Semantic Segmentation of Degraded Images Using Layer-Wise Feature Adjustor

Kazuki Endo
Teikyo Heisei University
Nakano-ku, Tokyo, Japan
k.endo@thu.ac.jp

Masayuki Tanaka
Tokyo Institute of Technology
Meguro-ku, Tokyo, Japan
mtanaka@sc.e.titech.ac.jp

Masatoshi Okutomi
Tokyo Institute of Technology
Meguro-ku, Tokyo, Japan
mxo@sc.e.titech.ac.jp

Abstract

Semantic segmentation of degraded images is important for practical applications such as autonomous driving and surveillance systems. The degradation level, which represents the strength of degradation, is usually unknown in practice. Therefore, the semantic segmentation algorithm needs to take account of various levels of degradation. In this paper, we propose a convolutional neural network of semantic segmentation which can cope with various levels of degradation. The proposed network is based on the knowledge distillation from a source network trained with only clean images. More concretely, the proposed network is trained to acquire multi-layer features keeping consistency with the source network, while adjusting for various levels of degradation. The effectiveness of the proposed method is confirmed for different types of degradations: JPEG distortion, Gaussian blur and salt&pepper noise. The experimental comparisons validate that the proposed network outperforms existing networks for semantic segmentation of degraded images with various degradation levels.

1. Introduction

Semantic segmentation has been remarkably progressed by using a convolutional neural network (CNN) for the last decade [23, 17, 30, 24, 2, 6, 5, 31]. Many studies of CNN-based semantic segmentation have focused on only clean images without any image degradation. However, digital images usually include some degradations like distortion, noise and blurring. For example, people generally prefer JPEG compressed images to raw digital images because they want to reduce the data size of their images. Moreover, semantic segmentation of degraded images is practically important for autonomous driving, surveillance systems, etc. In a practical semantic segmentation, it is important to take into account image degradations. The quality of a degraded image is strongly depended on a degradation level like the JPEG quality factor for JPEG distortion, the standard deviation for Gaussian blur kernel, etc. The degradation level is usually unknown for semantic segmentation algorithms. Therefore, semantic segmentation of degraded images has to cope with various unknown levels of degradation. This paper aims to construct a CNN-based semantic segmentation network for degraded images under a known degradation type with an unknown degradation level.

A very naive approach for the semantic segmentation of degraded images is to feed degraded images into a network trained with clean images. This approach shows poor performance because the network trained with only clean images does not have enough knowledge of degraded images. Then, we can consider training the network with clean and degraded images. Although the network trained with clean and degraded images can improve the performance for degraded images, the network shows lower performance for clean images than a network trained with clean images only. Endo *et al.* [8, 7, 9] have reported the same phenomenon in the classification task of degraded images. Guo *et al.* [12] have proposed a Dense-Gram-Network (DGN) based on knowledge distillation [22, 13] for the semantic segmentation of degraded images. Guo *et al.* verified their approach under a known degradation type with a known degradation level. We propose a semantic segmentation network by following a feature adjustor for the classification network [10]. Note that Guo *et al.* focus on only one degradation level, while we take account of various degradation levels.

Our contributions are the following three points. 1) The proposed semantic segmentation network is able to deal with degraded images over various levels of degradation without sacrificing the performance of clean images. 2) This paper combines the feature adjustor [10] and the layer-wise knowledge distillation [1] to be consistent with not only the final features of a degraded image but also its multi-layer features. 3) The proposed method was confirmed for several kinds of degradation with two famous datasets.

2. Related works

There are several studies for the recognition of degraded images. However, most of those studies mainly focus on classification [4, 21, 19, 8, 11, 20, 7, 18, 28, 9, 10]. There

