IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 4, 2021, accepted April 16, 2021, date of publication April 22, 2021, date of current version May 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3075084

Highly Contrast Image Correction for Dim Boundary Separation of Image Semantic Segmentation

JINYEOB CHOI[1] AND BYEONGDAE CHOI[2,3]
[1]School of Electronics Engineering, College of IT Engineering, Kyungpook National University, Daegu 41566, South Korea
[2]ICT Research Institute, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu 42988, South Korea
[3]Department of Interdisciplinary Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu 42988, South Korea
Corresponding author: Byeongdae Choi (bdchoi1@dgist.ac.kr)

ABSTRACT The efficiency and accuracy of the image semantic segmentation algorithm represent a trade-off relationship, and the loss of accuracy tends to increase as the model structure simplifies to improve efficiency. Developing more efficient and accurate algorithms requires methods to complement them. In this study, we applied the logarithmic-exponential mixture (LEM) function for gamma correction of images to improve the accuracy of image semantic segmentation. The basic model used in this work was produced by constructing a full convolution neural network based on MobileNetV2. To avoid the noise of input compression, we corrected training and validation images with gamma from 1/8 to 8 (7 different levels) before doing convolution. We evaluated models using Tensorflow deep-learning library based on Python. We compared models using LEM function to models using conventional gamma function. The prediction masks of the proposed model using the LEM function had relatively small fluctuations of accuracy upon gamma change. For images that have shadows overlapped on the object, the object was better distinguished in small gamma values. For dark images, the increase in accuracy was more effective. The results indicated that the proposed gamma correction could improve image segmentation accuracy in images with unclear edges. We believe that the presented results will guide further studies for accuracy improvement of image recognition algorithms applicable to future devices, such as autonomous vehicles and mobile robots.

INDEX TERMS Convolutional neural networks, image semantic segmentation, gamma correction, logarithmic function, exponential function.

I. INTRODUCTION

In neural network of machine learning algorithms, taking in convolution which applied to small and specific areas made the accuracy of image classification and recognition highly improved [1]–[7]. LeCun *et al.* introduced multilayered convolutional neural networks (CNNs) for handwritten digits and zip codes image classification. Simard *et al.* used data augmentation to improve quality and quantity of datasets [8]. Also, back-propagation and using GPU made complex algorithms possible to run and decreased performing time [9]–[13]. Based on these successes in image recognition fields, CNN algorithms have shown excellence in other machine learning areas, such as natural language and signal processing [14]–[16].

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

In recent years, mobile infrastructure is rapidly expanding and developing. In addition, the needs of deep learning in mobile environments increase. However, CNN-based algorithms require repetitive processing of high-density image data, so they are not suitable for mobile devices with low computing power. To solve this problem, many models, including MobileNets, have been proposed [17]–[20]. These models simplify the structure and decrease repeats and calculations, resulting in high performance on mobile devices. Although, efficiency and accuracy represent a trade-off, and the loss of accuracy increases as the model structure is simplified. For practical application of models, it is required to improve the accuracy that is degraded by structural simplification.

In this study, we tested the performance of the models designed for working in low computing power and considered their improvement by applying gamma correction.

We evaluated models with images that have clear edges between object and background. Datasets that applied gamma correction were inserted to models as training datasets and we observed the difference in a variety of gamma values. Conventional gamma correction controls image brightness based on a human perception structure, we considered that this is not fitted in computing image segmentation [21].

The main contributions of this study are as follows: (1) we applied an enhanced gamma correction function, logarithmic-exponential mixture (LEM) function, preferable to image segmentation model based on the encoder-the decoder framework. (2) We show how this function can be applied to backbone models. (3) We observed results from datasets in the LEM function. Then, we compared and explained them with the results from the datasets in the conventional gamma function. Models trained by datasets corrected with LEM function showed improved accuracy in areas where brightness was saturated. Images with unclear edges and with edges overlaid on shadows have improved accuracy at specific gamma values.

II. RELATED WORK

Segmentation is labeling image pixels one by one to discriminate objects. In order to do this, we need features of the input image, coarse maps. Conventional CNN-based image segmentation models are constructed with two parts. One part is an encoder that performs convolution and pooling repeatedly to extract feature maps, while the other part is a decoder that restores the size of feature maps to the input image size. Thus, it provides pixelwise prediction. Fully convolutional neural network (FCN) is a representative method. The encoder part is constructed based on classification models such as VGG and ResNet, and the fully connected layer is removed [22]–[24]. In the decoding part, information near feature is recovered by bilinear interpolation, backward convolution, and skip-connection. Another representative segmentation model is U-net, which constructs skip architecture that uses information from down-sampling to recover information in up-sampling which improves spatial details [25]. The decoder-encoder networks have been applied to many recent semantic segmentation tasks, including DeepLabV3+, and have provided successful results [26]–[32].

Meanwhile, depthwise separable convolution is designed to perform spatial separable convolution and pointwise convolution tasks in succession and has been applied to many neural network designs to improve the efficiency of the model [33]. Recently, MobileNet adapted depthwise separable convolution algorithm to reduce calculations with floating point operations and convolutional parameters [17]. This allowed MobileNet to work well enough in a low computing power environment. One step further, MobileNetV2 suggested inverted bottleneck residual blocks, and made more efficient performance in image classification [18]. We expected to further improve accuracy by adding

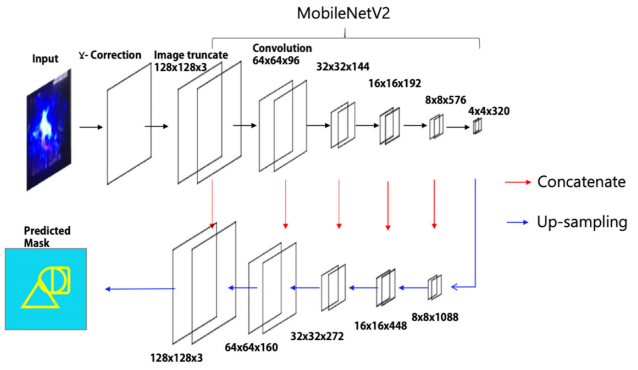

FIGURE 1. Model architecture to predict mask of the image.

gamma correction features to these functionally efficient algorithms, and we used MobileNetV2 as a backbone of our conceptual verification model. We also used ResNet-50 as a backbone for a comparison model [34]. ResNet is a classification model that applies residual blocks [24]. This occurs shortcuts inside the training process, and solves the vanishing gradient problem. Table 1 compares the features of this work and the previously proposed works.

III. METHODOLOGY

A. OVERALL ARCHITECTURE

Fig. 1 shows the architecture for the image segmentation model used in the study. First, the intensity of the input image is corrected before changing the input image size. Input images have various sizes with uint8 digits, consequently, images are compressed into 128 × 128 size format. In this process, 256 intensity levels are normalized and changed into float numbers from 0 to 1. If we apply gamma correction after normalization, this correction may differ in image by image. We tried to reduce the impact on the intensity of brightness in process of compression. Therefore, we corrected brightness when the image was in uint8 format and created a variety of training datasets.

