J. KSIAM Vol.25, No.2, 26–38, 2021 http://dx.doi.org/10.12941/jksiam.2021.25.026

GRAYSCALE IMAGE COLORIZATION USING A CONVOLUTIONAL NEURAL NETWORK

MINJE JWA[1] AND MYUNGJOO KANG [2,†]

[1]DEPARTMENT OF COMPUTATIONAL SCIENCE AND TECHNOLOGY, SEOUL NATIONAL UNIVERSITY, SEOUL, 08826, REPUBLIC OF KOREA

[2]DEPARTMENT OF MATHEMATICAL SCIENCES, SEOUL NATIONAL UNIVERSITY, SEOUL, 08826, REPUBLIC OF KOREA

Email address: †mkang@snu.ac.kr

ABSTRACT. Image coloration refers to adding plausible colors to a grayscale image or video. Image coloration has been used in many modern fields, including restoring old photographs, as well as reducing the time spent painting cartoons. In this paper, a method is proposed for colorizing grayscale images using a convolutional neural network. We propose an encoder-decoder model, adapting FusionNet to our purpose. A proper loss function is defined instead of the MSE loss function to suit the purpose of coloring. The proposed model was verified using the ImageNet dataset. We quantitatively compared several colorization models with ours, using the peak signal-to-noise ratio (PSNR) metric. In addition, to qualitatively evaluate the results, our model was applied to images in the test dataset and compared to images applied to various other models. Finally, we applied our model to a selection of old black and white photographs.

1. INTRODUCTION

For a long time, colorizing has been achieved by through direct painting; however, this requires significant time and cost. Nevertheless, people have continued to color greyscale images to capture the vividness of many old historical photographs. As more computational techniques have been developed, methods have been devised for automatic painting via computers. Image colorization techniques refer to techniques that add colors (RGB) to grayscale images or videos. When users see grayscale images, they can estimate the colors from the known information. However, it is impossible for humans to find the perfect color. Colorization is an ill-posed problem and does not offer a unique solution and thus focuses on finding plausible colors rather than accurate colors.

Grayscale image coloring is an important task in many areas, such as the film industry, historical data review, and photography technology. In particular, as the mass production of

Received by the editors May 31 2021; Accepted June 24 2021; Published online June 25 2021.
2000 *Mathematics Subject Classification.* 93B05.
Key words and phrases. ImageNet, convolutional neural network, image colorization, FusionNet.
† Corresponding author.

webtoons increases around the world, the field of automatic cartoon coloring has drawn increased attention [25]. Significant progress has been made in this area but automatic image coloring remains a challenge.

There are three methodologies in the colorization field, i.e., scribble-based, exemplar-based, and fully automatic. The first and second methods are classified as user-guided edit propagation methods and the third is a data-driven automatic colorization method. Deep learning significantly progressed over the past decade, especially in the field of image processing, where very good results are being obtained via convolutional neural networks CNNs). These techniques have recently been applied to colorizing methods and the results are shown to be very good. Prior to the emergence of deep learning techniques, the most effective methods relied on human intervention.

In this paper, we propose a colorization technique using a fully automatic deep learning method. Most colorization methods utilize a structure called the encoder-decoder model; for this, we have modified FusionNet [18]. Owing to the poor performance of the model using the underlying MSE loss function, training was conducted by defining a novel loss function suitable for the coloring objective. We used the ImageNet dataset for training and validation and tested datasets of various images using the training model. The performance of the proposed model was quantitatively and qualitatively verified through various experiments.

The remainder of this paper is organized as follows. In Section 2, we review recent trends and methods in research related to colorizing. Section 3 provides detailed information about the proposed model and the loss function. The quantitative and qualitative results of the various experiments are explained in Section 4. Finally, we conclude the results and discuss further work in Section 5.

2. Related Work

The grayscale image coloring field is commonly divided into three categories: scribble-based, exemplar-based, and fully automatic methods. The first and second methods require human intervention and are called user-induced editing propagation methods. An end-to-end automatic method that minimizes human intervention is required. A fully automatic method is introduced, which produces a color image that fits the grayscale image end-to-end, without human intervention for any image.