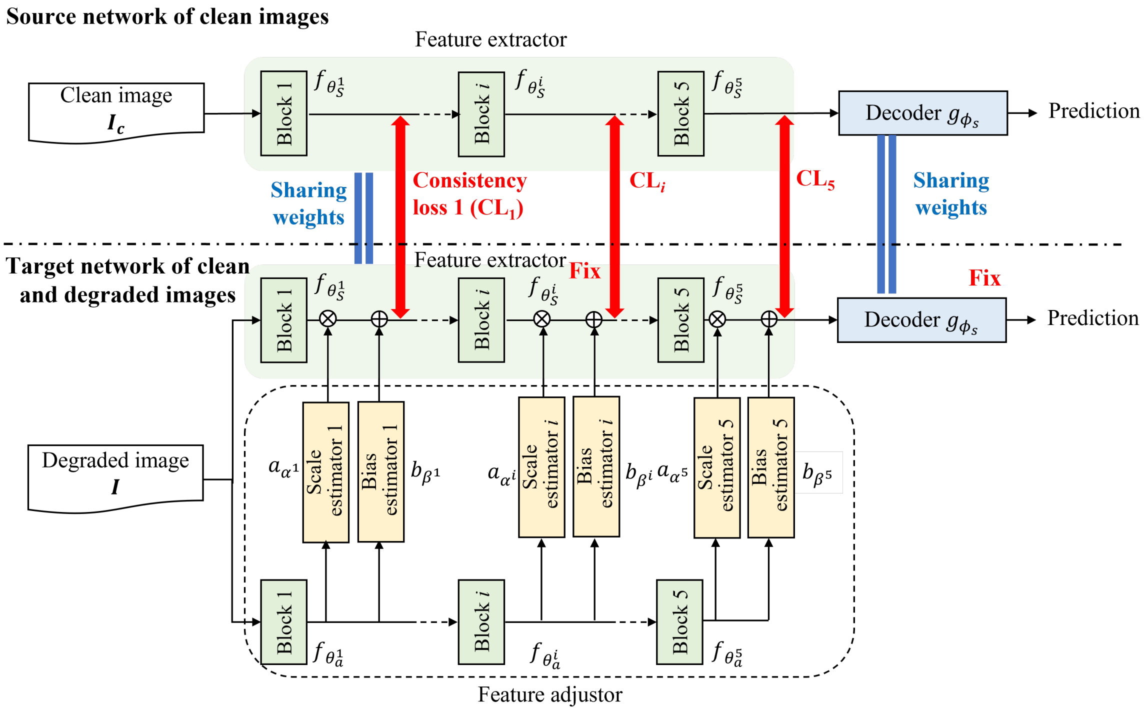

Figure 1. Layer-wise feature adjustor (proposed framework). A source segmentation network is trained with clean images. A target segmentation network is trained with degraded images to be consistent with the features of clean images extracted from the source network. Degraded images have various degradation levels (including no degradation, i.e. clean image). In this paper, SegNet [2] is used as the source network.

are few studies for the semantic segmentation of degraded images. Regarding the classification of degraded images, Peng *et al.* [21] have compared training strategies from the usage of training data: using high-resolution data before using low-resolution data, using low-resolution data before using high-resolution data and mixing low-resolution and high-resolution data. The last strategy is called "mixed training". This paper trains segmentation networks of degraded images by using mixed training. Pei *et al.* [18] and Endo *et al.* [10] have proposed methods based on consistency regularization for the features of clean images extracted from a source network which is trained with clean images only. Pei *et al.* mainly focused on the classification of degraded images only and did not pay much attention to the classification of clean images. On the other hand, Endo *et al.* [10] have proposed a network to classify degraded images over various levels of degradation without sacrificing the classification performance of clean images. They have adjusted the features of degraded images from the final layer of the feature extractor to fit the features of clean images. This paper extends the method proposed by Endo *et al.* [10] to the semantic segmentation of degraded images and the regularization of multi-layer image features.

Regarding the semantic segmentation of degraded images, Guo *et al.* [12] have proposed DGN based on knowledge distillation, where a source network is a segmentation network trained with clean images. Though their proposed network was trained with clean images and degraded images, the degradation level of degraded images could not take multiple values but only a single value. Our proposed framework is also based on knowledge distillation but can deal with various levels of degradation while keeping the segmentation performance of clean images.

This paper uses SegNet proposed in Badrinarayanan *et al.* [2] as a semantic segmentation network. SegNet has an encoder-decoder architecture where the encoder is based on VGG16 [26].

3. Proposed method

3.1. Proposed framework

Figure 1 shows our proposed framework which we call a "layer-wise feature adjustor." The proposed framework is based on layer-wise knowledge distillation [1]. A source segmentation network gives its knowledge to a target segmentation network. This paper uses SegNet [2] as the base segmentation network. However, our proposed framework can be based on the other segmentation networks. The

