and the $1 \times 1$ convolution operation is changed to a $3 \times 3$ operation. Consequently, only $3 \times 3$ kernels are used in ProposedNet.

E. PERFORMANCE OF OUR APPROACH

The experimental results using ProposedNet are shown in the far-right column of Table 1 and 2. The values of ProposedNet were compared with the most excellent index values from the previous experimental results. DenseNet121 and ResNet50 show the best performance and computation speed for the auto-encoder-type method. Compared to the six highest values of the performance indexes of these two models, the index value of ProposedNet showed a higher performance. In particular, the number of parameters is reduced by 85.7% than those of DenseNet121, and the computation speed is 6.1% faster than that of ResNet50.

V. DISCUSSION

Through this experiment, we conclude that when the structure of the deep neural network used in the encoder-type method is well modified, the algorithm’s recognition performance and computational speed are improved. This could be attributed to two characteristics of the new deep neural network. The first characteristic is that transition layers and dense block are added to the tail of the network by removing the first max pooling that is used for the down sampling. This is to minimize the disadvantage of feature reduction of the original image with decrease in image size. In the case of a small object, such as a crack in the road surface, it is assumed that the smaller the size of the image, the faster the loss of its spatial feature. In the object recognition for the APMV, a large object, such as cars or buses, usually occupies most of the area in the image. There is a high possibility that the unique feature of the object is extracted even if the size of the image is reduced. As the area occupied by cracks on the road surface is relatively small, the deep neural network model has an advantage when the transition layers are positioned at the tail. Therefore, such feature information affects recognition performance improvement.

The second characteristic is that in the ProposedNet, a $3 \times 3$ convolution is applied to all neural networks instead of a $1 \times 1$ convolution. Bottleneck neural networks with $1 \times 1$ convolution operation are generally utilized for increasing the operation speed. In ProposedNet, the $3 \times 3$ convolution is used to extract the features highly related to the neighboring pixels to improve the accuracy. Owing to this modification, the computational speed is slow. To compensate for this disadvantage, the growth rate and the number of iterations are reduced to make the deep neural network lightweight and to ensure rapid inference. In conclusion, these two characteristics enable the accuracy and computational speed to be improved simultaneously.

We implemented a deep neural network algorithm that recognizes road-surface damage. Based on this algorithm, in future research, a deep neural network fused with an object recognition algorithm that is capable of recognizing cars, buses, motorcycles, and people, which can be found on roads, is required. Currently, research for recognizing and avoiding obstacles is actively being studied in the field of autonomous driving. These technologies are expected to be applied to the personal mobility field soon, which has recently gained considerable attention. In particular, it is anticipated that a safer APMV would be designed in the future if the obstacle algorithm works with road-surface-damage recognition simultaneously.

Moreover, various deep learning methods have been introduced to improve recognition performance. In the case of semi-supervised learning [38], a method was proposed to overcome the difficulty of producing labeled images. Papandreou et al. used a method to increase the performance of segmentation by labeling only the presence or absence of cracks in the input image. Next, adversarial learning increases the recognition performance by generating a virtual labeled image using a generative adversarial network [39]. These studies have been developed based on most auto-encoder types [40], [41], and they are expected to be able to create a faster and more accurate algorithm when fused with the proposed encoder type. Finally, it can integrate with reinforcement learning, which is actively researched in the field.
of intelligence transportation system [42]. It is also expected that this technology can be applied to the road-surface damage recognition algorithm as it has various ways modification methods for segmentation [43], [44].

**VI. CONCLUSION**

In this paper, we proposed a new deep neural network algorithm for road-surface-damage object recognition. To develop this algorithm, the detection strategy was changed to semantic segmentation. This is because when the strategy is in the form of bounding boxes, the accurate extraction of the damaged area of the road surface is difficult because the image contains other unnecessary areas that are larger than the damaged area. To improve this drawback, in this study, a new image dataset was created for segmentation with 1,650 image sets for training and validation. Then, a new semantic segmentation algorithm was developed based on these image sets. The contribution of this study includes the proposal of a new type of semantic segmentation. Until now, the basis of the algorithm was auto-encoding; however, the auto-encoder-type
method has a slow computational speed. To overcome this disadvantage, we proposed an encoder-type semantic segmentation and modification strategy to enhance the accuracy and computational speed. The strategy was evaluated with respect to four performance indexes to determine its accuracy compared to the existing method. Furthermore, we measured the number of parameters used and elapsed time to determine its speed. The results showed that the ProposedNet has higher performance index values than other deep neural networks with the auto-encoder-type strategy. Moreover, the number of parameters was significantly reduced and the operational speed was improved. Finally, we applied the developed algorithm to various images, and the results are as shown in Table 4. This is the result of the experiments on images that show general road-surface damage. In the road image, the ground truth, which is a damaged area in the images, was generated and the results were compared using four algorithms. Three deep neural network models were utilized with the auto-encoder type, and the other was the encoder type using the deep neural network model proposed herein. As indicated, the results with respect to the encoder type are more similar to the ground truth than those of the auto-encoder type.

