IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 24, 2021, accepted July 8, 2021, date of publication July 12, 2021, date of current version July 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3096530

Tunable U-Net: Controlling Image-to-Image Outputs Using a Tunable Scalar Value

SEOKJUN KANG[1,2], (Member, IEEE), SEIICHI UCHIDA[1], (Member, IEEE), AND BRIAN KENJI IWANA[1], (Member, IEEE)

[1]Department of Advanced Information Technology, Kyushu University, Fukuoka 819-0395, Japan
[2]Global Research and Development Center, MANDO Corporation, Seongnam 13486, South Korea

Corresponding author: Seokjun Kang (seokjun.kang@human.ait.kyushu-u.ac.jp)

This work was supported in part by the Japan Society for the Promotion Science (JSPS) KAKENHI under Grant JP17K19402 and Grant JP17H06100.

ABSTRACT Image-to-image conversion tasks are more accurate and sophisticated than ever thanks to advances in deep learning. However, since typical deep learning models are trained to perform only one task, multiple trained models are required to perform each task even if they are related to each other. For example, the popular image-to-image convolutional neural network, U-Net, is normally trained for a single task. Based on U-Net, this study proposes a model that outputs variable results using only one trained model. The proposed method produces a continuously changing output by setting an external parameter. We confirm the robustness of our proposed model by evaluating it on binarization and background blurring. According to these evaluations, we confirmed that the proposed model can generate well-predicted outputs by using un-trained tuning parameters as well as the outputs by using trained tuning parameters. Furthermore, the proposed model can generate extrapolated outputs outside the learning range.

INDEX TERMS Image-to-image conversion, multiple tasks, U-Net, image binarization, background blur.

I. INTRODUCTION

Since the introduction of Convolutional Neural Networks (CNN) [1], studies on image analysis using deep learning have been actively conducted [2], [3]. For example, U-Net [4], a CNN-based image-to-image model, can perform precise image segmentation. These U-Net-based models have shown significant strength in image transformations [5], [6]. However, to be trained, these U-Net-based models require a dataset for each purpose. This means that even for task with little variations, such as a threshold for global binarization, a new U-Net needs to be trained. In contrast, traditional rule-based image processing techniques can easily change the output based on a parameter. Therefore, these rule-based techniques still are used for image transformation tasks [7]. Fig. 1 demonstrates examples of generating multiple outputs of traditional rule-based image processing (global binarization).

Even though U-Net-based models also can perform tasks similar to rule-based image processing methods, separately trained U-Nets models for each purpose are required.

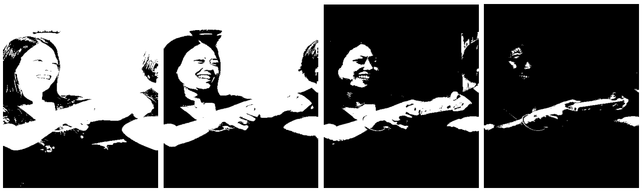
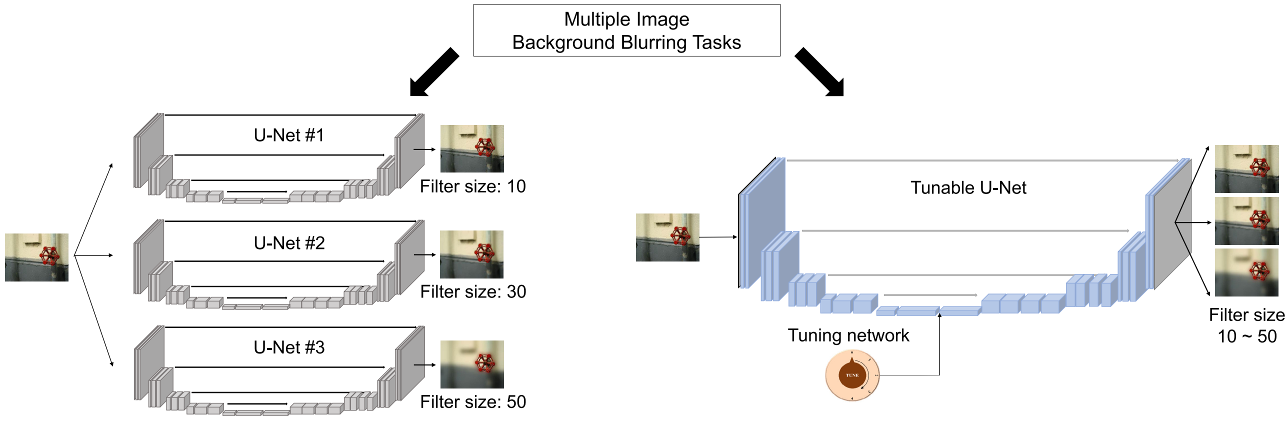

(a) 0.1 (b) 0.3 (c) 0.5 (d) 0.7

FIGURE 1. Examples of a traditional image processing technique. The traditional image processing technique (global binarization) can generate multiple outputs by tuning threshold values. Images from left to right correspond to the generated result by threshold values as 0.1, 0.3, 0.5, and 0.7.

For example, in the case of image binarization, traditional rule-based methods can utilize global threshold values easily, as in Fig. 1. However, if a U-Net-based model wants to perform the same task, individual models at each threshold is required.

In addition, U-Nets require a large number of well-annotated training samples. Generating annotated image data is can be difficult because it often requires expert guidance. Furthermore, the number of published training data can be insufficient, like historical document image

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



Typical U-Net training process

Tunable U-Net training process

FIGURE 2. Comparison of model training process between traditional U-Net and the proposed model. For conducting multiple similar tasks, several pre-trained U-Nets should be required. However, the proposed model can conduct multiple similar tasks by a single training process.

binarization. Consequently, securing several sets of training data and training several individual U-Nets is a difficult and time-consuming task.

Therefore, various solutions have been adopted in U-Net-based models, such as using data augmentation [8], utilizing compressed U-Nets [9], or designing conditional U-Nets with generative models [10]. In the case of data augmentation and compressed U-Net methods, even though it might be possible to secure the data, the process of training several models is required for generating outputs under different conditions. In the case of U-Nets with generative models like Generative Adversarial Networks (GAN) [11], even though they can tune outputs and generate multiple outputs [12], [13], the training process of the generative models are relatively complicated compared with the conventional U-Net's training process.

In this paper, we propose Tunable U-Net (TU-Net), a novel U-Net-based model to control the output efficiently. TU-Net is designed to conduct multiple similar tasks like Fig. 2. TU-Net is trained using a tuning parameter which can change the output without the requirement of multiple trained models. Through the proposed method, we can perform traditional image processing techniques that require a parameter like Fig. 1. The proposed model is designed as an end-to-end system.

To show the output controlling ability of TU-Net, we trained the proposed model to conduct two tasks: image binarization and image background blurring. Since traditional methods of image binarization can control the results by tuning threshold values, we train the proposed method similarly. Also, since background blurring can adjust the degree of background blurring by controlling filter size, it is used as the second task. In addition, background blurring demonstrates the usefulness of using a network over traditional methods because the model needs to adaptively learn the edges of the foreground and background. Through these tasks, we can evaluate the tuning ability of the proposed model effectively.