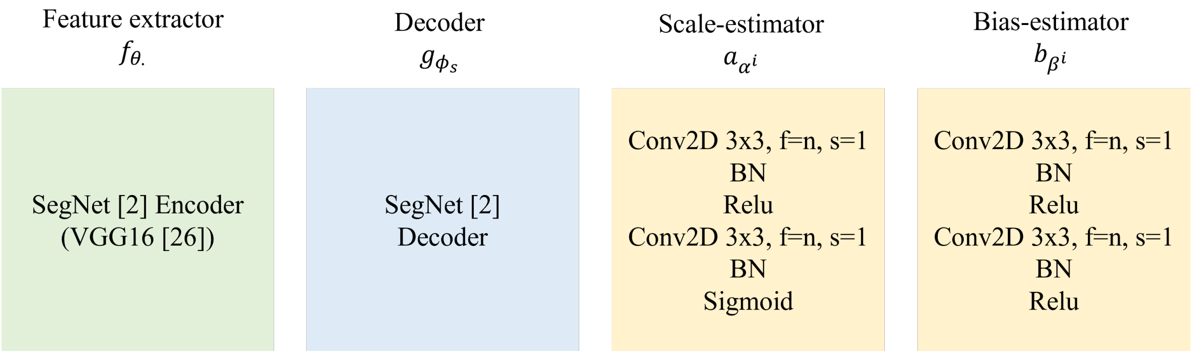

Figure 2. Network structure. The encoder of SegNet is the same structure as the feature extractor of VGG16 [26]. "BN" and "s" represent a batch normalization and the size of stride, respectively. Regarding α^i and β^i, the dimension of feature is $n = 64, 128, 256, 512, 512$ for $i = 1, 2, 3, 4, 5$, respectively.

source network is divided into a feature extractor (encoder) and a decoder. The feature extractor, which is denoted by $\left\{f_{\theta_s^i}\right\}_{i=1,2,..,5}$, produces image features of clean images, and the decoder g_{ϕ_s} outputs predictions of segmentation labels. The source network is trained with clean images only. Now, i-th block of source features x_i, which denotes the output of $f_{\theta_s^i}$, is defined by

$$x_i \overset{def}{=} f_{\theta_s^i}\left(x_{i-1}\right), \quad x_0 = I_c, \quad i = 1, 2, ..., 5, \tag{1}$$

where I_c denotes clean images. On the other hand, the target network has two streams: a feature extractor, whose weights are fixed and shared with the source network, and a trainable feature adjustor. The feature adjustor is composed of $\left\{f_{\theta_a^i}\right\}$, $\left\{a_{\alpha^i}\right\}$ and $\left\{b_{\beta^i}\right\}$ for $i = 1, 2, ..., 5$. $\left\{f_{\theta_a^i}\right\}_{i=1,2,..,5}$ is the same structure as the feature extractor of the source network but is trainable. The output of $f_{\theta_a^i}$ is fed into a scale estimator a_{α^i} and a bias estimator b_{β^i} for each i-th block of features. The scale and bias estimators infer parameters which adjust image features generated by $f_{\theta_s^i}$. Now, $\left\{F^i\right\}_{i=1,2,...,5}$ is defined by

$$F^i\left(y_{i-1}, z_i\right) \overset{def}{=} f_{\theta_s^i}\left(y_{i-1}\right) \otimes a_{\alpha^i}\left(z_i\right) + b_{\beta^i}\left(z_i\right), \tag{2}$$

$$y_i \overset{def}{=} F^i\left(y_{i-1}, z_i\right), \quad y_0 = I, \tag{3}$$

$$z_i \overset{def}{=} f_{\theta_a^i}\left(z_{i-1}\right), \quad z_0 = I, \tag{4}$$

for $i = 1, 2, ..., 5$ where I denotes degraded images with various levels of degradation (including no degradation, i.e. clean images). $\otimes$ denotes an element-wise product. $F^5\left(y_4, z_5\right)$ is fed into the decoder. The prediction of the target network is calculated by

$$g_{\phi_s}\left(F^5\left(y_4, z_5\right)\right). \tag{5}$$

To train the target network, the following total loss function is defined by

$$\mathbf{E}\left[\sum_{i=1}^{5} w_{CL}^i \cdot CL\left(x_i, y_i | \theta_a^i, \alpha^i, \beta^i\right)\right], \tag{6}$$

$$w_{CL}^i = 0.2, \tag{7}$$

where CL denotes a consistency loss function. This paper uses the cosine similarity for a CL function followed by Endo *et al.* [10]. This paper does not consider the loss function of g_{ϕ_s} in Eq. (6) and does not train the weights of g_{ϕ_s} as followed by Endo *et al.* [10]. In the optimization procedure, an expectation operator **E** is replaced by the sample mean of the mini-batch.

Figure 2 shows a concrete network structure. A feature extractor and a decoder are based on SegNet. The feature extractor of SegNet has the same structure as the feature extractor of VGG16 [26]. A scale-estimator and a bias-estimator are composed of convolution layers, batch normalization layers and activation functions. The dimension of features are $n = 64, 128, 256, 512, 512$ for $i = 1, 2, 3, 4, 5$, respectively.