First, the input image is corrected with gamma correction. After changing the image to 128 × 128 size format with 3 RGB channels, it goes through the MobileNetV2 algorithm for down-sampling to make 4 × 4 × 320 feature map. Then, extracted feature map like Fig. 1 passes 5 up-sampling processes and becomes 128 × 128 × 3 image. 3 channels of the input image are RGB channels, but 3 channels of the resulting mask are an object, edges of the object, and others.

Up-sampling process has been constructed with Convolution2d-Transpose and dropout. If we simply conduct up-sampling, the loss cannot be recovered properly, therefore we employed skip-connection to support up-sampling like U-Net. In up-sampling process, we used the layer of the same size that was in down-sampling process. In the convolution process in down-sampling, if the map passes the block of stride 2, the map shrinks in half, and these maps are used in skip-connection by concatenating after up-sampling layer to support recovering features.

TABLE 1. Comparison of image semantic segmentation models.

Network	Backbone	Feature	Strength	Weakness
FCN	VGG-16	Pixelwise end to end	Pixelwise prediction	Low performance
U-Net	VGG-16	U-shape architecture	Separation of touching objects of the same class, Fast segmentation	Different input and output size, Computational complexity [35]
DeeplabV3+	Xception	Encoder-Decoder, Atrous Separable Convolution	Refine object boundary, Lightweight,	Low performance for objects with similar properties, Long model training time [36]
EfficientNet-L2+NAS-FPN	EfficientNet-L2	Self-training	With pre-training shows even higher performance	Complicated process
Eff-B7 NAS-FPN	EfficientNet-B7	Data augmentation through image editing, copy-paste	Portability, Robustness to training schedule, High performance	Complicated process, Computational complexity
PSA w/ EADER DeepLab	Xception	End to end adversarial erasing method (EADER)	Reduce singularity in attention	Complicated process, Low performance
Our approach	MobileNetV2 ResNet-50	Gamma correction for enhancing boundary contrast, LEM function for gamma correction	Refine object boundary, Portability, Lightweight	-

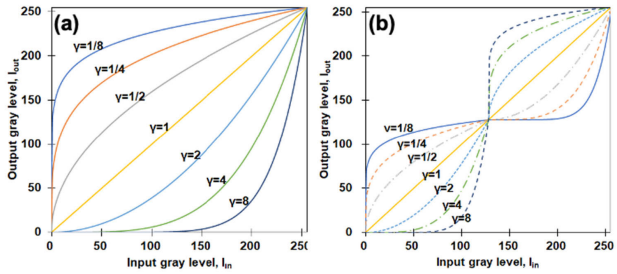

FIGURE 2. Intensity correction by gamma function (gray level: 0 ∼ 255) (a) conventional function (b) LEM function.

B. GAMMA CORRECTION

Gamma correction is one of the methods that can be used to create augmented images. In general, gamma correction applies a nonlinear function to change the brightness of an image to suit the human vision [21]. When the natural image is taken by a camera, it is visualized in a monitor, applied gamma correction that is suited for human sight.

$$I_{out} = G\left(\frac{I_{in}}{I_M}\right)^{\gamma} * I_M + I_O \tag{1}$$

The non-linear function is expressed as equation (1) where I_{out} is corrected intensity, G is gain, I_{in} is the original intensity of image, I_M is maximum intensity, and I_O is offset. Normally, image has uint8 format, which means the image has 256 levels of brightness, from 0 to 255. The corrected intensity of equation (1) is represented in Fig. 2 (a). In uint8 format, the maximum intensity level is 255, so I_M is equal to 255. Each pixel intensity is changed according to gamma value. If gamma is bigger than 1, the intensity decreases. If gamma is smaller than 1, the intensity increases.

C. PROPOSED LOGARITHMIC-EXPONENTIAL MIXTURE FUNCTION

Edges of an image are created by discontinuities in surface color, depth, surface normal, and shadow [37]. Humans can respond quickly to these discontinuities, but CNN algorithms are designed to be less sensitive to noise when generating feature maps, so responsiveness to noise is low. Thus, a small difference is hard to detect in the CNN process. Gamma correction is also suited for human vision, which responses sensitively to small changes. As a result, corrected images using ordinary gamma correction may not considerably improve the performance of CNN algorithm, and need of a new method that can emphasize the difference at the edges of objects appeared.

Like in Fig. 2 (a), ordinary gamma correction forms the logarithm or exponential function, which means saturated areas cannot emphasize the gradation difference. In low gamma value correction, the bright area is saturated and does not make a large gradation difference. In high gamma value correction, the dark area is saturated and does not make a large gradation difference. Convolution extracts feature maps based on the strong correlations between pixels, so it is difficult to detect the difference in brightness between objects and backgrounds in these brightness saturation areas. In a real dark environment, image brightness is saturated in the dark area and the segmentation model cannot define edges between object and background. We thought that the function that can make a large gradation difference in the brightness saturated areas could solve this problem.

$$I_{out} = \begin{cases} G\left(\dfrac{I_{in}}{\frac{1}{2} * I_M}\right)^{\gamma} * \dfrac{1}{2} * I_M + I_O \\ for \;\; I_{in} \le \dfrac{1}{2} * I_M \\ \left(G\left(\dfrac{I_{in} - \frac{1}{2} * I_M}{\frac{1}{2} * I_M}\right)^{\gamma} + 1\right) * \dfrac{1}{2} * I_M + I_O \\ for \;\; I_{in} > \dfrac{1}{2} * I_M \end{cases} \tag{2}$$

Sigmoid-form function and its symmetry function would be suitable for this. Suggested function is expressed as equa-

TABLE 2. The gamma value of training datasets for each model.

Conventional γ-correction		LEM γ-correction	
MODEL	γ, Train Dataset	Model	γ, Train Dataset
C_Model 1	1/8	LEM_Model 1	1/8
C_Model 2	1/4	LEM_Model 2	1/4
C_Model 3	1/2	LEM_Model 3	1/2
C_Model 4	1.0	LEM_Model 4	1.0
C_Model 5	2.0	LEM_Model 5	2.0
C_Model 6	4.0	LEM_Model 6	4.0
C_Model 7	8.0	LEM_Model 7	8.0

tion (2). This function is the LEM function. Fig. 2 (b) shows corrected intensity by equation (2) based on uint8 image. It can be seen that unlike conventional gamma function, LEM function can make the gradation difference clearly in the dark area and bright area.

D. TRAINING

To evaluate the LEM function, we first applied the conventional gamma function on the image dataset and trained models with it. By changing the gamma value, we got a variety of datasets. The value was changed in 7 levels from 1/8 to 8, all of them are powers of 2. We trained one model for one corrected dataset and made 7 models, which can be defined by different gamma values. We named them from C_Model 1 to C_Model 7. Also, we made 7 different test images using gamma correction and used them as input to models. This process resulted in 49 predicted masks. To observe more precisely a small change of accuracy, we made the more small degree of gamma value difference for test images from 0.7 to 1.3. For the LEM function, we repeated the same process as the conventional gamma function and made the results. The models are named from LEM_Model 1 to LEM_Model 7. Table 2 shows gamma values for each condition.