The primary contributions in this paper are as follows:

- We propose a novel architecture based on U-Net for tuning outputs called a TU-Net. In the proposed model, we propose the use of a novel tuning network for transforming from scalar values to features.
- For evaluating the tuning ability of the proposed model, TU-Net is evaluated on two tasks: image binarization and image background blurring.
- We evaluate the ability of TU-Net with detailed quantitative and qualitative results by using MSCOCO [14] and MSRA [15] datasets. The proposed model achieves better results compared with individually trained conventional U-Nets.
- We confirmed that TU-Net is able to generate extrapolated results by using untrained tunable input parameters. We analyze the qualitative results of the extrapolated outputs compared with the interpolated outputs.

The remaining of this paper is organized as follows. Section II reviews related work. Section III provides details of the proposed model and our considerations when we designed the proposed framework. Section IV demonstrates the quantitative and qualitative results. Finally, Section V is the conclusion.

II. RELATED WORK

In this section, we will briefly describe related works in image binarization and blurring, and conditional generative models.

A. IMAGE BINARIZATION AND BACKGROUND BLURRING METHODS

Since traditional binarization methods need a threshold value for dividing foreground and background areas, research for defining an optimal threshold value has been performed [16], [17]. However, since setting the optimal threshold values can be difficult, the generated results by traditional image processing methods are not always consistent [18].

With the development of CNNs, image binarization systems also have been improved. CNN-based binarization systems have shown significant performance [19], [20]. He and Schomaker [21] suggested a document image binarization method that adopts an iterative deep learning framework. Vo *et al.* [22] designed a CNN-based end-to-end binarization method by using multi-scale deep supervised networks. Also, after the introduction of generative models, image binarization systems with generative models have been suggested [23]. For enhancing the performance of image binarization, Bhunia *et al.* [24] and Zhao *et al.* [25] develop binarization systems using a conditional GAN (cGAN).

Image background blurring, called the *bokeh* effect, is a photographic technique used to highlight a subject. Recently, due to the increasing number of photographs taken with smartphones, background blurring systems based on various image processing techniques are utilized for complementing the limited specifications of smartphone camera sensors. Early image background blurring techniques were implemented based on image processing techniques [26]–[28]. However, with the development of image segmentation using CNNs, background blurring models using CNNs have been proposed. Shen *et al.* [29] suggested an image background blurring system with newly defined matting components. Their end-to-end system, by using novel matting layers, classifies background, foreground, and unknown labels without the user's intervention. Wadhwa *et al.* [30] performed an effective image background blurring by constructing a precise person segmentation network using several U-Nets.

B. CONDITIONAL GENERATIVE MODELS

GANs, proposed by Goodfellow *et al.* [11], perform a model training process through competitive learning between a generative model and a discriminate model. According to GAN's competitive learning process, GAN can create fake results similar to real data. However, since the outputs of GANs are uncontrollable, cGAN [31] was proposed by providing a condition. Since cGANs has shown the ability that can control the output, various cGAN-based models have been suggested [12], [32], [33].

Advanced cGAN-based models can be divided into three types of training data utilization: paired mapping, un-paired mapping, and multi-domain mapping. Isola *et al.* [12] proposed a novel photo-realistic image-to-image conversion model (Pix2pix) which uses a cGAN with paired mapping training data. However, preparing sufficient paired mapped training image data is difficult, cycleGAN [32], DiscoGAN [34], and GauGAN [35] were suggested for unpaired image-to-image translation. For conducting multi-domain image translation, starGAN was suggested to be trained with different multi-domain datasets [33]. These cGAN-based models have been also introduced in various fields: complex image translation with auxiliary classifier [36], high-resolution image synthesis [37], video synthesis [38], and pose-based image generation [39].

Variational Auto-Encoders (VAE), proposed by Kingma and Welling, can create results by utilizing latent values that are produced from the encoder [40]. Sohn *et al.* [41] suggested a conditional VAE (cVAE) for supervised and semi-supervised learning. cVAE is trained and generates results with the condition information. Esser and Sutter [10] performed conditional appearance and shape generation using U-Net and VAE. They can synthesize outputs with that appearance in different geometrical layouts by concatenating the inferred appearance representation of VAE with the bottle-neck representation of U-Net. cVAE based models have been introduced in various fields: cVAE and GAN for fine-grained image generation [42], and image generation of people in clothing [43].

C. DIFFERENCE BETWEEN TU-NET AND cGAN-BASED MODELS

The proposed method, TU-Net, has several differences from existing conditional generative models. The differences are as follows:

- Compared with cGAN-based models, TU-Net adopts a single loss function and simpler architecture without a discriminator network. Since cGAN-based models create outputs heuristically by competing generator and discriminator networks, cGAN-based models should be trained by several loss functions. Therefore, the model training of cGAN-based models is difficult compared to TU-Net. However, TU-Net can generate outputs by adopting the general architecture of image generation networks like U-Net.
- Compared with cVAE-based models [41], TU-Net can use a single scalar parameter to change the output and it requires no modification to the input. In a TU-Net, the whole model can be frozen and the output is changed using only the tuning parameter.

III. THE PROPOSED MODEL

The TU-Net contains two parts. A conventional U-Net to perform the image processing task and a tuning network to control the output. This section will describe the parts.

A. U-NET IN TU-NET

The proposed model is based on a U-Net [4] architecture. As shown in Fig. 3, an input image first passes through the contracting path of U-Net. The contracting path is based on VGGNet [44]. Like VGGNet, the contracting path of U-Net extracts the features of an input image by using convolutional layers and down-samples the extracted features by using pooling layers. Through each down-sampling, the number of channels is doubled and the size of the feature maps is halved. The extracted features in the contracting path are compressed in the bottleneck of U-Net.

In a conventional U-Net, to up-sample the extracted features from the contracting path, the expanding path is designed with deconvolutional layers. The architecture of the expanding path is designed for dense prediction from coarse

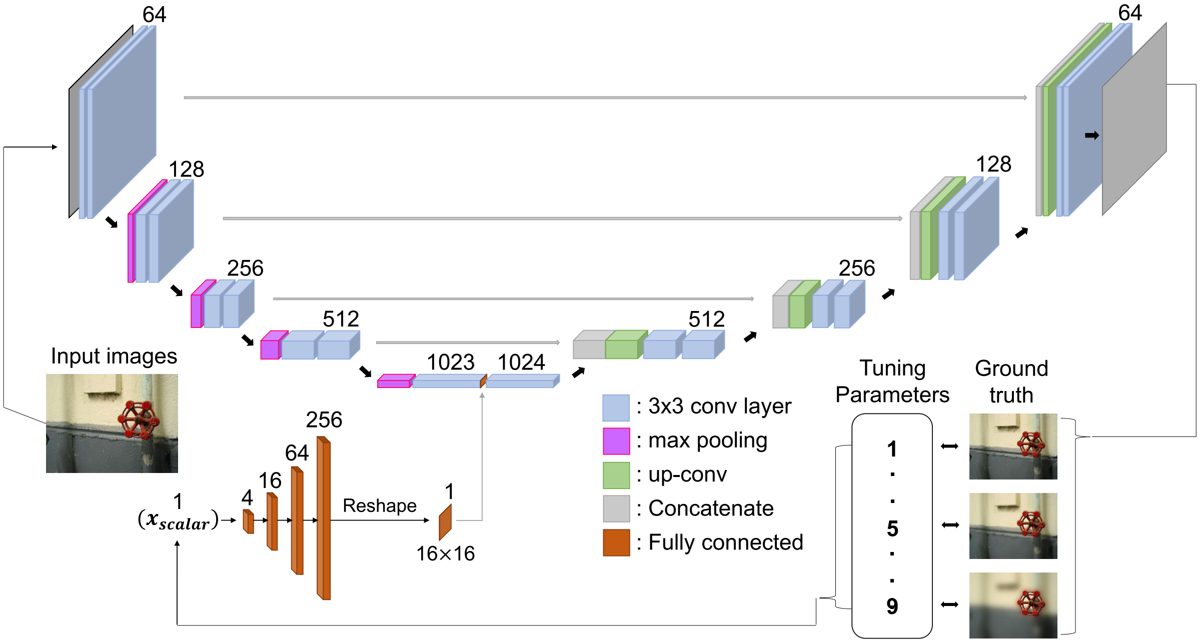


FIGURE 3. The detailed architecture of the proposed model. The proposed model has two inputs; an image and a paired scalar value.

maps. In the last layer of U-Net, a 1 × 1 convolution operation is conducted for generating a 1-channel image.