In the scribble-based method, users provide colored scribbles, giving the model color-based hints. Scribble-based methods are interactive methods for coloring grayscale images by placing doodles. The input value that corresponds to the input image is the appropriate colored doodle drawing or point of color in the grayscale image. Levin et al. [15] introduced this method, which assumes that pixels adjacent to the space containing pixels with color information should have similar colors. In addition, other related methods have been proposed [9, 24, 16]. Zhang et al. [27] developed a model that randomly selects several pixels from a color image and assembles them as scribble input. Xiao et al. [23] developed the method of Zhang et al. [27] and created a model that used the overall color theme of an image as a global input. However, scribble-based methods are less efficient because they require a large amount of direct user

input and selecting color schemes involves human subjectivity, allowing users to choose the wrong color.

In the exemplar-based method, the grayscale and color images are used as inputs; the color image is an example image, with a color that is expected to appear as a result of a grayscale image. This image is called an exemplar or reference image. Welsh et al. [22] developed a model that combines exemplar images and transforms colors by matching the statistics within the region adjacent to the input color image pixel. Kang et al. [13] proposed an improved method that uses an attention structure and webtoon images as a dataset. Various other methods have been devised [20, 5, 3]. These methods have significantly reduced user intervention compared to previous methods but are less efficient because they always require a reference image that is similar to the original color image of the grayscale image and require many images to be processed.

Fully automatic methods are end-to-end deep learning-based coloring methods and are widely used today. In particular, there are several methods for creating models based on CNNs that receive only grayscale images as inputs. Therefore, it is called data-driven automatic colorization. However, as mentioned earlier, this method has no information (scribbles or exemplar images) regarding the desired colors; thus, it is likely to choose the wrong color. Cheng et al. [4] first created a fully automatic colorization method that uses a deep neural network. By training large amounts of data, people can plausible colors can be predicted for grayscale images. These models are fundamentally based on the structure of encoder–decoder models because they need to extract the features of the grayscale image and reconstruct the color using this information; i.e., a similar structure to encoder–decoder models. Iizuka et al. [10] trained local and global inputs together to provide additional class information. Following Iizuka et al. [10], Baldassarre et al. [2] trained semantic information by providing classification classes. Zhao et al. [28] used semantic segmentation information to find the color. Zhang et al. [26] developed a very creative method for learning a particular label for all pixels while using a basic structure model.

Several recent methods for coloring grayscale images using generative adversarial networks (GANs) [7] have emerged. Isola et al. [12] used a U-net-based generator with conditional GAN. In addition, Nazeri et al. [17] used a similar method to that used by Isola et al. [12]. Recently, Vitoria et al. [21] discussed methods for using improved WGANs while providing class information, such as that by Iizuka et al. [10]. Variational autoencoders (VAE) [14] were used to learn color embedding in Deshpande et al. [6].

New models are emerging that use more effective inputs and differ from existing methods. One of these methods is influenced by natural language processing (NLP), whereby the user enters textual information regarding the color of the ground truth image [1]. Changqing et al. [29] created a new dataset [31] that directly contains text information and a model that allows users to enter text directly with sketch information. [30] developed the model further.

3. PROPOSED METHOD

3.1. **Method Overview.** To explain the detailed method for colorizing grayscale images with the proposed model, given a grayscale image L, the proposed model produces $aandb$ values, which are color channels in the CIE Lab color space. The proposed model receives the input image (grayscale image), passes through a feature extraction part (encoding part), a reconstruction part (decoding part), and finally creates two color channels. We selected FusionNet [18] as the encoder–decoder model, with several modifications. Furthermore, the new loss function was applied to infer the point estimates of the colors from the predicted color distribution.