IV. EXPERIMENTS

A. DATASET

In this work, we used the Oxford-IIIT pet dataset [38]. This dataset is a 37 category pet dataset with roughly 200 images for each class. Training and validation sets consisted of 3680 and 3669 images, respectively. The basic model trained with the dataset that was not affected by gamma correction showed training accuracy 0.93 and validation accuracy 0.89, respectively. To observe the performance of the LEM function, we used images with clear edges and images with shadows overlapped on the edges. We also used dark images to evaluate object segmentation performance in dark environments. For dataset evaluation, we used the PASCAL VOC2007 and the PASCAL VOC2012 datasets [39], [40].

B. IMPLEMENTAION DETAILS

We used Tensorflow library using python on Anaconda 3 virtual machine in Window 10 operation system. We used a

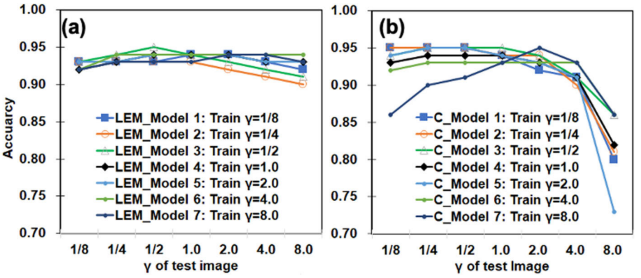

FIGURE 3. Accuracy of models trained with dataset corrected by gamma functions, (a) LEM function (b) conventional function. Image with clear edges.

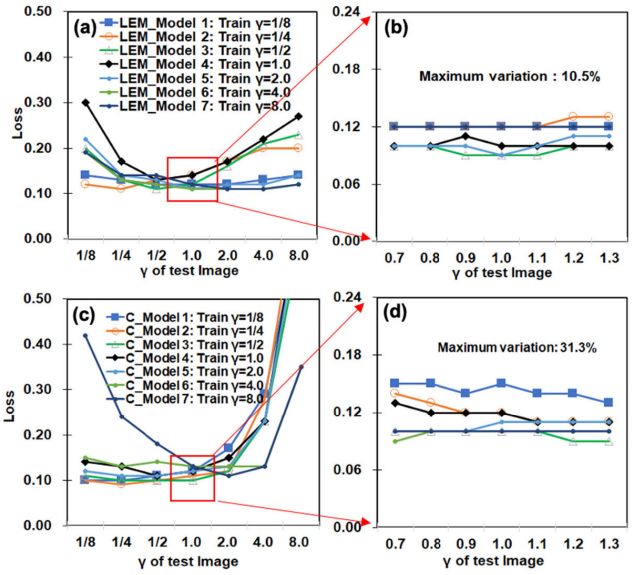

FIGURE 4. Loss of models trained with dataset corrected by gamma functions, (a) LEM function, (b) magnification in a narrow gamma range (0.7 ∼ 1.3) (c) conventional function (d) magnification in a narrow gamma range (0.7 ∼ 1.3).

laptop equipped with Nvidia GTX 1660 Ti with CUDA 11.0. We programmed the encoder part of the basic model based on MobileNetV2 and ResNet. In order not to deteriorate the original image characteristics, we configured it to correct the intensity before convolution. The feature map extracted from the encoder part was enlarged to the original image size through an up-sampling process in the decoder part. We first trained and validated the basic model with the Oxford-IIIT pet dataset and checked the process works properly. Then, we corrected datasets with conventional gamma correction method using gamma values from 1/8 to 8 and trained models with them. Next, we made 7 test data images with gamma values from 1/8 to 8 for the selected picture that has clear edges between object and background. Also, we made 7 test data images with gamma values from 0.7 to 1.3, each with 0.1 difference, for the selected picture and put them in models. For further testing, we created 7 test data images with gamma values from 1/8 to 8 for the pictures that have shadows overlapped on edges and pictures of dark and put them in models. The same method was employed for LEM function.

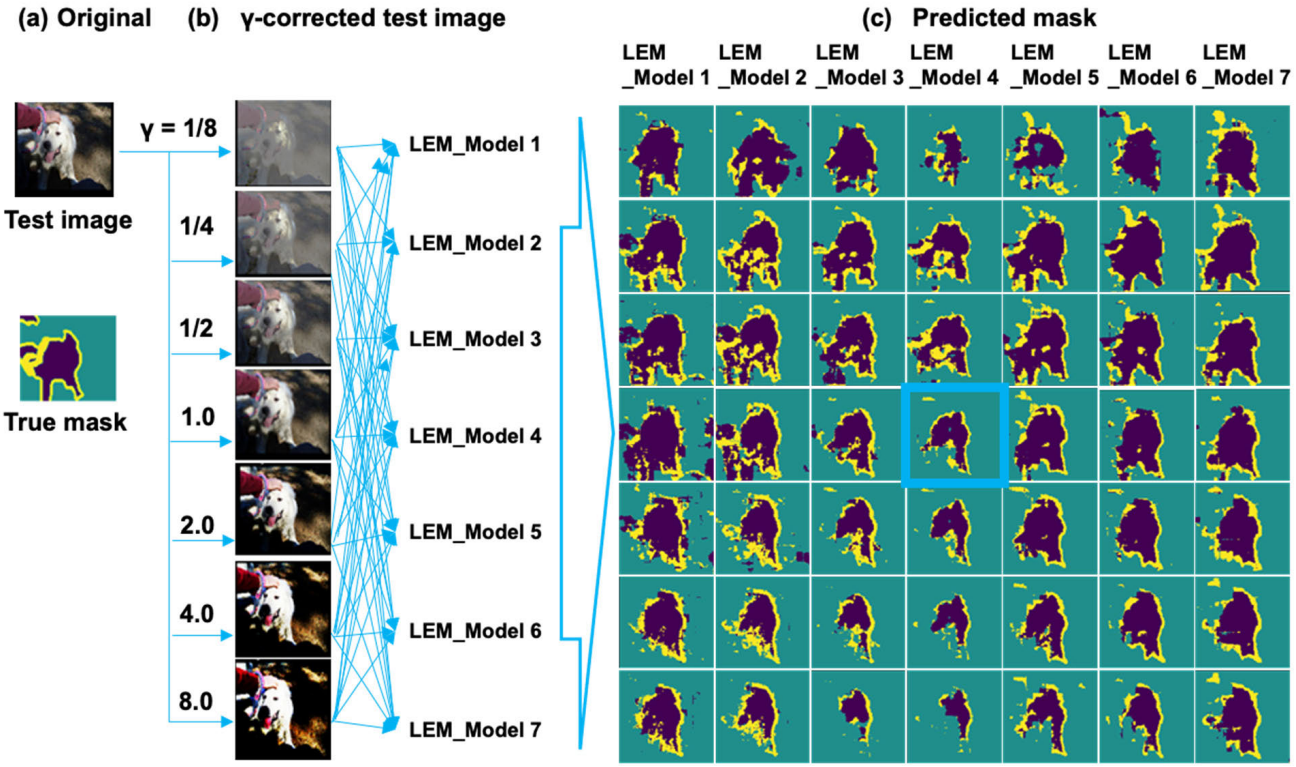

FIGURE 5. Evaluation of LEM_Models using test image with shadow overlapped on edges. (a) Original and true mask of the test image, (b) gamma-corrected test image, (c) predicted masks (reference mask: squared mask, gamma value of training dataset and test image is 1).