U-Nets also adopt skip-connections between the contracting path and the expanding path for effective pixel-wise image transformation. Through each up-sampling step, up-sampled feature maps are concatenated with cropped feature maps of the contracting path by using skip-connections.

1) TUNING NETWORK IN TU-NET

The proposed method has the same contracting and expanding path structures as a conventional U-Net. The difference between a conventional U-Net and the proposed TU-Net is the addition of a scalar input parameter, or the *tuning parameter*. The tuning parameter allows for the output of the network to be tuned accordingly. It should be noted that during training, multiple different tuning parameters are trained simultaneously in one TU-Net model. This is opposed to a conventional U-Net which would need to separate trained models for each tuning parameter.

We modify the conventional U-Net at the bottleneck layer with the addition of a *tuning network*, as shown in Fig. 3. The tuning parameter is a scalar value and the output is a feature map that is concatenated inside the U-Net. The tuning parameters are simple scalar values that reflect attribute information of the training data. This means that to change some attribute of the generated output image, different external scalar values can be adjusted without requiring re-training the network.

The proposed tuning network of TU-Net uses a Multi-Layer Perceptron (MLP) to incorporate the tunable parameters into the model. The MLP is organized with fully-connected layers. However, since a conventional U-Net only consists of convolutional layers, the output of the tuning network should be transformed to fit the U-Net. Therefore, we reshape the output of the tuning network from independent nodes into a matrix of the same size of the features maps in the bottleneck layer. By injecting the output of the tuning network with the bottleneck layer, the information from the tuning network is propagated through the expanding path layers to affect the output. This means that we can control the output easily by tuning the features of the bottleneck. To do this, we concatenate the reshaped matrix of the tuning network to the bottleneck of U-Net. In the TU-Net, 1,023 channels are feature maps from standard convolutional filters and 1 channel is the reshaped output of the tuning network. Since the size of the used training images is 256 × 256, the size of the injected matrix is 16 × 16.

IV. EXPERIMENTAL RESULTS

We created new datasets using published datasets to verify the proposed model. Since the proposed model changes the output by utilizing external scalar values, the ground truth images were generated using traditional image processing techniques with corresponding scalar values. We tackle two tasks, image binarization and background blurring. Image binarization is used as a simple target to verify the proposed model's effectiveness as well as test the ability to perform a global image processing technique. On the other hand, background blurring is a complex task that needs to simultaneously identify and ignore the subject of the image and blur the background. We trained the proposed model for conducting these two tasks, and the results were verified quantitatively and qualitatively and compared with the results of a conventional U-Net.

A. DATASETS

1) DATASET FOR IMAGE BINARIZATION

We trained the proposed model to perform image binarization by using the MSCOCO [14] dataset. The MSCOCO dataset is a popular dataset that contains more than 330,000 images. To generate the ground truth images, we used a global fixed threshold. The global fixed threshold binarizes the images by assigning all the pixels brighter than a given threshold to white and the rest to black. Using the global fixed threshold is ideal for the proposed scenario because the threshold can be used as the tuning parameter. To use the global fixed threshold as a tuning parameter, we generated binarized images at nine equidistant thresholds and used the thresholds as the scalar tuning parameters. For our experiment, the threshold values are normalized to integers between 1 and 9. The number of scalar tuning parameters is arbitrary and can be selected for other purposes. We use nine because it is a balance of having enough to not have dramatic steps between parameters and not so many that the differences between them are still present.

Even though we generated nine different sets of binarized ground truth images, we only utilized a portion of the sets for training in order to test the interpolation and extrapolation abilities. There were two versions of the proposed model, *Proposed (3)* that uses three scalar values (3, 5, and 7) and *Proposed (5)* that uses five scalar values (1, 3, 5, 7, and 9). The Proposed (5) model has the full range of threshold values from 1 to 9 but must infer the missing thresholds. Proposed (3) is similar except that it must extrapolate the binarization to the top and bottom tuning parameters.

For the experiments, 1,000, 200, and 100 randomly selected original images from MSCOCO were used for the training set, validation set, and test set, respectively. The original images were then transformed using the previously mentioned global fixed thresholding at the various levels. Thus, Proposed (3) had a total of 3,000 training and 600 validation images, and Proposed (5) had a total of 5,000 training and 1,000 validation images. For testing both models, all nine thresholds were used for a total of 900 images.

2) DATASET FOR IMAGE BACKGROUND BLURRING

To perform image background blurring, the MSRA [15] dataset was used. Similar to the MSCOCO dataset, the MSRA dataset contains natural scene images. However, the reason why the MSRA dataset is used is that it contains segmented ground truth for foreground and background areas. Therefore, we are able to create background blurring ground truth images by utilizing a Gaussian filter on only the background regions. Compared to the global fixed thresholding binarization, background blurring is a complex task that must inherently understand the parts of an image that are background and which are foreground without explicit identification.

Similar to the image binarization task, we used different levels of blurring on the input images to construct the ground truth. To blur the backgrounds, we set Gaussian blur filters to kernel sizes 5, 15, and 45. The tuning parameters are the kernel sizes normalized to 1, 3, and 9, respectively. For the background blurring experiment, the proposed model was trained with 3,000 training samples with 1,000 from each trained blurring level, 600 validation samples, and 900 test samples.

B. EXPERIMENTAL SETUP

Training of the model is similar to that of a conventional U-Net, except with the additional scalar input and a more appropriate loss function. Because our application is image-to-image instead of a segmentation task, we used Mean Absolute Error (MAE) loss instead of Cross-Entropy (CE) loss. The model is trained using Adam optimizer [45] with an initial learning rate of 0.0001. The model was trained with early stopping with patience of 50 epochs of no change and a maximum of 1,000 epochs.