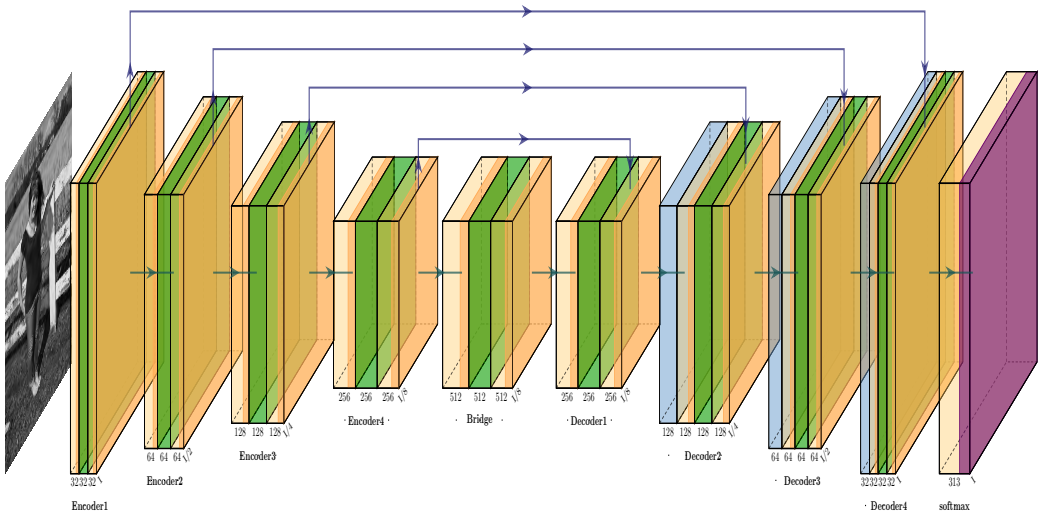


FIGURE 1. Proposed Architecture in this paper

3.2. **Loss Function.** We define a new loss function for the coloring task; i.e., for input grayscale data $X \in \mathbb{R}^{h\times w\times 1}$, the model has to find the a, b channels, such as $\hat{Y} = \mathcal{G}(X) \in \mathbb{R}^{h\times w\times 2}$. The most basic loss function is the mean square error (MSE), which is an image-to-image regression approach. The MSE loss function equation can be written as follows:

$$L_{MSE} = \frac{1}{2}\sum_{h,w} \|Y - \hat{Y}\|_2^2.$$

The optimal solution of the MSE is the average set for the image. When applied, the resulting values (i.e., the values a and b) are almost intermediate and thus do not represent the vivid saturation found in nature. Therefore, we obtain a grayish and awkward color.

Hence, we use a method to find a specific label for each output pixel and compare it with the original label for that pixel. This method was first proposed for colorful image colorization

by Zhang et al. [26]. In terms of multinomial classification, a multinomial cross-entropy loss function can be used. The model must learn the mapping $\hat{Z} = \mathcal{F}(X)$, where $\hat{Z} \in \mathbb{R}^{h \times w \times n}$ and $n = 313$. To compare the predicted $\hat{Z}$ against the ground truth Z, the ground truth color Y is converted to Z.

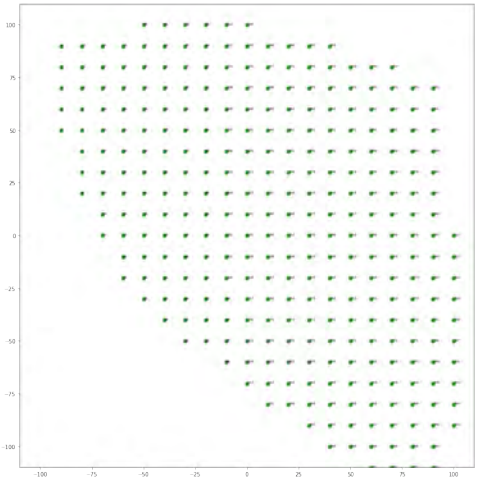


FIGURE 2. Quantized bins of 313 representative points in CIE Lab Color space for values of a and b

To obtain this point, we quantize 313 representative points of channels a and b, as shown in Figure 2. For a particular pixel's a and b values, the nearest representative point becomes the label of that pixel. Using the k-nearest neighbor method, each pixel in the ground truth image Y can be encoded as a vector that contains the label of the classification problem. Instead of a one-hot vector, we found five quantized bins in the order close to each pixel; i.e., the 5-nearest neighbors are found and, using a Gaussian kernel with $\sigma = 5$, are weighted proportionally to their distance from the ground truth. When this method is applied to each pixel, a vector of $h \times w$ dimensions is generated. This allows the cross-entropy loss function to be applied between vectors from the output of the model and vectors from each pixel of the ground truth. By adding all the loss values of each pixel, we specify the complete loss function.