3.2. Training procedure

Here, we describe the training procedure for the source network and the target network of the proposed framework as shown in Fig. 1. First, the source network is trained with only clean images. All the parameters of the trained source network are fixed when the target network is trained. Then, the target network is trained by "mixed training" [21] which uses degraded images with various levels of degradation including clean images. Degraded images are easily generated from clean images with a degradation operator, where degradation levels have to be input into the degradation operator. The degradation levels are randomly sampled from a uniform distribution. A clean image and its degraded image

Table 1. Training conditions

Name	Clean	Degrade	Proposed
Network structure	Source only	Source only	Layer-wise feature adjustor
Training data	Clean images	Clean images and degraded images	

are fed into the source network and the target network, respectively. After that, the loss function, defined in Eq. (6), is minimized.

4. Experimental validations

In this section, our proposed method is confirmed by using two datasets: CamVid dataset [3] and SUN RGB-D dataset [27]. CamVid is a road scene dataset applicable for autonomous driving. Although CamVid is originally the road scene video data, we only use the static image data of 11 categories except for unlabelled pixels. The number of training data, validation data and test data are 367, 101 and 237, respectively. The image size of each data is 480×360. SUN RGB-D is an indoor scene dataset which acquires RGB-D images from NYU depth v2 [25], Berkeley B3DO [14] and SUN3D [29]. The images of SUN RGB-D include 37 categories except for unlabelled pixels. The number of training images and test images are 5285 and 5050, respectively. This paper uses RGB images only though SUN RGB-D contains depth data. Each image was resized into 480×360 before experiments. We used the SegNet without dropout for CamVid images and the SegNet with dropout for SUN RGB-D images. Pytorch was used for all the implementation of our experiments.[1]

This section is organized as follows. First, the training conditions of experimental validations are described in 4.1. Second, we focus on the JPEG distortion and confirm the validity of the proposed framework by using CamVid and SUN RGB-D datasets in 4.2 because JPEG is the de-facto standard for the compression of digital images. Then, the effectiveness of the proposed framework for other degradations is confirmed by using CamVid in 4.3. Finally, the superiority of a layer-wise feature adjustor is confirmed by comparing it with an original feature adjustor [10] in 4.4.

4.1. Training conditions

To confirm the proposed layer-wise feature adjustor, we compare three networks trained with different conditions, as seen in Table 1. “Clean” is a segmentation network trained with clean images only and is also a source network for our proposed target network. “Degrade” is a segmentation network trained with both clean and degraded images, where its network structure is the same as the source network. “Proposed” is the proposed target network trained with clean and degraded images, where the source network is “Clean”. In the above training, degraded images have five degradation levels for each degradation. The degradation levels, including clean images, are uniformly distributed and randomly sampled. Median frequency balancing weight [6] was used to train the three networks because the frequencies of appearance are quite different among categories. The median frequency balancing weight was calculated by using training data. We applied five data augmentations for CamVid images: horizontal flip, random translation, random brightness, random saturation and random scaling. In the case of SUN RGB-D images, randomly changing the hue of images was added to the five augmentations. All data augmentations were applied after applying a degradation operator to clean images. Due to the random scaling, the degradation level of an input image is different from one of a degraded image. Thus, an original degradation level, which was input into the degradation operator, was modified into some proper degradation level as described in 4.4. RAdam [16] optimizer was used for almost training with the initial learning rate 10^{-3} and the weight decay 10^{-4}.[2] The model for test data was selected to maximize the global accuracy [24] of validation data for CamVid and test data for SUN RGB-D. The recognition performance of the three networks is evaluated by using the mean intersection over union (mIoU) [24]. The mIoU is widely used as the performance metric of semantic segmentation.

Table 2. mIoU for JPEG distorted CamVid images. “Average” denotes the mean of mIoUs over five degradation levels and clean images. The degradation level means the JPEG quality factor.

Degradation level	Clean	Degrade	Proposed
Clean images	**0.575**	0.543	**0.575**
90	0.572	0.543	**0.574**
70	0.567	0.541	**0.573**
50	0.563	0.539	**0.572**
30	0.545	0.534	**0.566**
10	0.460	0.505	**0.536**
Average	0.547	0.534	**0.566**

4.2. Validation with JPEG distortion

Here, the proposed method is validated for JPEG distortion.[3] JPEG quality factors are used as the degradation levels of JPEG distortion in this paper. This paper used five

[1] You can get the reproduction code of experiments from our website. (http://www.ok.sc.e.titech.ac.jp/res/CNNIR/IRDI/)

[2] Adam [15] optimizer was used to train only the source network of clean images for SUN RGB-D with an initial learning rate of 10^{-4} and an weight decay of 10^{-4}, where the learning rate was multiplied by 0.99 on each epoch.