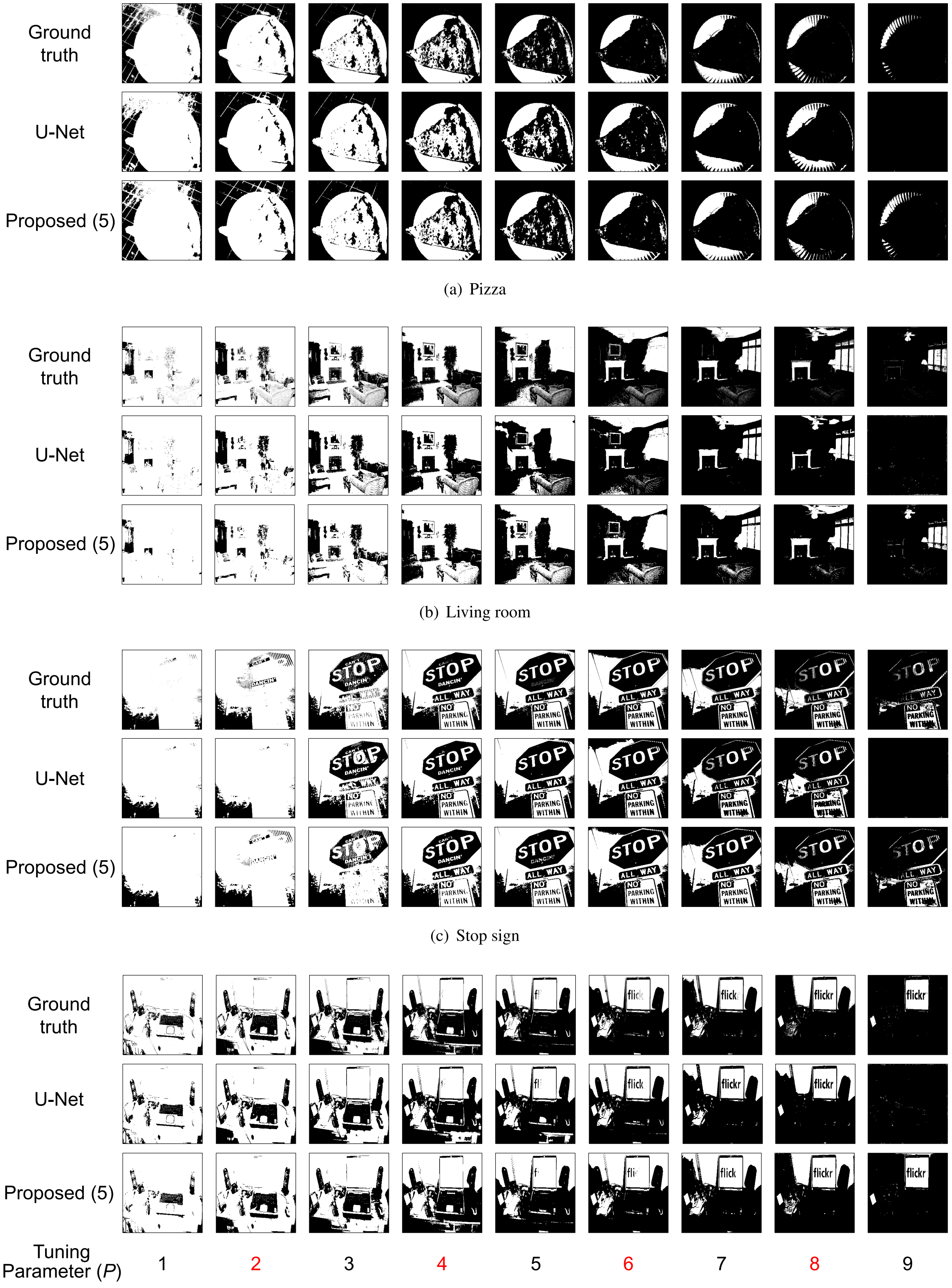
As the comparative method, a conventional U-Net is used. The U-Net has the same structure as the proposed TU-Net except without the tuning network. Also, the comparative U-Net has the same training regimen as the proposed method. Unlike the proposed method, individual independent U-Net models are required for each of the tuning parameters. Thus, nine U-Net models are trained. Also, this means that for the untrained tuning parameters, the U-Net comparisons are granted ground truth that the proposed method does not have access to.

C. BINARIZATION EXPERIMENT

In this section, we evaluate the proposed method by comparing independent U-Nets to the proposed TU-Net on image binarization. We demonstrate various qualitative and quantitative results for an in-depth analysis. First, we show comparative qualitative results that include the results of the conventional U-Net. Next, we compare the results of TU-Net and the conventional U-Net by utilizing quantitative evaluation criteria. The conventional U-Nets, our comparative model, were trained by nine models to generative different nine binarized images.

In Fig. 4, for the qualitative evaluation of the proposed model, we demonstrated the generated result images of the proposed model with five tuning parameters (Proposed (5)). The first row of Fig. 4 is the ground truth images generated by a global fixed thresholding method. The second row is the result images of the conventional U-Net. The third row is the results of Proposed (5). The last row indicates the nine tuning parameters (from $P = 1$ to 9) that are used for output generation. The red tuning parameters ($P = 2, 4, 6, 8$) are non-trained tuning parameters of TU-Net.

As shown in Fig. 4, Proposed (5) can control the output properly and generate well-predicted images in all tuning parameters. In Fig. 4, we demonstrated controlled outputs from $P = 1$ to 9 by using four input images. Even though we conducted just model training once by using TU-Net, the results of Proposed (5) are superior to the results of each nine U-Nets that were trained by each tuning parameter.



(a) Pizza

(b) Living room

(c) Stop sign

(d) Laptop computer

FIGURE 4. **Examples of binarized image results. The red tuning parameters are not explicitly trained by the proposed model.**

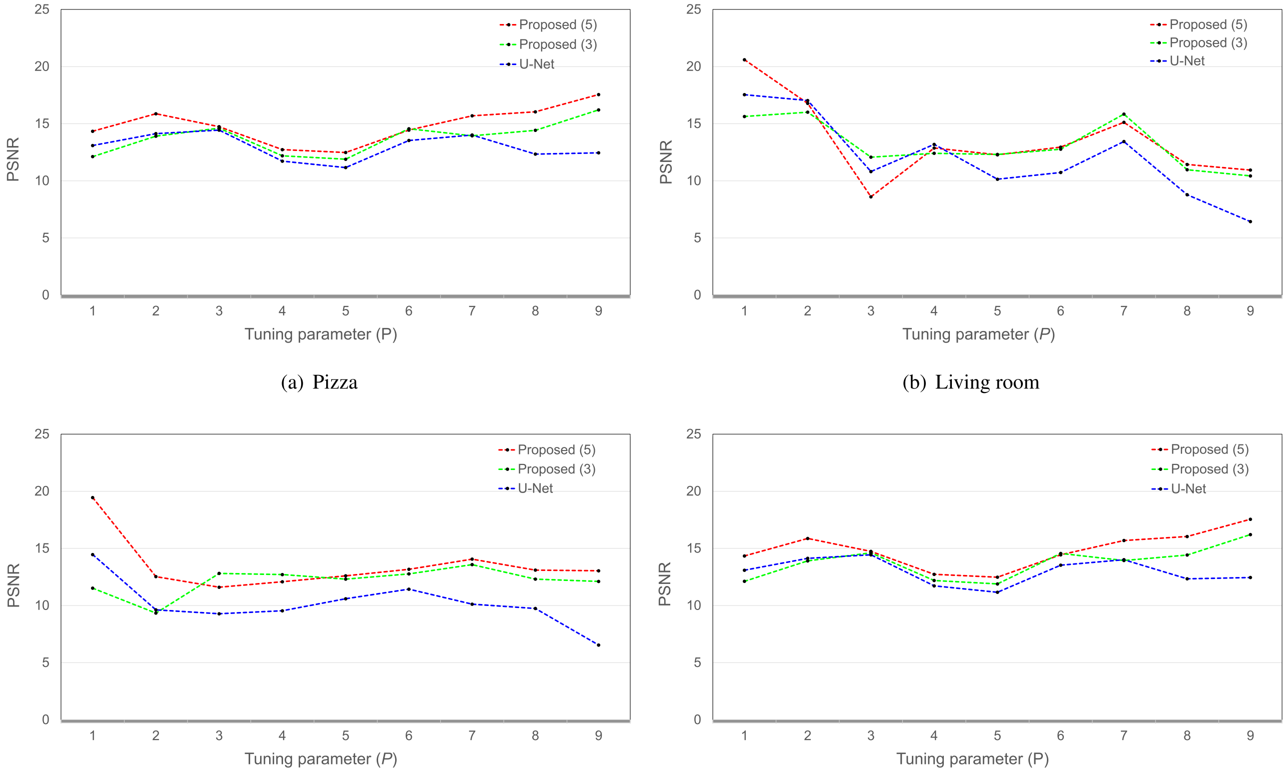
In Fig. 4, we want to discuss two issues why TU-Net is good to generate multiple similar outputs. First, even though the conventional U-Nets were also shown good image binarization from $P = 3$ to 7, the predicted images of the