Multinomial cross-entropy loss function, defined in Eq.(3.1).

$$L_{cl} = -\sum_{h,w} \sum_{n} Z_{h,w,n} \log(\hat{Z}_{h,w,n}). \tag{3.1}$$

3.3. **Architecture detail.** In the CIE lab color space, the grayscale lightness information is the L value and the representation of chrominance information is the a and b values. A detailed description of the proposed model, which takes L as the input and infers the values of a, b, is as follows: the proposed model is shown in Figure 1. The architecture comprises three parts

(encoder, decoder, and bridge), containing four encoding blocks, four decoding blocks, and a middle bridge.

As mentioned earlier, the proposed model was modified from FusionNet [18]. Unlike the existing FusionNet model, the input/output size was set to 256×256. If the input/output size is too large, such as FusionNet, the training is too long to handle a large number of images. To resize images, we used bicubic linear interpolation to use the lowest-order interpolation method to connect pixel values at adjacent grid boundaries smoothly. The number of downscaling and upscaling of the proposed model was reduced by one compared to FusionNet. Thus, the model reduces the feature map size from 256×256 to 32×32; this is because the input size of our model is smaller than that of FusionNet and thus much information can be lost if the feature map size is further reduced. Furthermore, we used a 2-stride convolution layer to reduce the size of the feature map while max-pooling is used in FusionNet.

ResBlock (Residual Block) placed between the encoding and decoding blocks is the same as that of FusionNet. It was first introduced by He et al. [8]. ResBlocks can take various forms. A detailed description of the block used in the proposed architecture is shown in Figure 3. It is designed to create a type of shortcut (skip connection) so that gradation can flow well, even if the layer is deep. There is a long skip connection that links information from previous feature maps to future feature maps. It provides the results of adding the first input feature map and the last feature map three times after convolution and batch normalization [11] layers. The filter size of all convolutional layers was 3×3.

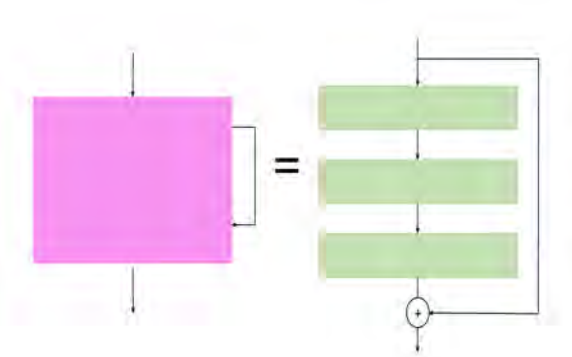


FIGURE 3. Detail description Residual Block from FusionNet [18]

Encoder

The encoder is a feature extraction part and comprises four encoding blocks. To elaborate on the encoding block, the feature map is first reduced by half through a 2-stride convolution. This downscaled feature map then passes through ResBlock (residual block). This process was repeated four times until the size of the feature map was reduced by one-third of the

input image. Finally, the feature map of Resblock passes through the convolutional + batch-normalization layer once more. This final feature map is used as the input for the next encoding block. The ReLU activation function was used for all the convolutional layers in the encoder.

Decoder

The decoder is a reconstruction part and has a structure similar to that of the encoder. The decoder was composed of four decoding blocks. First, the size of the feature map is expanded through a deconvolutional block. After the long skip-connection, which concatenates with the feature map of the same size in the encoding block to create an expanded feature map, the resulting feature map passes sequentially through the convolutional + batch-normalization layer and ResBlock. Resblocks are used in the same structure as those used in the encoding blocks. The feature map enters convolutional + batch normalization once more. This process was repeated four times until the size of the feature map was equal to the size of the input image. However, for the last convolutional layer of these iterations, the output depth was set to 313. Similar to the encoder, all convolution layers use the Relu activation function but, in the last layer, the Softmax activation function is used.