REFERENCES

[1] J. Nakane and M. Faroofaeg, “Elder care in Japan,” Perspect. (Gerontological Nursing Assoc. Canada), vol. 28, no. 1, pp. 17–24, 2004.
[2] T. L. Lawrence, “Health care facilities for the elderly in Japan,” Int. J. Health Services, vol. 15, no. 4, pp. 677–697, Oct. 1985.
[3] A. Argyros, P. Georgiadis, P. Trahanias, and D. Tsaikris, “Semantic autonomous navigation of a robotic wheelchair,” J. Intell. Robotic Syst., vol. 34, no. 3, pp. 315–329, Jul. 2002.
[4] Y. Kobayashi, Y. Kinpara, T. Shibasawa, and Y. Kuno, “Robotic wheelchair based on observations of people using integrated sensors,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., St. Louis, MS, USA, Oct. 2009, pp. 2013–2018.
[5] C. Ilas, “Electronic sensing technologies for autonomous ground vehicles: A review,” in Proc. 8th Int. Symp. Adv. Topics Electr. Eng. (ATEEE), Bucharest, Romania, May 2013, pp. 1–6.
[6] T. Kim and S. K. Ry, “Review and analysis of pothole detection methods,” J. Emerg. Trends Inf. Sci. Eng., vol. 5, no. 8, pp. 603–608, Aug. 2014.
[7] S.-K. Ryu, T. Kim, and Y.-R. Kim, “Feature-based pothole detection in two-dimensional images,” Transp. Res. Rec., J. Transp. Res. Board, vol. 2528, no. 1, pp. 9–17, Jan. 2015.
[8] G. Li, J. Wan, S. He, Q. Liu, and B. Ma, “Supervised-segmentation semantic segmentation using adversarial learning for pavement crack detection,” IEEE Access, vol. 8, pp. 51446–51459, 2020.
[9] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, and X. Yang, “Automatic pixel-level crack detection and measurement using fully convolutional network,” Comput.-Aided Civil Infrastruct. Eng., vol. 33, no. 12, pp. 1090–1109, Dec. 2018.
[10] A. Akagic, E. Buza, S. Omanovic, and A. Karabegovic, “Pavement crack detection using osu thresholding for image segmentation,” in Proc. 41st Int. Conf. Inv. Commun. Technol., Electron. Microelectron. (MIPRO), May 2018, pp. 1092–1097.
[11] H. Li, D. Song, Y. Liu, and B. Li, “Automatic pavement crack detection by multi-scale image fusion,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 6, pp. 2025–2036, Jun. 2019.
[12] R. Amhaz, S. Chambon, J. Idier, and V. Baltazar, “Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection,” IEEE Trans. Intell. Transp. Syst., vol. 17, no. 10, pp. 2718–2729, Oct. 2016.
[13] 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.
[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. C. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in Proc. 14th Eur. Conf. Comput. Vis. (ECCV), Amsterdam, The Netherlands, Oct. 2016, pp. 21–37.
[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 779–788.
[16] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), Montreal, QC, Canada, Dec. 2015, pp. 91–99.
[17] J. Li, X. Zhao, and H. Li, “Method for detecting road pavement damage based on deep learning,” in Proc. 13th Health Monitor. Structural Biol. Syst., Apr. 2019, Art. no. 109722D.
[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017, arXiv:1704.04861. [Online]. Available: http://arxiv.org/abs/1704.04861
[19] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, “Road damage detection and classification using deep neural networks with smartphone images,” Comput.-Aided Civ. Infrastruct. Eng., vol. 33, no. 12, pp. 1127–1141, Dec. 2018.
[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 2818–2826.
[21] W. Cao, Q. Liu, and Z. He, “Review of pavement defect detection methods,” IEEE Access, vol. 8, pp. 14531–14544, 2020.
[22] Z. Fan, Y. Wu, J. Lu, and W. Li, “Automatic pavement crack detection based on structured prediction with the convolutional neural network,” 2018, arXiv:1802.02208. [Online]. Available: http://arxiv.org/abs/1802.02208
[23] M. David Jenkins, T. A. Carr, M. I. Iglesias, T. Buggy, and G. Morison, “A deep convolutional neural network for semantic pixel-wise segmentation of road and pavement surface cracks,” in Proc. 26th Eur. Signal Process. Conf. (EUSIPCO), Rome, Italy, Sep. 2018, pp. 2120–2124.
[24] 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., Munich, Germany, Oct. 2015, pp. 234–241.
[25] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, “DeepCrack: Learning hierarchical convolutional features for crack detection,” IEEE Trans. Image Process., vol. 28, no. 3, pp. 1498–1512, Mar. 2019.
[26] S. Bang, S. Park, H. Kim, and H. Kim, “Encoder-decoder network for pixel-level road crack detection in black-box images,” Comput.-Aided Civ. Infrastruct. Eng., vol. 34, no. 8, pp. 713–727, Feb. 2019.
[27] C. Chen and S.-K. Ryu, “Road surface damage detection using fully convolutional neural networks and semi-supervised learning,” Sensors, vol. 19, no. 24, p. 5501, Dec. 2019.
[28] Y. Jo and S. Ryu, “Pothole detection system using a black-box camera,” Sensors, vol. 15, no. 11, pp. 29316–29331, Nov. 2015.
[29] M. Eisenbach, R. Stricker, D. Seichter, K. Amendt, K. Debes, M. Sesselmann, D. Ebersbach, U. Stoeckert, and H.-M. Gross, “How to get pavement distress detection ready for deep learning? A systematic approach,” in Proc. Int. Joint Conf. Neural Networks (IJCNN), May 2017, pp. 2039–2047.
[30] M. Marszalek and C. Schmid, “Accurate object localization with shape masks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Minneapolis, MN, USA, Jun. 2007, pp. 1–8.
[31] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Boston, MA, USA, Jun. 2015, pp. 3431–3440.
[32] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, “DenseNet: Implementing efficient ConvNet descriptor pyramids,” 2014, arXiv:1404.1869. [Online]. Available: http://arxiv.org/abs/1404.1869
[33] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[34] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, “DetNet: A backbone network for object detection,” 2018, arXiv:1804.06215. [Online]. Available: http://arxiv.org/abs/1804.06215
[35] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proc. Int. Conf. Artif. Intell. Statist., Sardinia, Italy, May 2010, pp. 249–256.