TABLE 1. Average test results for binarization. *P* is the tuning parameter.

P	Model	Precision	Recall	FM	PSNR	SSIM	NMSE
1*	U-Net	57.22	97.10	72.01	15.83	0.9789	0.7147
	Proposed (5)	**73.47**	**97.65**	**83.85**	**18.91**	**0.9917**	**0.5331**
	Proposed (3)	64.29	96.91	77.30	16.76	0.9861	0.6742
2**	U-Net	70.06	88.91	78.37	12.91	0.9461	0.8519
	Proposed (5)	**80.07**	**98.61**	**88.38**	**16.11**	**0.9820**	**0.6992**
	Proposed (3)	74.41	94.19	83.14	14.89	0.9654	0.7637
3	U-Net	71.71	96.02	82.10	11.63	0.9271	0.9114
	Proposed (5)	**79.84**	**98.27**	**88.10**	**12.47**	**0.9398**	**0.8593**
	Proposed (3)	76.91	97.55	86.01	11.74	0.9295	0.9027
4**	U-Net	78.02	94.08	85.30	9.61	0.8819	1.092
	Proposed (5)	**84.07**	**96.59**	**89.90**	**11.67**	**0.9214**	**0.9058**
	Proposed (3)	82.91	96.01	88.98	10.81	0.9109	0.9571
5	U-Net	82.73	96.68	88.78	9.55	0.8795	1.121
	Proposed (5)	**90.53**	**99.70**	**93.66**	**12.38**	**0.9351**	**0.8661**
	Proposed (3)	89.04	97.13	92.91	11.11	0.9203	0.9128
6**	U-Net	86.95	96.49	91.47	10.61	0.9087	0.9693
	Proposed (5)	**93.71**	**99.07**	**96.32**	**13.46**	**0.9541**	**0.7985**
	Proposed (3)	91.05	98.78	94.76	11.88	0.9304	0.8991
7	U-Net	86.67	98.12	92.04	9.91	0.8985	1.011
	Proposed (5)	**91.72**	**98.19**	**94.84**	**12.75**	**0.9419**	**0.8481**
	Proposed (3)	87.22	98.07	92.33	10.10	0.9012	0.9874
8**	U-Net	88.31	97.42	92.64	11.73	0.9293	0.9030
	Proposed (5)	**95.51**	**99.36**	**97.40**	**15.18**	**0.9705**	**0.7309**
	Proposed (3)	93.20	97.98	95.53	13.39	0.9518	0.8093
9*	U-Net	90.77	98.83	94.63	12.41	0.9381	0.8514
	Proposed (5)	**95.76**	**99.21**	**97.45**	**17.42**	**0.9869**	**0.6219**
	Proposed (3)	92.06	99.10	95.45	16.53	0.9805	0.6891

* Not trained by Proposed (3)

** Not trained by both Proposed (3) and Proposed (5)



(a) Pizza

(b) Living room

(c) Stop sign

(d) Laptop computer

FIGURE 5. Graphs of PSNR at different values of *P*. The graphs from the upper left to the lower right correspond to the images in Fig. 4.

U-Nets are worse than Proposed (5) in $P = 1$, 8, and 9. We thought that the conventional U-Net is only trained for image-to-image transformation like image binarization. However, TU-Net can be trained the degree of changing

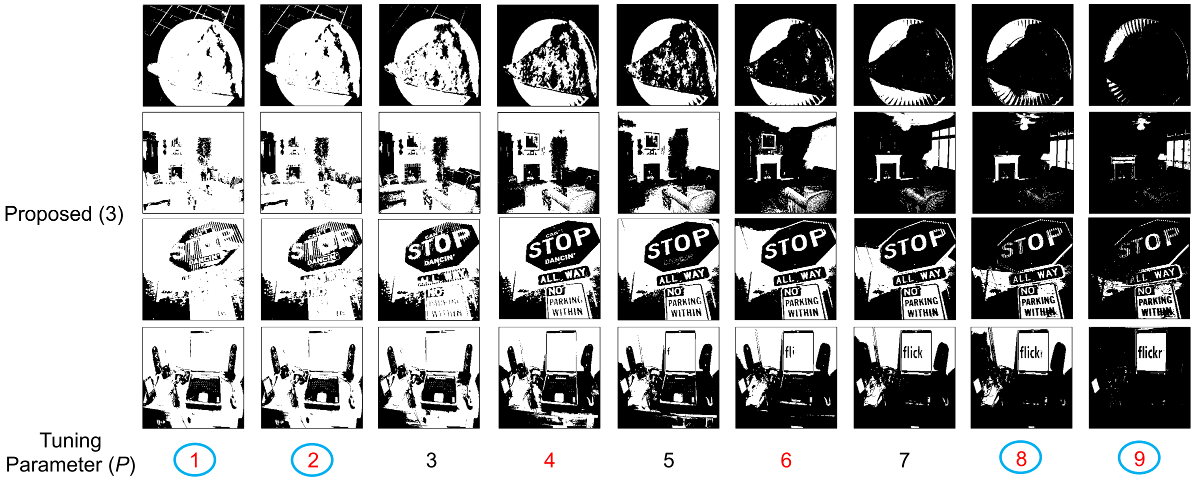


FIGURE 6. Examples from Proposed (3) which show results from tuning parameters external to the range that it was trained with. The red tuning parameters are not explicitly trained by the proposed model. The blue circles of tuning parameters indicate extrapolated tuning parameters.