Bridge

The bridge follows similar formats to the encoder and decoder. However, the bridge simply links information of the feature map from the encoder to the decoder and consists of only one block. FusionNet, based on the proposed models, also has an intermediate bridge, reducing the size of the feature map. However, the proposed model receives a feature map from the encoder and sends it to the decoder section at a constant size (the same resolution). After completing the encoder task, the feature map that enters the bridge goes through the same process as the encoding block to create a feature map to send the decoder.

4. Experiment Result

TABLE 1. Experimental environment

Hardware Specification	
CPU	Intel Core i5-6500
GPU	TITAN RTX
Memory Capacity	16GB
Software Specification	
Python	Version : 3.7.6
TensorFlow	Version : 1.13.1
Operation System	Linux Ubuntu 16.04

Various experiments were conducted to validate the architecture proposed in the previous section. We verified the end-to-end model using ImageNet [19] dataset. The environment used in the experiment is presented in Table 1. We implemented the proposed models on an NVIDIA TITAN RTX GPU, using TensorFlow and Keras architecture. We train the proposed

model using 32 of batch size and 100 of epoch size and took approximately 1.5 days on our GPU. The loss function is a multinomial categorical cross-entropy, as described in Section 3.2. We used the SGD as an optimizer.

Furthermore, to objectively compare the proposed architecture in this study, we chose colorful image colorization [26] and deep regularization [2]. These methods are state-of-the-art techniques for fully automatic methods using CNNs. In this study, the peak signal-to-noise ratio (PSNR) was selected as an evaluation indicator to verify the similarity between the ground truth color images and predicted color results.

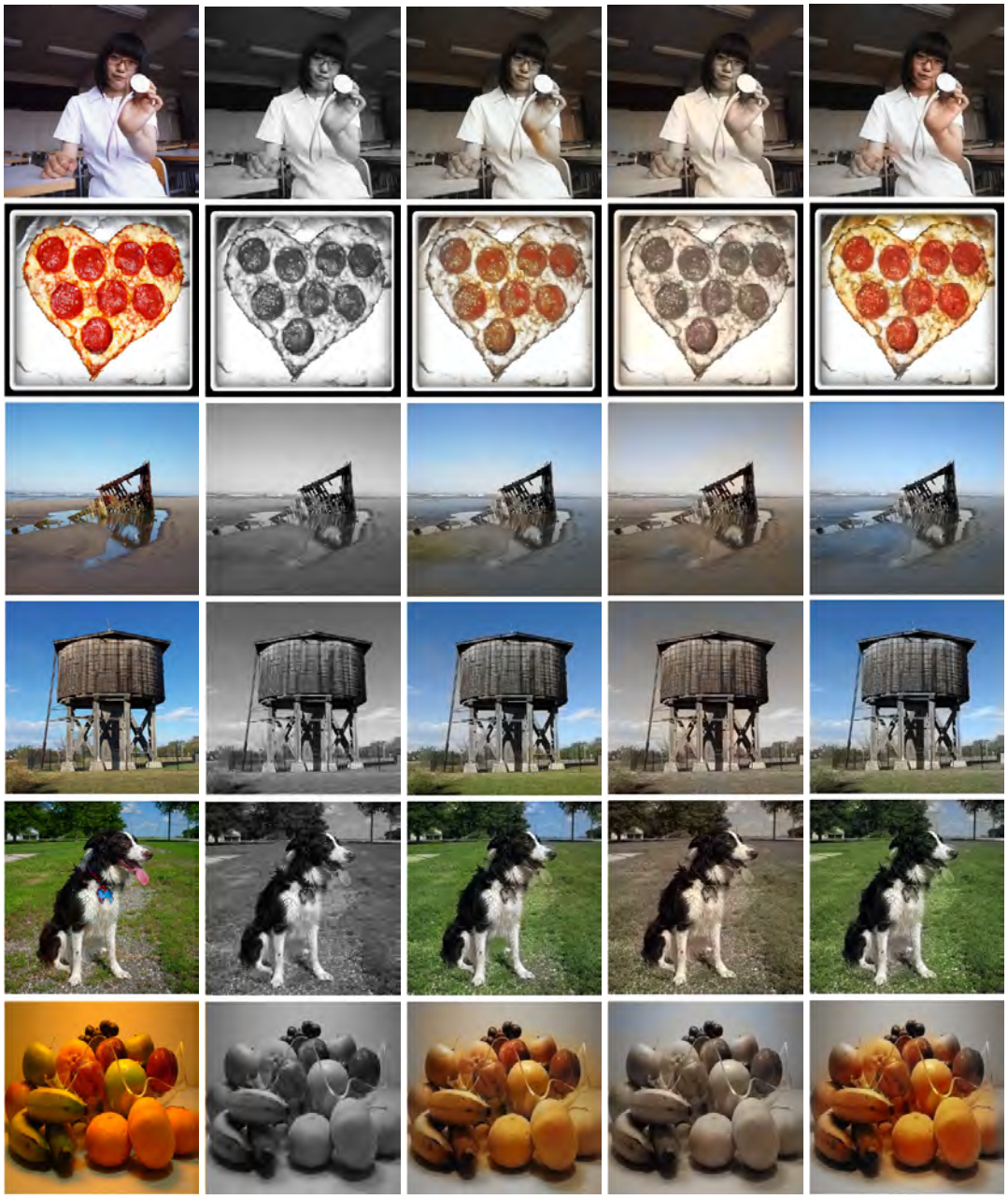
4.1. **Dataset.** ImageNet [19] datasets are already widely used for image processing. ImageNet has approximately 1.3M images with 1000 labels. In other words, the dataset consists of millions of photos within a wide variety of sets and contains various images. Due to the wide variety of images, this dataset is thought to be suitable for coloring tasks. This is because more diverse training images facilitates color prediction for more grayscale images. To simplify running time, we used a small subset of ImageNet (train : validation = 8:2). Thus, 80,000 and 20,000 images were randomly selected from ImageNet for the training and validation datasets, respectively. A description of the used dataset can be found in Table 2.