[3] We used Python Image Library (PIL) for JPEG compression.

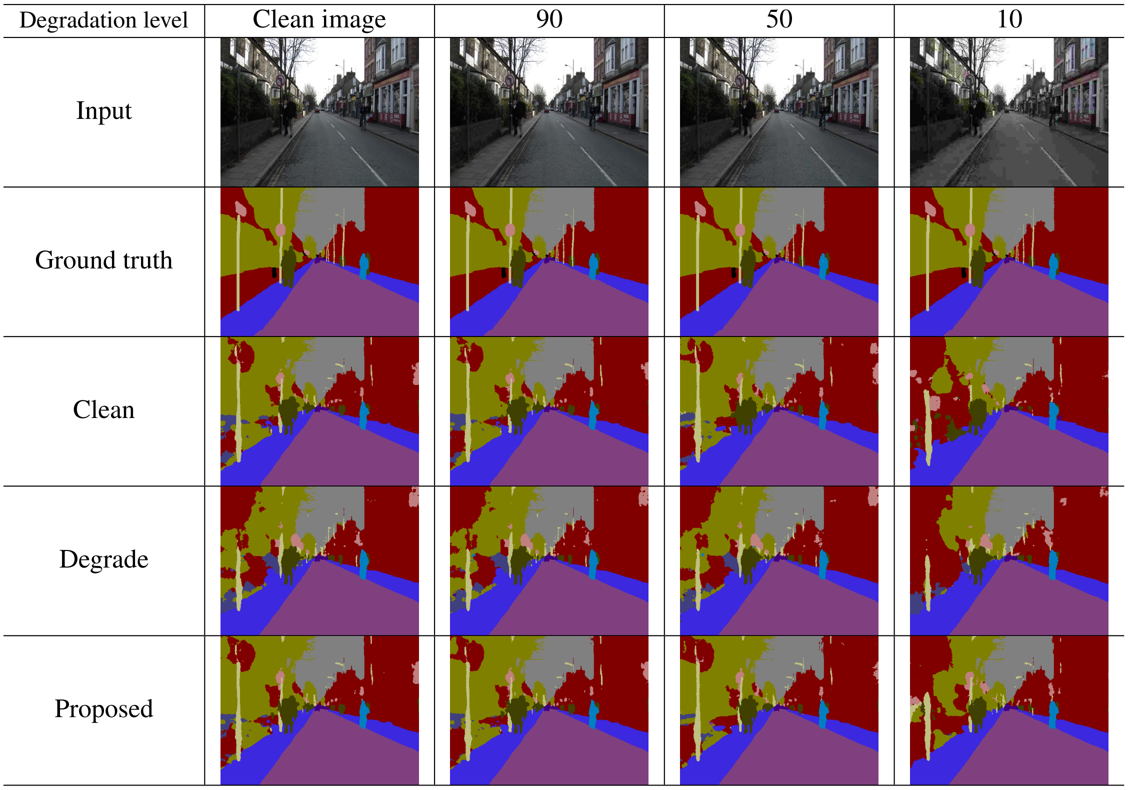

Figure 3. Estimated results for test samples of JPEG distorted CamVid. The degradation level means the JPEG quality factor.

degradation levels for the experiments: 10, 30, 50, 70 and 90.

Table 2 shows the mIoU of several degradation levels and clean images for CamVid images. The mIoU of a source network is 0.575 for clean images and higher than the number of 0.567 based on FCN8s reported by Guo *et al.* [12]. Although it is not state-of-the-art, it is enough performance for our validations. The mIoU of "Clean" roughly shows good performance but significantly drops for the degradation level 10. "Degrade" shows better performance for the degradation level 10 but worse than "Clean" for the other degradation levels. In other words, "Degrade" was averagely trained over all the degradation levels. This phenomenon has been reported for the classification of degraded images in Endo *et al.* [10]. "Proposed" shows the best performance of the three networks as shown in Table 2. Especially for clean images, "Proposed" shows the same mIoU as "Clean". Figure 3 shows estimated results for test samples of JPEG distorted CamVid images. "Proposed" looks better estimation than other networks even if the degradation level is high or low. That is, the proposed method is effective for the semantic segmentation of JPEG distortion over various degradation levels.