V. RESULTS AND DISCUSSION

A. PERFORMANCE COMPARISON OF LEM FUNCTION AND CONVENTIONAL FUNCTION

Fig. 3 compares the performance between models trained with datasets corrected by the LEM function and the conventional gamma function, respectively. Each line of graphs (Fig. 3 (a) and 3 (b)) represents each model, trained with datasets corrected by gamma 1/8, 1/4, 1/2, 1.0, 2.0, 4.0, 8.0, which was evaluated through test image with clear edges corrected by gamma 1/8, 1/4, 1/2, 1.0, 2.0, 4.0, 8.0.

In Fig. 3 (a), the accuracies in area of large gamma values of test image, such as LEM gamma 4.0 and 8.0 decreased about 0.9, but overall accuracy was placed between 0.95 and 0.9. For the accuracy of models using datasets corrected by conventional function (Fig. 3 (b)), C_Model 7 that was trained with datasets corrected by gamma 8.0, showed a high variance in accuracy rather than other models. Also, when the test image was corrected on gamma on 4.0 and 8.0, all models revealed bad accuracy. Comparing the results of conventional function and LEM function, models employed data corrected by LEM function showed fairly uniform results over the evaluated gamma range.

The loss of models showed the same trend as displayed in fig. 4. In the LEM gamma range from 1/4 to 2.0, the losses were under 0.2 and the others were over 0.2. (See Fig. 4 (a)) Losses by LEM function were in low variance in a wide gamma range compared to conventional function. (See Fig. 4 (c))

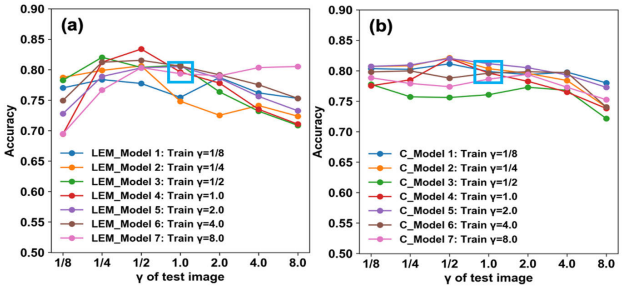

FIGURE 6. Image segmentation accuracy of models using test image with shadow overlapped on edges. Data images corrected by (a) LEM function, (b) conventional function (reference: squared points, gamma value of training dataset and test image is 1).

However, around gamma value of 1, the accuracy (Fig. 3) and loss (Fig. 4 (a) and 4 (c)) of models were too dense to directly clarify. Therefore, to investigate detailed variation around the gamma value of 1, we evaluated the loss in the correction gamma range of the test image from 0.7 to 1.3. The results from the LEM function and conventional function were shown in Fig. 4 (b) and 4 (d), respectively. In models using images corrected by the LEM function, LEM_Model 1, 2, and 7 had higher loss than other models, and the maximum loss variation between models was about 10.5%. Fig. 4 (d) shows the loss of models using the conventional function in the small gamma range from 0.7 to 1.3, the models revealed the maximum loss variation between

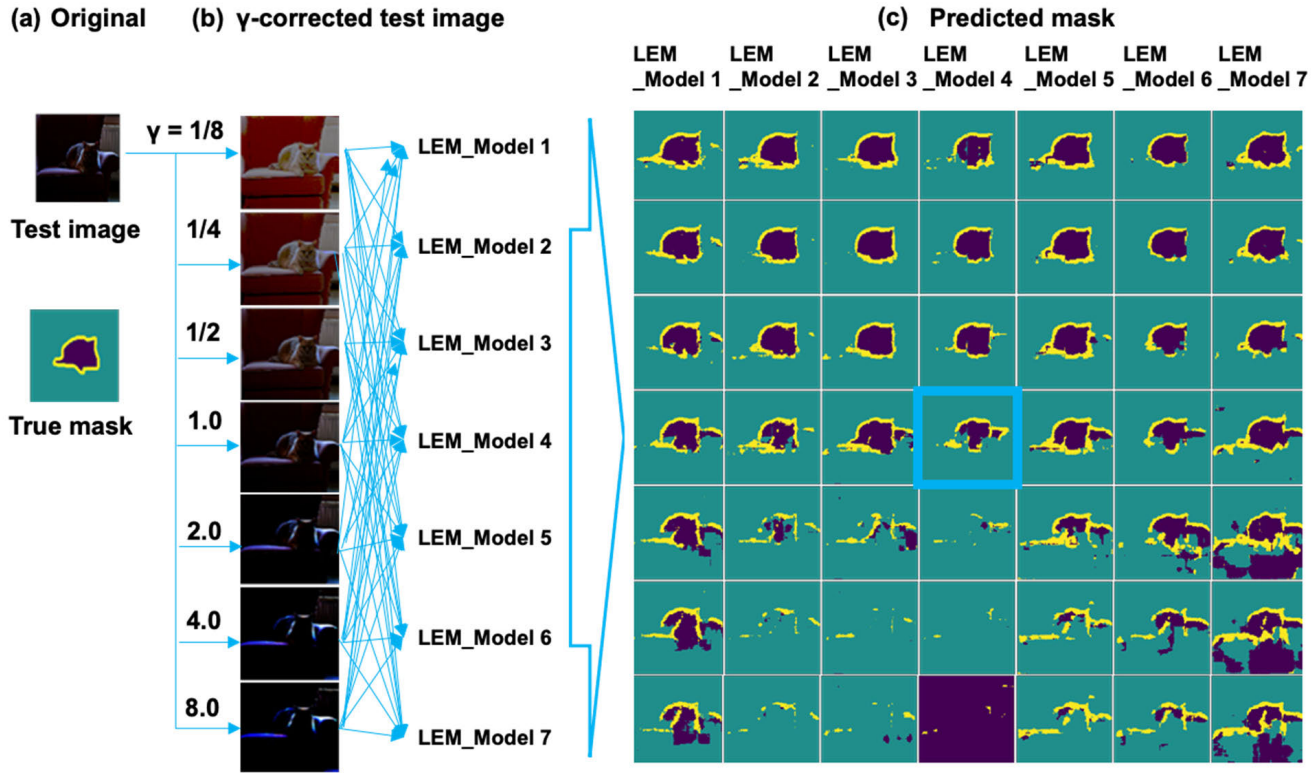

FIGURE 7. Evaluation of LEM_Models using dark test image that has unclear edges. (a) Original and true mask of the test image, (b) gamma-corrected test image, (c) predicted masks (reference: squared mask, gamma value of training dataset and test image is 1).

models of about 31.3%, which was higher than that using the LEM function. This result came from the difference in structure between LEM function and conventional function. Conventional gamma function was hard to make large gradation difference in the brightness saturated areas at the same time. In contrast, the LEM function was designed to make large gradation difference in dark and light areas. Therefore, when the brightness was in the saturated areas, models using images corrected by LEM function revealed higher accuracy and lower loss than that using the conventional function.