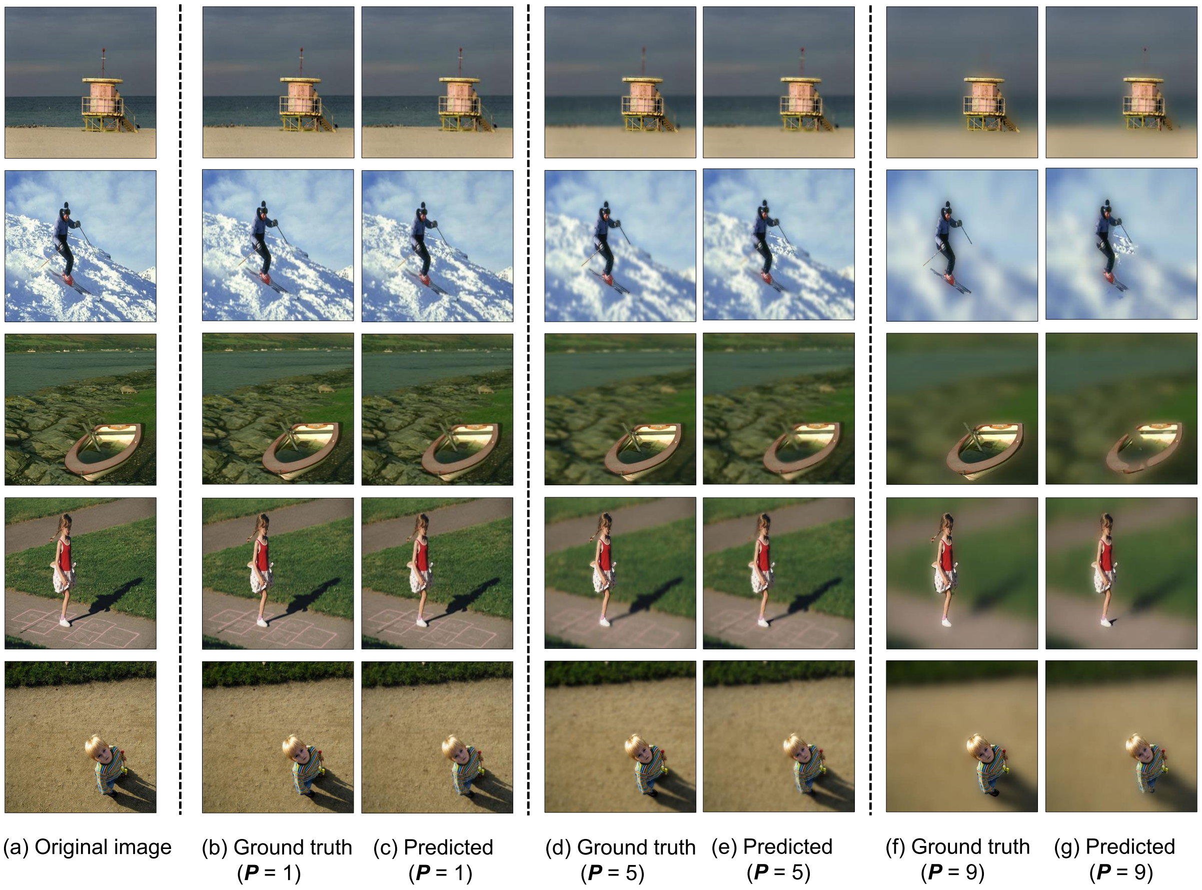

(a) Original image (b) Ground truth (**P** = 1) (c) Predicted (**P** = 1) (d) Ground truth (**P** = 5) (e) Predicted (**P** = 5) (f) Ground truth (**P** = 9) (g) Predicted (**P** = 9)

FIGURE 7. Examples of background blurring results. ***P*** is the scalar tuning parameter.

output as well as image-to-image transformation simultaneously. Therefore, even when it is not easy to perform image conversion, such as P = 1,8, and 9, TU-Net can generate well-binarized images compared with the U-Nets. Second, TU-Net can predict images with the non-trained tuning parameters ($P = 2, 4, 6, 8$). Due to the inference ability of TU-Net, computational cost and memory burden for conducting multiple similar tasks by U-Net-based model can be decreased.

In Fig. 5, for the quantitative evaluation of the proposed model, we plot graphs that include the Peak Signal-to-Noise Ratio (PSNR) values of the conventional U-Net, TU-Net with

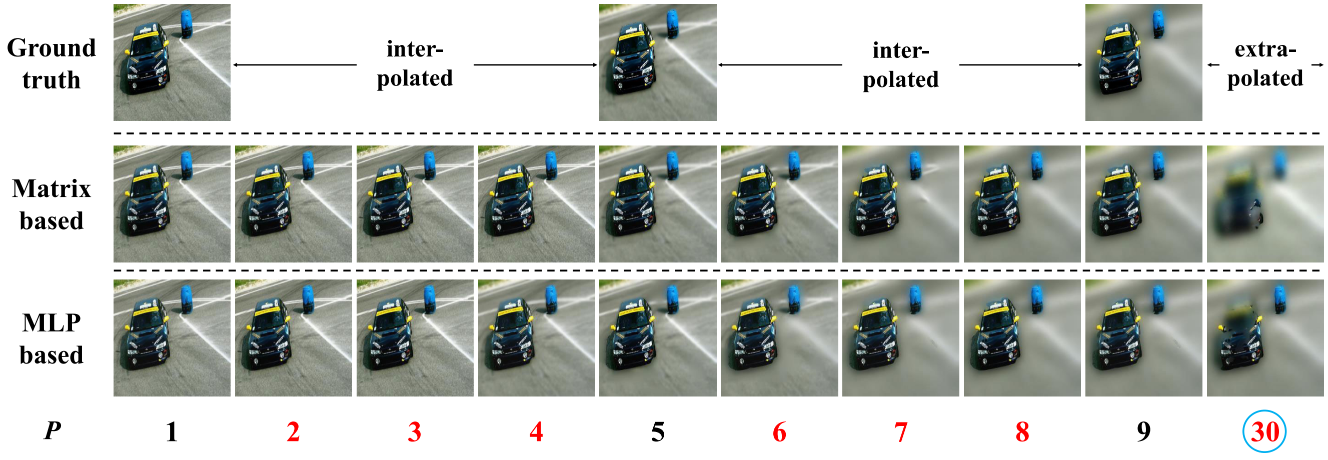


FIGURE 8. Comparative results of background blurring between the MLP-based proposed model and TU-Net with a flat-valued matrix. Images from left to right correspond to the result image of $P = 1, 2, 3, 4, 5, 6, 7, 8, 9$, and 30. The red tuning parameters are not explicitly trained by the proposed model. The blue circles of tuning parameters indicate extrapolated tuning parameters.

three tuning parameters (Proposed (3)) and Proposed (5). Since the PSNR measurement criterion can calculate the degree of image distortion by the Mean Square Error (MSE), the PSNR is still utilized with other measurement criteria in an evaluation of document image binarization [46].

In Fig. 5, Proposed (3) and Proposed (5) achieve better PSNR results in almost all tuning parameters compared with the U-Nets. According to Fig. 5, we recognized two things in the proposed model. First, several PSNR results of Proposed (3) are good as well as the results of Proposed (5). It means even if we train TU-Net with fewer tuning parameters, the trained TU-Net can generate well-predicted images. Second, The proposed model (Proposed (3) and (5)) achieved a huge gap of PSNR values in $P = 1$, 8, and 9 compared with the results of the U-Net. These quantitative results prove that TU-Net can generate well-predicted results. According to these two things, the proposed model can control outputs effectively in all tuning parameters.

The average test results for each tuning parameter P are shown in Table 1. The table has results for six measures, Precision, Recall, F-measure (FM), Structural Similarity Index Measure (SSIM), Normalized MSE (NMSE) as well as PSNR. The quantitative results were calculated by utilizing 100 test images. Compared with U-Net, Proposed (5) achieves better results for all measures and at all values of P despite not being explicitly trained at $P = 2$, 4, 6, and 8. Furthermore, Proposed (3) also achieves better results in all P for the most part compared with U-Net. In addition, we can confirm that Proposed (3) achieves the robust performance of output control and image generation in the extrapolated tuning parameters ($P = 1, 9$). This indicates that the proposed method is able to accurately infer and interpolate the missing P values, although the missing P values are extrapolated values. In Fig. 6, the result images of Proposed (3) prove that TU-Net can predict outputs in the extrapolated range of the tuning parameters as well as the interpolated range of the tuning parameters.