Table 3 and Fig. 4 show the mIoU and estimated images of test samples for JPEG distorted SUN RGB-D images, respectively. Badrinarayanan *et al.* [2] reported that mIoU was between 0.225 and 0.321 in different training conditions by using SegNet. On the other hand, the mIoU of a source network is 0.247 for clean images and is inside the above interval. The source network does not show state-of-the-art performance, but it is enough for the following validations. "Proposed" shows the best performance of the three networks for all degradation levels, as seen in Table 3. Figure 4 shows that the prediction of "Proposed" is improved against "Clean" for the degradation level of 10. Table 3 and Fig. 4 show almost the same tendency as described in the case of CamVid images. Thus, the proposed method is effective for JPEG distortion over various levels

Table 3. mIoU for JPEG distorted SUN RGB-D images. "Average" denotes the mean of mIoUs over five degradation levels and clean images. The degradation level means the JPEG quality factor.

Degradation level	Clean	Degrade	Proposed
Clean images	0.247	0.214	**0.250**
90	0.246	0.214	**0.249**
70	0.245	0.214	**0.248**
50	0.241	0.213	**0.246**
30	0.233	0.212	**0.245**
10	0.175	0.200	**0.229**
Average	0.231	0.211	**0.244**

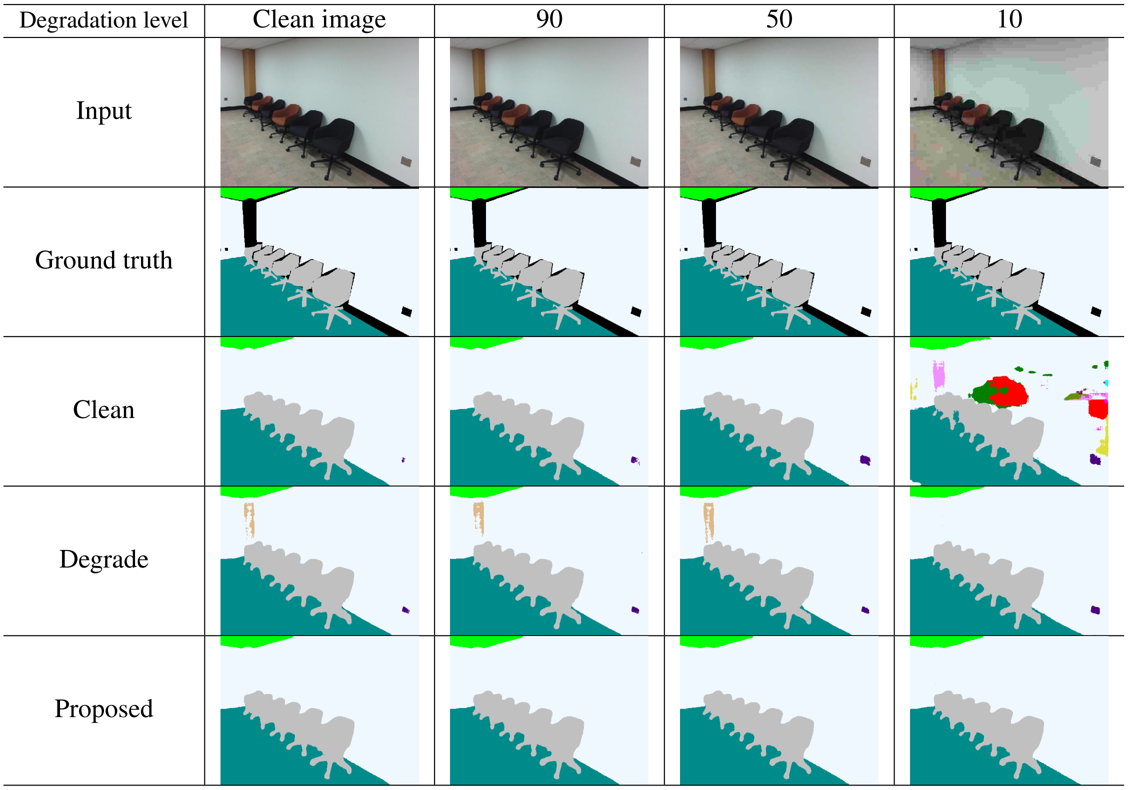

Figure 4. Estimated results for test samples of JPEG distorted SUN RGB-D. The degradation level means the JPEG quality factor.

Table 4. mIoU for Gaussian blurring CamVid images. "Average" means the mean of mIoUs over five degradation levels and clean images. The degradation level means the standard deviation of Gaussian blur kernel.

Degradation level	Clean	Degrade	DGN [12] (FCN8s)	Proposed
Clean images	**0.575**	0.477	0.567	**0.575**
1	0.533	0.475	0.543	**0.560**
2	0.401	0.471	0.521	**0.550**
3	0.266	0.464	0.498	**0.541**
4	0.189	0.456	0.487	**0.527**
5	0.144	0.449	0.485	**0.507**
Average	0.351	0.465	0.517	**0.543**

Table 5. mIoU for CamVid images applied to salt&pepper noise. "Average" means the mean of mIoUs over five degradation levels and clean images. Unfortunately, we could find only two mIoUs for clean images and the degradation level 0.1 for DGN (FCN8s). The degradation level means the density of salt&pepper noise.