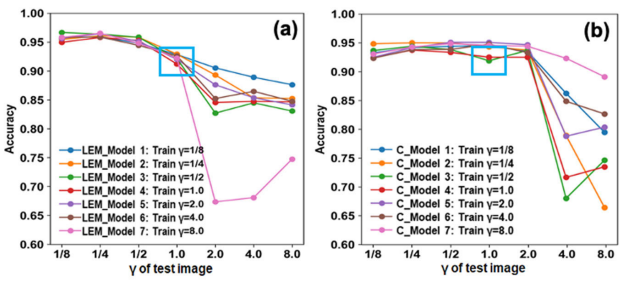

FIGURE 8. Image segmentation accuracy of models using dark test image that has unclear edges. Data images corrected by (a) LEM function, (b) conventional function (reference: squared points, gamma value of training dataset and test image is 1).

B. IMAGES WITH SHADOW OVERLAPPED ON EDGES

Observing and checking the features of the LEM function in detail will be helpful for using it in the real environment. In order to do this, we tried to test on an image that has a shadow overlapped on the object.

We added LEM gamma correction on the test image and evaluate the performance of LEM_Models. Fig. 5 shows the results. First, the original test image (Fig. 5 (a)) was corrected by LEM function for gamma from 1/8 to 8.0, and the resulting images (Fig. 5 (b)) input to each model. The predicted masks derived by each model were shown in Fig. 5 (c). Comparing to the reference mask, blue-squared mask in Fig. 5 (c), the predicted masks derived using test images corrected with gamma values of 1/4 and 1/2 showed much closer shapes to the true mask. As the LEM gamma values became smaller, the test images became more blur and gray. For this reason, on test images with LEM gamma 1/2 and 1/4, we could define the edge more precisely which was hidden in the shadow.

The accuracy figures of LEM_Models shown in Fig. 6 (a) demonstrates these results quantitatively. The accuracy of the model was improved by over 6% through the optimization of data images. This is also superior to the results from the conventional function of Fig. 6 (b).

C. EVALUATION OF DARK IMAGES

As mentioned above, the LEM function can provide a large gradation difference in the brightness saturated areas. It can be more effective in distinguishing objects from dark images

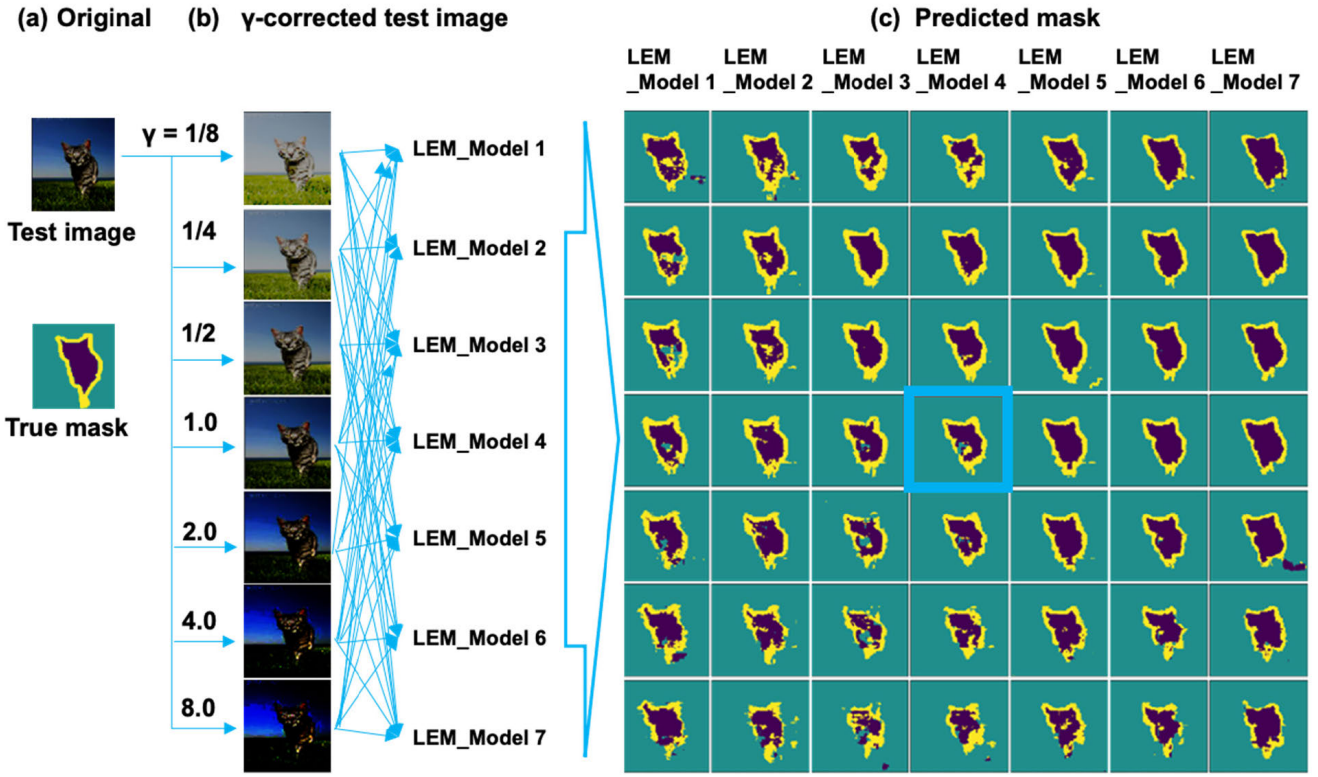

FIGURE 9. Evaluation of LEM_Models using dark test image that has perceptible edges. (a) Original and true mask of the test image, (b) gamma-corrected test image, (c) predicted masks (reference: squared mask, gamma value of training dataset and test image is 1).

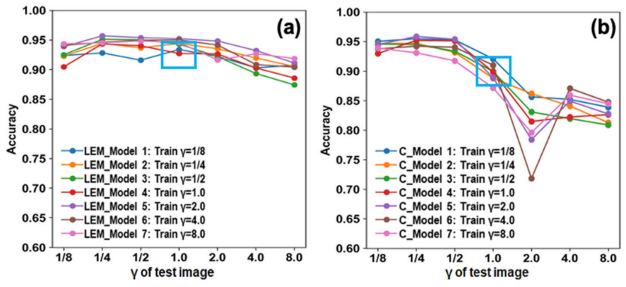

FIGURE 10. Image segmentation accuracy of models using dark test image that has perceptible edges. Data images corrected by (a) LEM function, (b) conventional function (reference: squared points, gamma value of training dataset and test image is 1).

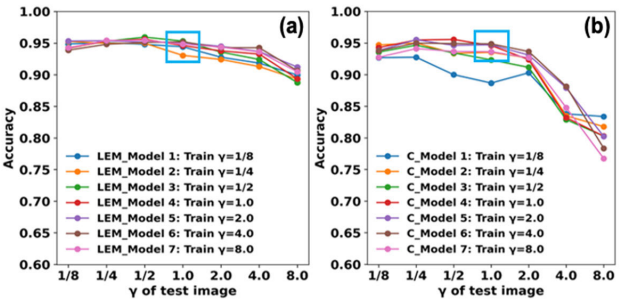

FIGURE 11. Image segmentation accuracy of ResNet-50-based models using dark test image that has perceptible edges. Data images corrected by (a) LEM function, (b) conventional function (reference: squared points, gamma value of training dataset and test image is 1).

critical for the practical use of the model. To test this idea, we selected two kinds of dark images; one image has no clear edges between the object and the background, and the other has perceptible edges.