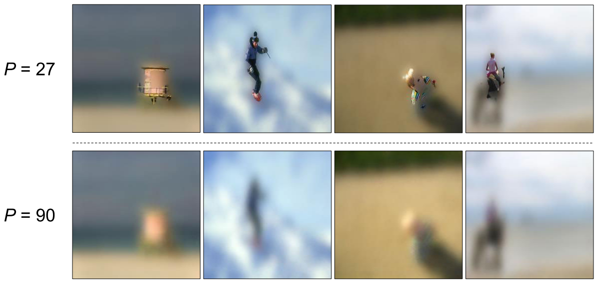


FIGURE 9. The result images by injecting an external range of tuning parameters used for the model training.

D. BACKGROUND BLURRING EXPERIMENT

Since the goal of image background blurring is to blur background areas and to preserve foreground areas simultaneously, before blurring the background areas of an image, detecting the foreground areas is important. In [26], Yan *et al.* proposed an image processing-based blurring system that has several techniques for segmenting foreground and background areas such as Lazy Snapping [47], face detection, and depth map generation method. Therefore, if researchers want to design a CNN-based image background blurring system, they should design two CNN models for segmentation and background blurring at least. In addition, bokeh effect in a photography can control the degree of blurring background areas using aperture sizes. Therefore, if CNN-based image background blurring systems want to generate various types of blurred outputs at different aperture sizes, several pre-trained CNN-based systems are required. That is why the proposed model is suggested for generating multiple blurred outputs.

In this section, we trained the proposed model by using three scalar values for background blurring in Fig. 7. When we trained the proposed model, we used only three scalar values ($P = 1, 5, 9$) since we confirmed that the proposed model with three scalar values also shows robust output tuning

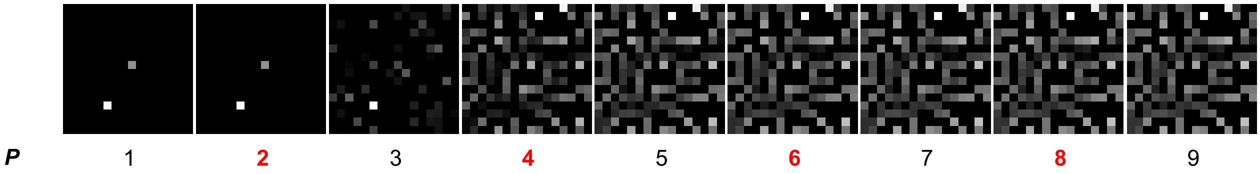


FIGURE 10. The output of the tuning network at different values of *P*. The red tuning parameters are not explicitly trained by the proposed model.

performance in Fig. 6. Therefore, except for these three scalar values, used other scalar values in Fig. 8 is created. However, these created scalar values do not get out of the scalar values used for the training. We can call these results interpolated results.

Fig. 7 illustrated comparative results between the predicted and ground truth images. In Fig. 7, we confirmed that the proposed model blurs the image's background areas and detects a proper foreground area in the image at the same time. Also, according to the third and fifth row's images of Fig. 7, the proposed model can conduct background blurring with non-central foreground images. It means that the proposed model is not biased or process background blurring in the middle of an image. In Fig. 8, the results of the U-Net and the proposed model on images with multiple foreground objects. The proposed model shows clear boundaries between the foreground and background. According to Fig. 7 and Fig. 8, we can recognize two strengths of the proposed model, first is controlling successive blurring steps by using only scalar values, and the second is detecting the real foreground targets in an image even though there are two foreground areas in an image. Also, by increasing scalar values in the proposed model, the relatively stronger target (foreground) still remains.

E. WHY WE USE A MLP-BASED TUNING NETWORK IN TU-NET

In this section, we demonstrate the results of the proposed model by modifying the input part. We demonstrate the modified TU-Net by injecting a matrix instead of the MLPs. When we designed the proposed model, we considered that used MLPs in the proposed model are biased in the middle-side of the input image. According to Fig. 7, and 8, even though we can check this fact by using non-central targets of images, we should verify our proposed model by using matrices.

Since the matrices filled by scalar values are not only the same size as the output of an MLP-based tuning network but also consist of exceedingly few parameters compared with an MLP-based tuning network, we thought that using a matrix in TU-Net can be efficient compared with using an MLP.

In Fig. 8, the comparative results between the proposed model and the modified TU-Net by injecting a matrix instead of an MLP. Fig. 8 demonstrates the background blurring results of the proposed model by using scalar values from $P = 1$ to 9 and 30. Since the proposed model was trained by using three scalar values ($P = 1, 5, 9$), the results ($P = 2, 3, 4, 6, 7, 8$) are interpolated predicted results. Even though TU-Net with a flat-values matrix also can control outputs as well as MLP-based TU-Net, the successive output change of TU-Net with a flat-values matrix is worse than MLP-based TU-Net. Especially, the extrapolated predicted results of MLP-based TU-Net can detect multiple foreground areas and blur background areas compared with TU-Net with a flat-values matrix.

To evaluate the performance of predicting extrapolated values in TU-Net, we inject an external range of the scalar values as shown in Fig. 9. Fig. 9 shows the result images of $P = 30$ and 90. We also injected the smaller values of the minimum value of used scalar values for model training ($P = 0.1, 0.5$). We could not find a big difference compared with the result image by using $P = 1$. According to Fig. 9, even though we did not consider cases of using an external range of used scalar values when we train the proposed model, the proposed model's outputs can be controlled by using the external range of used scalar values. In addition, by increasing scalar values, the degree of blurring effect in the images is increased.

Fig. 10 demonstrates the outputs of an MLP-based and a flat-values matrix. We can confirm that an MLP-based tuning network can control the output of TU-Net compared with a flat-values matrix. As shown in Fig. 8 and 10, we can confirm that the modified TU-Net cannot predict well when the modified TU-Net uses created scalar values compared with the proposed model.

According to this experiment, we confirmed two things. First is that the proposed TU-Net shows robust performance compared with TU-Net with a flat-values matrix. It means the predicted results of the proposed model are not biased in the middle from the MLPs. Second, even though the modified TU-Net shows better performance, in the prediction ability to use non-trained scalar values, the proposed model shows much better performance since MLPs can contain lots of information compared with matrices.

V. CONCLUSION

In this study, we proposed Tunable U-Net that can control the output by utilizing scalar values. In the proposed model, for controlling an output of the U-Net, an MLP-based tuning network is suggested. The proposed model shows good performance in image binarization and image background blurring by using the MSCOCO and MSRA datasets compared with conventional U-Nets. According to these evaluation results, the proposed model not only shows robust performance in trained tuning parameters but also shows well-predicted outputs in untrained tuning. Moreover, the proposed model shows the extrapolated results by injecting an external range of used scalar values for mode training. Furthermore, according to the results of the proposed model by using a matrix input instead of the proposed tuning network, we showed that the MLP is essential.

In the future, we hope to utilize the proposed model in various research fields. It has a wide range of applications, especially for computer vision tasks that require a manual parameter. In addition, in the future, we will pursue the idea of training Tunable U-Nets that perform multiple tunable tasks simultaneously.