TABLE 2. Dataset Explanation

Original ImageNet dataset	
Training	1,281,167
Validation	50,000
Test	100,000
Selected dataset for the proposed model	
Training	80,000
Validation	20,000
Test	20,000

4.2. **Qualitative Evaluation.** Zhang et al. [26] and Baldassarre et al. [2] were selected for comparison with the proposed method. The ImageNet dataset was used by Zhang et al. [26] and Baldassarre et al. [2]. However, because we used a subset of the ImageNet dataset, we trained the models directly based on newly selected training and validation datasets for fairness.

Several colorization results are shown in the test dataset in Figure 4. The first column is the ground truth, the second is the input grayscale image, the third is the result of Zhang et al. [26], the fourth is the result of Baldassarre et al. [2], and the final column shows the results of our proposed method. The network of Zhang et al. [26] and the proposed network using a multinomial cross-entropy loss function produced plausible colors. The proposed model produced the clearest colors, generating near-photo-realistic results. However, Baldassarre et al. [2] used the MSE loss function, resulting in relatively desaturated colors. In particular, as seen in the fourth and fifth rows, the proposed model tends to be particularly colorful when the background is blue or green.



(A) Original (B) Grayscale (C) Zhang (D) Baldassarre (E) Proposed

FIGURE 4. Qualitative Comparing results

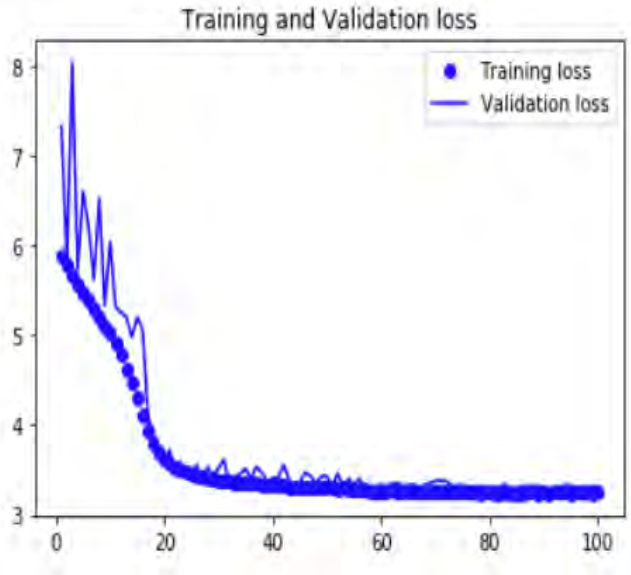


FIGURE 5. Training history on our proposed model

4.3. **Quantitative Evaluation.** To ensure that the proposed model is well trained, we determine the convergence of the loss values. Figure 5 shows the training history. The proposed model converged well for both the training and validation datasets.