[36] D. P. Kingma and J. L. Ba, “ADAM: A method for stochastic optimization,” in Proc. 3rd Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, 2015, pp. 1–15.

[37] L. Prechelt, “Early stopping—but when?” Neural Netw., vol. 11, no. 4, pp. 761–767, 1998.

[38] D. P. Kingma and J. L. Ba, “ADAM: A method for stochastic optimization,” in Proc. 3rd Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, 2015, pp. 1–15.

[39] L. Prechelt, “Early stopping—but when?” Neural Netw., vol. 11, no. 4, pp. 761–767, 1998.

[40] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, “Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Santiago, Chile, Dec. 2015, pp. 1742–1750.

[41] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst. 27 (NIPS), Montréal, QC, Canada, 2014, pp. 2672–2680.

[42] Z. Ning, Y. Li, P. Dong, X. Wang, M. S. Obaidat, X. Hu, L. Guo, Y. Guo, J. Huang, and B. Hu, “When deep reinforcement learning meets 5G-enabled vehicular networks: A distributed offloading framework for traffic big data,” IEEE Trans. Ind. Informat., vol. 16, no. 2, pp. 1352–1361, Feb. 2020.

[43] Z. Ning, R. Y. K. Kwok, K. Zhang, X. Wang, M. S. Obaidat, L. Guo, X. Hu, B. Hu, Y. Guo, and B. Sadoun, “Joint computing and caching in 5G-envisioned Internet of vehicles: A deep reinforcement learning-based traffic control system,” IEEE Trans. Intell. Transp. Syst., early access, Feb. 5, 2020, doi: 10.1109/TITS.2020.2970276.

[44] K. M. Lee, H. Myeong, and G. Song, “SeedNet: Automatic seed generation with deep reinforcement learning for robust interactive segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 1760–1768.

[45] Y. Zhou, X. Sun, Z.-J. Zha, and W. Zeng, “Context-reinforced semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Long Beach, CA, USA, Jun. 2019, pp. 4046–4055.

SEUNGBO SHIM was born in Donghae, South Korea, in 1982. He received the B.S. degree in electrical engineering from Pusan National University, Pusan, South Korea, in 2009, and the M.S. degree in electronic and electrical engineering from the Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2011. He is currently pursuing the Ph. D. degree with the Korea Advanced Institute of Science and Technology.

From 2010 to 2017, he was a Research Engineer with Samsung Heavy Industries. Since 2017, he has been a Research Specialist with the Korea Institute of Civil Engineering and Building Technology. His current interests include machine vision, deep learning, automatic measurement systems, sensor networks, and infrastructure maintenance.

GYE-CHUN CHO was born in South Korea, in 1970. He received the B.S. degree in civil engineering and the M.S. degree in geotechnical engineering from Korea University, Seoul, South Korea, in 1994 and 1996, respectively, and the Ph.D. degree in geotechnical engineering from the Georgia Institute of Technology, GA, USA, in 2001. He was an Assistant Professor with the Pennsylvania State University, PA, USA, from 2001 to 2002, and an Assistant Professor with the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, from 2005 to 2012, and a Visiting Professor with the Georgia Institute of Technology, from 2009 to 2010. Since 2012, he has been a Professor with the Department of Civil and Environmental Engineering, KAIST. His current research interests include infrastructure maintenance, tunnel excavation, biopolymers, and soil behavior analysis.