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, Nov. 1998.

[2] S. Harada, H. Hayashi, and S. Uchida, "Biosignal generation and latent variable analysis with recurrent generative adversarial networks," *IEEE Access*, vol. 7, pp. 144292–144302, 2019.

[3] G. Atarsaikhan, B. K. Iwana, and S. Uchida, "Guided neural style transfer for shape stylization," *PLoS ONE*, vol. 15, no. 6, Jun. 2020, Art. no. e0233489.

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

[5] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 565–571.

[6] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2016* (Lecture Notes in Computer Science), vol. 9901, S. Ourselin, L. Joskowicz, M. Sabuncu, G. Unal, and W. Wells, Eds., Athens, Greece. Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-46723-8_49.

[7] E. Ahmadi, Z. Azimifar, M. Shams, M. Famouri, and M. J. Shafiee, "Document image binarization using a discriminative structural classifier," *Pattern Recognit. Lett.*, vol. 63, pp. 36–42, Oct. 2015.

[8] A. Cohen-Hadria, A. Roebel, and G. Peeters, "Improving singing voice separation using deep U-Net and wave-U-Net with data augmentation," in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2019, pp. 1–5.

[9] 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)*, Jun. 2020, pp. 364–365.

[10] P. Esser, E. Sutter, and B. Ommer, "A variational U-Net for conditional appearance and shape generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8857–8866.

[11] 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.*, 2014, pp. 2672–2680.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

[13] E. Schonfeld, B. Schiele, and A. Khoreva, "A U-Net based discriminator for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8207–8216.

[14] T. Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Computer Vision—ECCV 2014* (Lecture Notes in Computer Science), vol. 8693, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, doi: 10.1007/978-3-319-10602-1_48.

[15] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 815–828, Apr. 2019.

[16] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[17] W. Niblack, *An Introduction to Digital Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Jan. 1986.

[18] A. S. Abutaleb, "Automatic thresholding of gray-level pictures using two-dimensional entropy," *Comput. Vis., Graph., Image Process.*, vol. 47, no. 1, pp. 22–32, 1989.

[19] C. Tensmeyer and T. Martinez, "Document image binarization with fully convolutional neural networks," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 99–104.

[20] S. Kang, B. K. Iwana, and S. Uchida, "Complex image processing with less data—Document image binarization by integrating multiple pre-trained U-Net modules," *Pattern Recognit.*, vol. 109, Jan. 2021, Art. no. 107577.

[21] S. He and L. Schomaker, "DeepOtsu: Document enhancement and binarization using iterative deep learning," *Pattern Recognit.*, vol. 91, pp. 379–390, Jul. 2019.

[22] Q. N. Vo, S. H. Kim, H. J. Yang, and G. Lee, "Binarization of degraded document images based on hierarchical deep supervised network," *Pattern Recognit.*, vol. 74, pp. 568–586, Feb. 2018.

[23] R. De, A. Chakraborty, and R. Sarkar, "Document image binarization using dual discriminator generative adversarial networks," *IEEE Signal Process. Lett.*, vol. 27, pp. 1090–1094, 2020.

[24] A. K. Bhunia, A. K. Bhunia, A. Sain, and P. P. Roy, "Improving document binarization via adversarial noise-texture augmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2721–2725.

[25] J. Zhao, C. Shi, F. Jia, Y. Wang, and B. Xiao, "Document image binarization with cascaded generators of conditional generative adversarial networks," *Pattern Recognit.*, vol. 96, Dec. 2019, Art. no. 106968.

[26] C.-Y. Yan, M.-C. Tien, and J.-L. Wu, "Interactive background blurring," in *Proc. 17th ACM Int. Conf. Multimedia (MM)*, Oct. 2009, pp. 817–820.

[27] J. Wu, C. Zheng, X. Hu, and G. Ouyang, "Realistic rendering of bokeh effects," *J. Comput.-Aided Design Comput. Graph.*, vol. 22, no. 5, pp. 746–752, Jun. 2010.

[28] J. Moersch and H. Hamilton, "Variable-sized, circular bokeh depth of field effects," in *Proceedings of Graphics Interface 2014*, Montréal, QC, Canada. Toronto, ON, Canada: Canadian Human-Computer Communications Society, 2014, pp. 103–107.

[29] X. Shen, X. Tao, H. Gao, C. Zhou, and J. Jia, "Deep automatic portrait matting," in *Computer Vision—ECCV 2016* (Lecture Notes in Computer Science), vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-46448-0_6.

[30] N. Wadhwa, R. Garg, D. E. Jacobs, B. E. Feldman, N. Kanazawa, R. Carroll, Y. Movshovitz-Attias, J. T. Barron, Y. Pritch, and M. Levoy, "Synthetic depth-of-field with a single-camera mobile phone," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–13, Aug. 2018.

[31] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

[32] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[33] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789–8797.

[34] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn.*, Jul. 2017, pp. 1857–1865.

[35] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2337–2346.

[36] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

[37] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8798–8807.

[38] C. Chan, S. Ginosar, T. Zhou, and A. Efros, "Everybody dance now," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5933–5942.

[39] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe, "Deformable GANs for pose-based human image generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3408–3416.

[40] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Represent.*, 2014, pp. 1–14.

[41] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3483–3491.

[42] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, ''CVAE-GAN: Fine-grained image generation through asymmetric training,'' in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2745–2754.

[43] C. Lassner, G. Pons-Moll, and P. V. Gehler, ''A generative model of people in clothing,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 853–862.

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

[45] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA, May 2015. [Online]. Available: http://arxiv.org/abs/1412.6980 and https://dblp.org/rec/journals/corr/KingmaB14.bib

[46] I. Pratikakis, K. Zagori, P. Kaddas, and B. Gatos, ''ICFHR 2018 competition on handwritten document image binarization (H-DIBCO 2018),'' in *Proc. 16th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Aug. 2018, pp. 489–493.

[47] Y. Li, J. Sun, C. Tang, and H. Shum, ''Lazy snapping,'' *ACM Trans. Graph.*, vol. 23, no. 3, pp. 303–308, Aug. 2004.

SEOKJUN KANG (Member, IEEE) received the B.S. and M.S. degrees from Kyungpook National University, Daegu, South Korea, in 2016 and 2018, respectively, and the Ph.D. degree from Kyushu University, Fukuoka, Japan, in 2021. He is currently working as a Senior Researcher for MANDO Corporation. His research interests include image processing, pattern recognition, and neural networks in embedded systems.

SEIICHI UCHIDA (Member, IEEE) received the B.E., M.E., and Dr.Eng. degrees from Kyushu University, in 1990, 1992, and 1999, respectively. From 1992 to 1996, he joined SECOM Company Ltd., Japan. He is currently a Professor with Kyushu University. His research interests include pattern recognition and image processing. He is a member of IEICE and IPSJ. He received the 2007 IAPR/ICDAR Best Paper Award, the 2010 ICFHR Best Paper Award, and many domestic awards. He is an Associate Editor of *Pattern Recognition*.

BRIAN KENJI IWANA (Member, IEEE) received the B.S. degree from the University of California at Irvine, Irvine, CA, USA, in 2005, and the Ph.D. degree from Kyushu University, Japan, in 2018. He is currently working as an Associate Professor with the Department of Advanced Information Technology, Kyushu University. His research interests include pattern recognition, time series, and neural networks.