Degradation level	Clean	Degrade	DGN [12] (FCN8s)	Proposed
Clean images	**0.575**	0.513	0.567	0.573
0.05	0.213	0.511	-	**0.570**
0.1	0.076	0.508	0.504	**0.567**
0.15	0.037	0.504	-	**0.562**
0.2	0.026	0.498	-	**0.559**
0.25	0.021	0.491	-	**0.554**
Average	0.158	0.504	-	**0.564**

of degradations.

4.3. Validation with other degradations

In this section, two types of degradations are analyzed here: Gaussian blur and salt&pepper noise. In addition to the three networks as shown in Table 1, we also compare the performance results of DGN reported by Guo *et al.* [12]. Although Guo *et al.* have applied DGN to several networks of semantic segmentation, this paper uses their numbers based on FCN8s [24], denoted by DGN (FCN8s), as references. As both SegNet and FCN8s have the feature extractor based on VGG16, we use DGN (FCN8s) in this paper. Guo *et al.* [12] have reported all the numbers using DGN (FCN8s) trained with clean images and unique degradation level.

Table 4 shows mIoU for Gaussian blurring CamVid images, where the degradation level is a standard deviation of Gaussian blur kernel and takes five numbers: 1, 2, ..., 5.

Table 6. mIoUs with single-layer and layer-wise feature adjustors for degraded CamVid images. "Average" denotes the mean of mIoUs over five degradation levels and clean images. In each degradation, the degradation level means the quality factor of JPEG, the standard deviation of Gaussian blur kernel and the density of salt&pepper noise, respectively.

JPEG			Gaussian blur			Salt&pepper noise		
Degradation level	Single-layer	Proposed (layer-wise)	Degradation level	Single-layer	Proposed (layer-wise)	Degradation level	Single-layer	Proposed (layer-wise)
Clean images	**0.575**	**0.575**	Clean images	**0.575**	**0.575**	Clean images	0.571	**0.573**
90	0.572	**0.574**	1	0.554	**0.560**	0.05	0.440	**0.570**
70	0.569	**0.573**	2	0.495	**0.550**	0.1	0.374	**0.567**
50	0.565	**0.572**	3	0.455	**0.541**	0.15	0.326	**0.562**
30	0.552	**0.566**	4	0.418	**0.527**	0.2	0.292	**0.559**
10	0.506	**0.536**	5	0.385	**0.507**	0.25	0.267	**0.554**
Average	0.557	**0.566**	Average	0.480	**0.543**	Average	0.378	**0.564**

Comparing "Clean" with "Proposed", the mIoU of "Proposed" is the same mIoU as "Clean" for clean images. Moreover, "Proposed" shows the best performance of the four networks for all degradation levels. That is, the proposed method is effective for Gaussian blur.

Table 5 shows mIoU for CamVid images applied to salt&pepper noise, where the degradation level is a density of salt&pepper noise and takes five values: 0.05, 0.1, 0.15, 0.2, and 0.25. Unfortunately, we could find only two mIoUs for clean images and the degradation level of 0.1 from Guo *et al.* [12]. Comparing "Clean" with "Proposed", the mIoU of "Proposed" is 0.002 lower than "Clean" for clean images. However, "Proposed" shows the best performance of the four networks for all degradation levels except for only clean images. The result shows almost the same tendency as the Gaussian blur mentioned above. That is, the proposed method is also effective for salt&pepper noise.

Therefore, the proposed method is effective for not only JPEG but also other degradations.

4.4. Comparison with single-layer feature adjustor

An original feature adjustor, proposed by Endo *et al.* [10], only fits the final output of its feature extractor to the final output inferred by the feature extractor of a source network. In other words, the original feature adjustor is based on single-layer knowledge distillation. Here, we call the original feature adjustor a single-layer feature adjustor. On the other hand, our proposed framework is based on layer-wise knowledge distillation and a feature adjustor. To verify the effectiveness of the proposed layer-wise feature adjustor, we compare it with the single-layer feature adjustor which is denoted by "Single-layer". To apply the single-layer feature adjustor to SegNet, we replaced a feature extractor and a classifier seen in Endo *et al.* [10] with an encoder and a decoder of SegNet, respectively. The single-layer feature adjustor has to estimate the degradation level of degraded images as an auxiliary task. Due to the random scaling of data augmentations, we estimated the root mean square error instead of the JPEG quality factor as a degradation level for JPEG distortion. Regarding Gaussian blur and salt&pepper noise, we estimated the re-scaled standard deviation and the root mean square error, respectively.