Peak signal-to-noise ratio (PSNR) metrics were used for quantitative evaluation. The PSNR is used to evaluate the loss information of the overall quality of the image. This results from calculating the degree of difference between the actual ground truth image and predicted output image pixel-wise. The equation for PSNR is as follows:

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}(\frac{MAX_i^2}{MSE}) \\ &= 20 \cdot \log_{10}(\frac{MAX_i}{\sqrt{MSE}}) \\ &= 20 \cdot \log_{10}(MAX_i) - 10 \cdot \log_{10}(MSE). \end{aligned}$$

The PSNR cannot be a complete indicator of coloring tasks. This is because colorization models aim to find plausible colors and not accurate colors. For example, when we need to find the color of a grayscale flower image and the ground truth is red, colorization models can be estimated to be yellow or blue. In this case, yellow or blue flowers are also plausible colors; hence, we cannot say that the model has decided on a completely wrong color. However, in terms of PSNR, each pixel value between the predicted image and ground truth is very different, which has a significant impact on PSNR. Nevertheless, it can contain important information for finding single pixels that predict different colors from the entire image or pixels that remain

gray. It can capture cases where the model is poorly trained and the results are predicted to be brownish or grayish. Therefore, the PSNR metric was used in the colorizing field.

The average PSNR values of the proposed model and other models using the test dataset are listed in Table 2. As previously mentioned, 20,000 images were randomly selected. The table indicates that the proposed model provides the best results in terms of PSNR among all the other methods in the experiment. The PSNR of Zhang et al. [26] is 23.4657, i.e., lower than the PSNR of the proposed model (23.9219). In particular, Baldassarre et al. [2] had the lowest PSNR values. Baldassarre et al. [2] used the MSE loss function and the output color image was grayish overall compared to other models.

TABLE 3. Comparing PSNR results with other networks.

Model	PSNR
Zhang et al. [26]	23.4657
Baldassarre et al. [2]	21.9914
Proposed method	23.9219

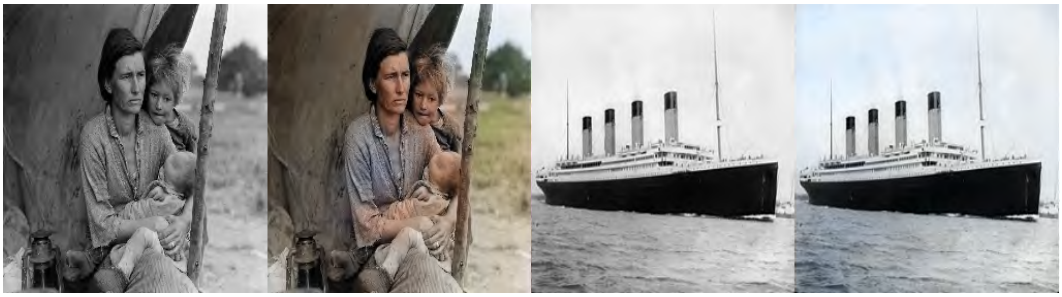


FIGURE 6. Old legacy image colorizing

4.4. **Legacy Old Image Colorizing.** The proposed model was trained based on an ImageNet dataset containing a variety of images; thus, we were able to apply images in a variety of situations. For application to real situations, several old legacy images were searched and tested to see if they could be colored and how natural they were using old black-and-white pictures as input without ground truth. The results are shown in Figure 6.