Fig. 7 shows image segmentation result from dark images with unclear edges between the object and the background. Dark original image (Fig. 7 (a)) was corrected with the LEM function, and corrected images were shown in Fig. 7 (b). We evaluated 7 models from LEM_Model 1 to LEM_Model 7 with images of Fig. 7 (b), and Fig. 7 (c) are the predicted masks. The result indicated that if the LEM gamma value of the test image was smaller than 1.0, the predicted mask became closer to the true mask than the reference. However, when the LEM gamma value of the test image was higher than 1, from LEM_Model 2 to Model 6, predicted masks could not define the object. LEM_Model 7 caught the object, but the background noise was much higher than other masks. This was because, in gamma 8, LEM function increased the difference in gradation only in medium gray.

In contrast, LEM_Model 1 shown a better-predicted mask than other models in the LEM gamma range over 1. LEM_Model 1 and 7 roughly caught the outline of the object, but LEM_Model 1 showed much less background noise. This was due to LEM gamma correction by very low values, such

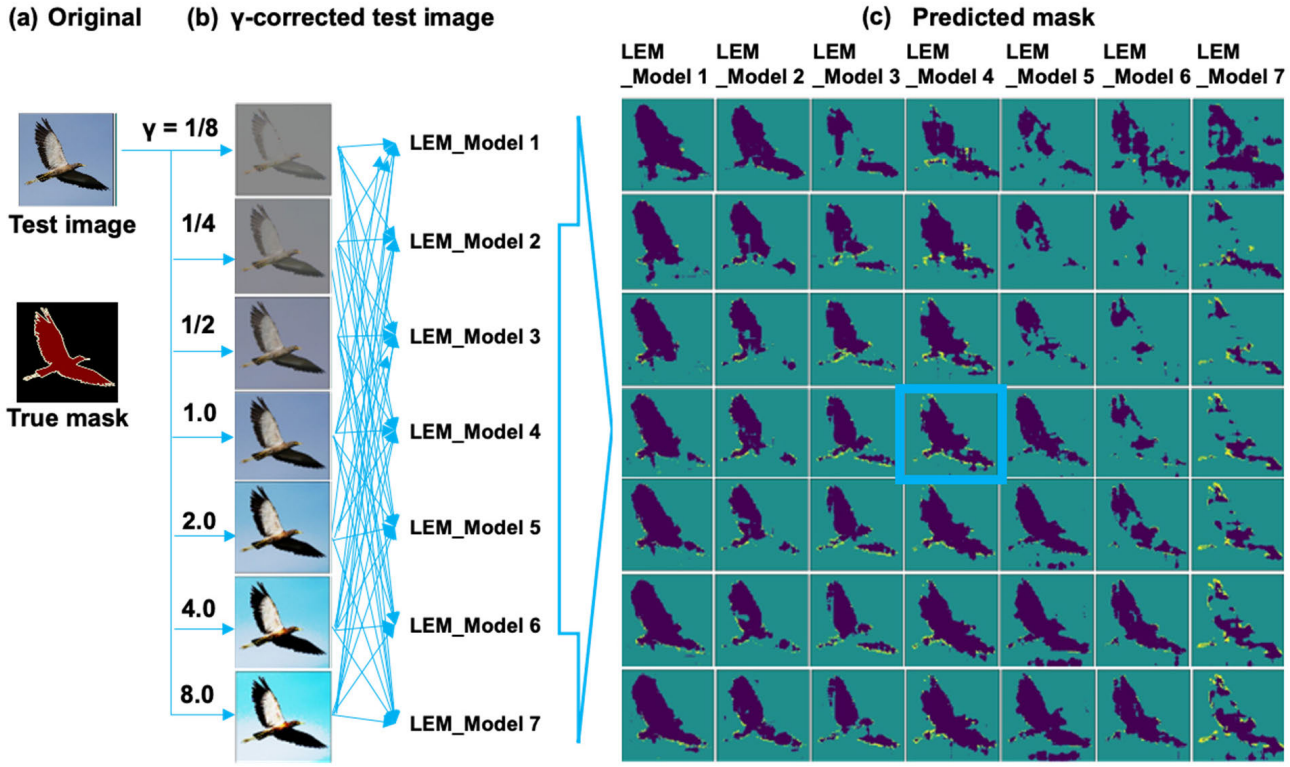

FIGURE 12. Evaluation of LEM_Models using a flying bird test image with clear edges. (a) Original and true mask of the test image, (b) gamma-corrected test image, (c) predicted masks (reference: squared mask, gamma value of training dataset and test image is 1).

as 1/8, which finely differentiates even the smallest difference in brightness in dark areas. Since the LEM_Model 1 was trained with a dataset corrected by gamma 1/8, it was good to distinguish very small brightness differences compared to other models when very dark images were input.

Fig. 8 compares the accuracy of models using dark test images with unclear edges corrected by LEM function and conventional function. For the models using the LEM function (Fig. 8 (a)), if the test image had lower gamma than 1, the models revealed higher accuracy than that with gamma 1. All of the test images corrected with a gamma of less than 1 had an accuracy of over 90%. Comparing to models using the conventional function, as shown in Fig. 8 (b), the accuracy of LEM_Models using image data corrected on a lower gamma value than 1 superior to that using conventional functions.

In addition, the accuracy of models for dark images with perceptible edges between object and background was uniform over the evaluated gamma range as presented in Fig. 9. Comparing the reference in Fig. 9 (c) with the other predicted masks, if the LEM gamma value of test image correction was lower than 1, the predicted mask was closer to the true mask. This trend increased with the LEM gamma value of training dataset correction; models trained with dataset corrected by LEM function on high gamma value showed higher accuracy than that by low gamma value. This resulted from the feature of the LEM function. Dark image corrected with a high LEM

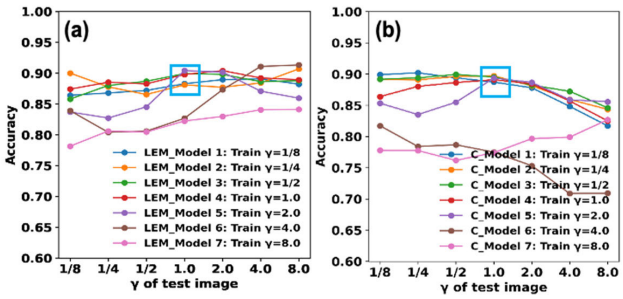

FIGURE 13. Image segmentation accuracy of models using a flying bird test image with clear edges. Data images corrected by (a) LEM function, (b) conventional function (reference: squared points, gamma value of training dataset and test image is 1).

gamma value got more clear edges after correction as shown in Fig. 9 (b).