Table 6 shows mIoUs of the single-layer and the proposed feature adjustors with three degradations: JPEG distortion, Gaussian blur and salt&pepper noise. Regarding clean images, "Single-layer" shows almost the same performance as "Proposed" for three types of degradation. However, "Proposed" shows better performance than "Single-layer" for low-quality images. Comparing average mIoUs, "Proposed" shows 0.009 higher than "Single-layer" for JPEG distortion. In the case of Gaussian blur and salt&pepper noise, average mIoUs of "Proposed" shows much higher than ones of "Single-layer."

Therefore, the proposed layer-wise feature adjustor is superior to the single-layer feature adjustor for the semantic segmentation of degraded images over various levels of degradation.

5. Conclusions

This paper has proposed the network of semantic segmentation to recognize degraded images over various levels of degradation, including clean images. The proposed layer-wise feature adjustor is trained to acquire the multi-layer features of digital images from a source network trained with only clean images. The effectiveness of the proposed framework was confirmed for several degradations and two famous datasets.

Although this paper only focused on semantic segmentation, we want to apply the proposed method to the object detection of degraded images. We also need to reduce the number of parameters for the proposed network because it is much bigger than the source network. Moreover, we would like to tackle the mixture of some degradations which is more realistic than only one degradation. These tasks are our future works.

References

[1] N. Aghli and E. Ribeiro. Combining weight pruning and knowledge distillation for cnn compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, September 2021.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, December 2017.

[3] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

[4] D. Cai, K. Chen, Y. Qian, and J.K. Kämäräinen. Convolutional low-resolution fine-grained classification. *Parttern Recognition Letters*, 119:166–171, March 2019.

[5] 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 Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.

[6] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE International Conference on Computer Vision*, December 2015.

[7] K. Endo, M. Tanaka, and M. Okutomi. Classifying degraded images over various levels of degradation. In *IEEE International Conference on Image Processing*, October 2020.

[8] K. Endo, M. Tanaka, and M. Okutomi. Cnn-based classification of degraded images. In *Proceedings of IS&T International Symposium on Electronic Imaging*, January 2020.

[9] K. Endo, M. Tanaka, and M. Okutomi. Cnn-based classification of degraded images with awareness of degradation levels. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):4046–4057, October 2021.

[10] K. Endo, M. Tanaka, and M. Okutomi. Cnn-based classification of degraded images without sacrificing clean images. *IEEE Access*, 9:116094–116104, August 2021.

[11] S. Ghosh, R. Shet, P. Amon, A. Hutter, and A. Kaup. Robustness of deep convolutional neural networks for image degradations. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2916–2920, 2018.

[12] D. Guo, Y. Pei, K. Zheng, H. Yu, Y. Lu, and S. Wang. Degraded image semantic segmentation with dense-gram networks. *IEEE Transactions on Image Processing*, 29:782–795, 2020.

[13] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2014.

[14] A. Janoch, S. Karayev, Y. Jia, J. T. Baron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3-d object dataset: Putting the kinect to work. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1168–1174, 2011.

[15] D.P. Kingma and J. Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[16] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*, 2020.

[17] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision*, volume 1, pages 1520–1528, 2015.

[18] Y. Pei, Y. Huang, and X. Zhang. Consistency guided network for degraded image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(6):2231–2246, June 2021.

[19] Y. Pei, Y. Huang, Q. Zou, H. Zang, X. Zhang, and S. Wang. Effects of image degradations to cnn-based image classification. *arXiv:1810.05552*, 2018.

[20] Y. Pei, Y. Huang, Q. Zou, X. Zhang, and S. Wang. Effects of image degradation and degradation removal to cnn-based image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(4):1239–1253, 2021.

[21] X. Peng, J. Hoffman, S.X. Yu, and K. Saenko. Fine-to-coarse knowledge transfer for low-res image classification. In *IEEE International Conference on Image Processing*, 2016.

[22] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations*, 2015.

[23] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

[24] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015.

[25] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012.

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[27] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 567–576, 2015.

[28] S. Wan, T. Wu, H. Hsu, W. Wong, and C. Lee. Feature consistency training with jpeg compressed images. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(12):4769–4780, December 2020.

[29] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *IEEE International Conference on Computer Vision*, pages 1625–1632, 2013.

[30] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations*, May 2016.

[31] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. Deep layer aggregation. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.