5. CONCLUSION AND FUTURE WORKS

In this paper, a new fully automatic colorization model using a deep learning CNN network is proposed. It is based on FusionNet, which performs very well as an encoder–decoder model, but with some modifications made. An appropriate loss function was used to achieve good results in colorization tasks. The validity of this model was compared with other existing deep learning colorizing models, both qualitatively and quantitatively. Thus, it was confirmed that the proposed model exhibited better performance with respect to the other models.

ACKNOWLEDGMENTS

Myungjoo Kang was supported by the NRF grant [2015R1A5A1009350][2021R1A2C3010887] and the ICT R&D program of MSIT/IITP[1711117093].

REFERENCES

[1] H. Bahng, S. Yoo, W. Cho, D. Keetae Park, Z. Wu, X. Ma, and J. Choo. Coloring with words: Guiding image colorization through text-based palette generation. In *Proceedings of the european conference on computer vision (eccv)*, pages 431–447, 2018.

[2] F. Baldassarre, D. G. Morín, and L. Rodés-Guirao. Deep koalarization: Image colorization using cnns and inception-resnet-v2. *arXiv preprint arXiv:1712.03400*, 2017.

[3] A. Bugeau, V.-T. Ta, and N. Papadakis. Variational exemplar-based image colorization. *IEEE Transactions on Image Processing*, 23(1):298–307, 2013.

[4] Z. Cheng, Q. Yang, and B. Sheng. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–423, 2015.

[5] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin. Semantic colorization with internet images. *ACM Transactions on Graphics (TOG)*, 30(6):1–8, 2011.

[6] A. Deshpande, J. Lu, M.-C. Yeh, M. Jin Chong, and D. Forsyth. Learning diverse image colorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6837–6845, 2017.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu. An adaptive edge detection based colorization algorithm and its applications. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 351–354, 2005.

[10] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (ToG)*, 35(4):1–11, 2016.

[11] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[13] S. Kang, J. Choo, and J. Chang. Consistent comic colorization with pixel-wise background classification. In *NIPS'17 Workshop on Machine Learning for Creativity and Design*, 2017.

[14] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[15] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*, pages 689–694. 2004.

[16] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum. Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 309–320, 2007.

[17] K. Nazeri, E. Ng, and M. Ebrahimi. Image colorization using generative adversarial networks. In *International conference on articulated motion and deformable objects*, pages 85–94. Springer, 2018.

[18] T. M. Quan, D. G. Hildebrand, and W.-K. Jeong. Fusionnet: A deep fully residual convolutional neural network for image segmentation in connectomics. *arXiv preprint arXiv:1612.05360*, 2016.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[20] Y.-W. Tai, J. Jia, and C.-K. Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 747–754. IEEE, 2005.

[21] P. Vitoria, L. Raad, and C. Ballester. Chromagan: Adversarial picture colorization with semantic class distribution. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2445–2454, 2020.

[22] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 277–280, 2002.

[23] Y. Xiao, P. Zhou, and Y. Zheng. Interactive deep colorization with simultaneous global and local inputs. *arXiv preprint arXiv:1801.09083*, 2018.

[24] L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. *IEEE transactions on image processing*, 15(5):1120–1129, 2006.

[25] S. Yoo, H. Bahng, S. Chung, J. Lee, J. Chang, and J. Choo. Coloring with limited data: Few-shot colorization via memory augmented networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11283–11292, 2019.

[26] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[27] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017.

[28] J. Zhao, J. Han, L. Shao, and C. G. Snoek. Pixelated semantic colorization. *International Journal of Computer Vision*, pages 1–17, 2019.

[29] C. Zou, H. Mo, R. Du, X. Wu, C. Gao, and H. Fu. Lucss: Language-based user-customized colourization of scene sketches. *arXiv preprint arXiv:1808.10544*, 2018.

[30] C. Zou, H. Mo, C. Gao, R. Du, and H. Fu. Language-based colorization of scene sketches. *ACM Transactions on Graphics (TOG)*, 38(6):1–16, 2019.

[31] C. Zou, Q. Yu, R. Du, H. Mo, Y.-Z. Song, T. Xiang, C. Gao, B. Chen, and H. Zhang. Sketchyscene: Richly-annotated scene sketches. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 421–436, 2018.