In Fig. 10 (a), the model accuracy was more than 85% over the evaluated gamma range. Some results, which were a combination of low-test image gamma and high-training dataset gamma, were above 95%. Unlike that using conventional function displayed in Fig. 10 (b), dark images with perceptible edges showed good accuracy at all gamma values.

On the other hand, testing reproducibility is very important for understanding the scalability of the proposed concept and finding potential applications of the model. To do this, we constructed an image segmentation model using

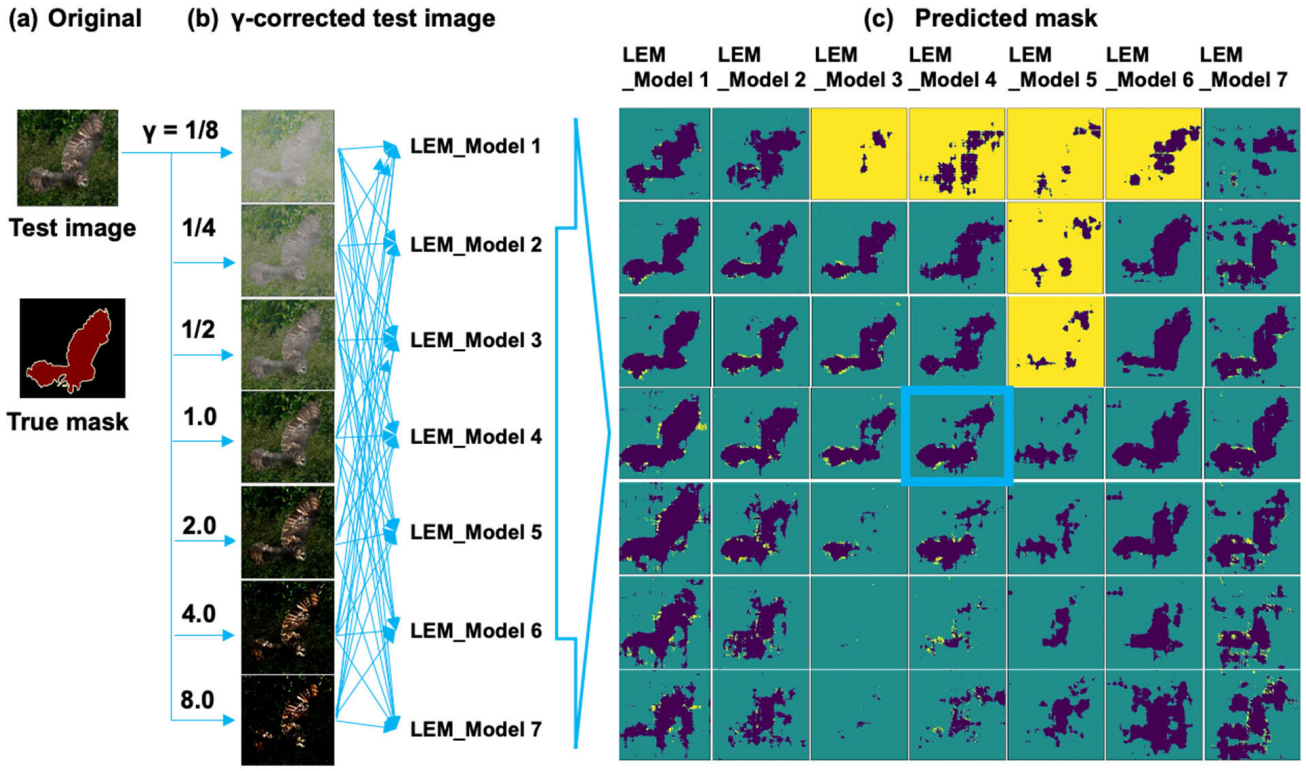

FIGURE 14. Evaluation of LEM_Models using a bird in bush test image with unclear edges. (a) Original and true mask of the test image, (b) gamma-corrected test image, (c) predicted masks (reference: squared mask, gamma value of training dataset and test image is 1).

ResNet-50 as the backbone. We trained and evaluated the model the same as we did in MobileNetV2-based models. Fig. 11 shows the model accuracy on a dark test image that has perceptible edges. The accuracy variation of the model using the LEM function (Fig. 11 (a)) was much smaller than that of the model (Fig. 11 (b)) using the conventional function. These results are similar to those of the MobileNetV2-based model described in Fig. 10. Furthermore, tests on dark images with unclear edge showed similar results on ResNet-50-based models as on MobileNetV2-based models.

D. DATASET EVALUATION

Dataset is a collection of different images. Model parameters can be changed when training with various datasets. To investigate the variability for datasets, we trained our models with the PASCAL VOC2007 dataset. Also, we picked some pictures with clear and unclear edges in the PASCAL VOC2012 dataset to validate the trained models.

Fig. 12 shows the set of predicted masks of a flying bird with clear edges. Fig. 12 (a) shows the original image and true mask. In the original image, part of the left wing is shadowed, and the main object is distinguished from the background. Corrected images by the LEM function and predicted masks are represented in Fig 12 (b) and 12 (c), respectively. The results indicated that the test image with a large gradation difference, where the LEM gamma value of the test image

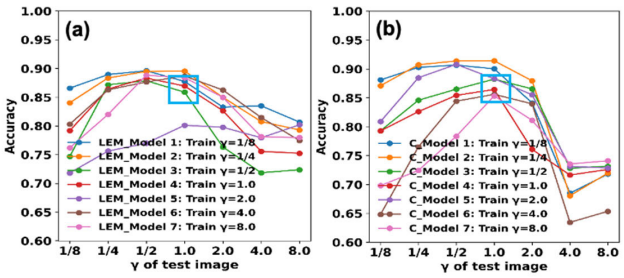

FIGURE 15. Image segmentation accuracy of models using a bird in bush test image with unclear edges. Data images corrected by (a) LEM function, (b) conventional function (reference: squared points, gamma value of training dataset and test image is 1).

was high, made the predicted mask more precisely to the true mask than reference marked by blue rectangular.

The accuracy of the LEM models shown in Fig 13 (a) also increased as the gamma value of the test image increased. This was caused by a clearer separation of the boundaries between objects and backgrounds by LEM correction at high gamma values. In contrast, the accuracy of models using conventional function decreased as gamma values increased. (Fig. 13 (b)) Furthermore, the models trained with the gamma values of 4.0 and 8.0 have shown low accuracy. These were because increasing gamma values by conventional function darkened the image and made it difficult to distinguish boundaries.

We could find slightly different results from the test image of a bird surrounded by grasses with unclear edges. Fig. 14(a) shows the original image and true mask. Because the wings are buried in the background, it is hard to identify the precise edge. In Fig. 14 (b), LEM corrected images are represented and these images tried to make unclear edges to clear edges, but still hard to clarify to the human vision. However, the predicted masks in Fig. 14 (c) showed that results from the LEM models trained low or high gamma values were more accurate than the reference mask marked by blue rectangular.

Further, the accuracy shown in Fig. 15 (a) revealed a large value in the test image corrected with a low gamma value of less than 1.0. However, in some conditions, the predicted masks showed poor results. This was because the similarity between the object and the background of the image was so large that they could not be distinguished under those conditions. On the other hand, compared to the conventional function shown in Fig. 15 (b), the LEM models had higher accuracy in the high gamma values and showed lower variation among models than the conventional function models. This trend is similar to the results using the Oxford-IIIT pet dataset described earlier.

VI. CONCLUSION

In this work, we suggested a gamma correction algorithm using the LEM function to improve the image segmentation performance of CNN-based models. Before convolution, we added functions for gamma correction of input images. In the models trained with dataset corrected by LEM function, the image with clear edges showed insensitive on gamma difference than conventional function. Image with shadow overlapped edges corrected by LEM function on small gamma value was defined edges precisely. A dark image with perceptible edges of the object was defined well, regardless of LEM gamma value. The optimized accuracy of the model using the LEM function was superior to that of the conventional function. This was because the LEM function made gradation difference large at the boundaries of the image.

REFERENCES

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Dec. 1998.

[2] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1090–1098.

[3] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. Lecun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3626–3633.

[4] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *CoRR*, vol. 2014, pp. 1–5, Aug. 2014.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[6] I. S. Fomin, S. R. Orlova, D. A. Gromoshinskii, and A. V. Bakhshiev, "Object detection on docking images with deep convolutional network," in *Proc. Adv. Neural Comput., Mach. Learn., Cognit. Res.*, 2019, pp. 136–143.

[7] L. V. Utkin, "An imprecise extension of SVM-based machine learning models," *Neurocomputing*, vol. 331, pp. 18–32, Feb. 2019.

[8] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Proc. 7th Int. Conf. Document Anal. Recognit.*, Edinburgh, U.K., 2003, pp. 958–963, doi: 10.1109/ICDAR.2003.1227801.

[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[10] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[11] B. Catanzaro, N. Sundaram, and K. Keutzer, "Fast support vector machine training and classification on graphics processors," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 104–111, doi: 10.1145/1390156.1390170.

[12] B. He, M. Lu, K. Yang, R. Fang, N. K. Govindaraju, Q. Luo, and P. V. Sander, "Relational Query coprocessing on graphics processors," *ACM Trans. Database Syst.*, vol. 34, no. 4, pp. 1–39, Dec. 2009.

[13] I. Buck, T. Foley, D. Horn, J. Sugerman, K. Fatahalian, M. Houston, and P. Hanrahan, "Brook for GPUs: Stream computing on graphics hardware," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 777–786, Aug. 2004.

[14] E. N. N. Ocquaye, Q. Mao, Y. Xue, and H. Song, "Cross lingual speech emotion recognition via triple attentive asymmetric convolutional neural network," *Int. J. Intell. Syst.*, vol. 36, no. 1, pp. 53–71, Jan. 2021, doi: 10.1002/int.22291.

[15] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplin. Rev. Data Mining Know Discov.* vol. 8, no. 4, p. e1253, 2018.

[16] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Cernocky, "Strategies for training large scale neural network language models," in *Proc. IEEE ASRU*, Dec. 2011, pp. 196–201.

[17] A. G. Howard, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861.

[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf.*, Salt Lake City, UT, USA, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[19] D. Sinha and M. El-Sharkawy, "Thin mobileNet: An enhanced mobileNet architecture," in *Proc. IEEE UEMCON*, New York, NY, USA, Jun. 2019, pp. 0280–0285, doi: 10.1109/UEMCON47517.2019.8993089.

[20] M. Tan and V. Quoc Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," 2019, *arXiv:1905.11946*. [Online]. Available: https://arxiv.org/abs/1905.11946

[21] C. Poynton, *Digital Video and HD*. Burlington, MA, USA: Morgan Kaufmann, 2012.

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

[23] X. Zhang, J. Zou, K. He, and J. Sun, "Accelerating very deep convolutional networks for classification and detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 1943–1955, Oct. 2016, doi: 10.1109/TPAMI.2015.2502579.

[24] 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.

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention* (Lecture Notes in Computer Science), vol. 9351, N. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds. Cham, Switzerland: Springer, 2015, pp. 1–7, doi: 10.1007/978-3-319-24574-4_28.

[26] J. Fu, J. Liu, Y. Wang, and H. Lu, "Stacked deconvolutional network for semantic segmentation," 2017, *arXiv:1708.04943*. [Online]. Available: http://arxiv.org/abs/1708.04943

[27] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," 2016, *arXiv:1611.06612*. [Online]. Available: https://arxiv.org/abs/1611.06612

[28] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*. [Online]. Available: http://arxiv.org/abs/1706.05587

[29] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with Atrous separable convolution for semantic image segmentation," 2018, *arXiv:1802.02611*. [Online]. Available: https://arxiv.org/abs/1802.02611

[30] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. V. Le, "Rethinking pre-training and self-training," 2020, *arXiv:2006.06882*. [Online]. Available: http://arxiv.org/abs/2006.06882
[31] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," 2020, *arXiv:2012.07177*. [Online]. Available: http://arxiv.org/abs/2012.07177
[32] E. Stammes, T. F. H. Runia, M. Hofmann, and M. Ghafoorian, "Find it if you can: End-to-End adversarial erasing for weakly-supervised semantic segmentation," 2020, *arXiv:2011.04626*. [Online]. Available: http://arxiv.org/abs/2011.04626
[33] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.
[34] H. G. Barrow and J. M. Tenenbaum, "Interpreting line drawings as three-dimensional surfaces," *Artif. Intell.*, vol. 17, nos. 1–3, pp. 75–116, 1981.
[35] N. Beheshti and L. Johnsson, "Squeeze U-net: A memory and energy efficient image segmentation network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Seattle, WA, USA, Jun. 2020, pp. 1495–1504, doi: 10.1109/CVPRW50498.2020.00190.
[36] B. Ayhan and C. Kwan, "Tree, shrub, and grass classification using only RGB images," *Remote Sens.*, vol. 12, no. 8, p. 1333, Apr. 2020, doi: 10.3390/rs12081333.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," 2016, *arXiv:1603.05027*. [Online]. Available: http://arxiv.org/abs/1603.05027
[38] O. M. Parkhi, A. Vedaldi, A. Zisserman and C. V. Jawahar. *The Oxford-IIIT PET Dataset*. Accessed: Oct. 10, 2020. [Online]. Available: https://www.robots.ox.ac.uk/~vgg/data/pets
[39] M. Everingham, L. Van-Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. Accessed: Mar. 11, 2021. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
[40] M. Everingham, L. Van-Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. Accessed: Mar. 11, 2021. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

JINYEOB CHOI is currently pursuing the B.S. degree in electronics engineering with Kyungpook National University, Daegu, South Korea. His research interests include machine learning and natural language processing.

BYEONGDAE CHOI received the B.S. and M.S. degrees in metallurgical engineering from Seoul National University, Seoul, South Korea, in 1988 and 1990, respectively and the Ph.D. degree in materials science and engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1999. Since 2005, he has been a Principal Researcher with the ICT Research Institute and a Professor with the Department of Interdisciplinary Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, South Korea. His research interests include innovative displays, oxide electronics, and deep learning applications.