Colorization With Convolutional Neural Network

Yutaro Miyata
Division of Computer Science
University of Minnesota, Morris
Morris, Minnesota, USA 56267
miyat001@morris.umn.edu

ABSTRACT

Currently, deep learning is used in various fields. The development of image processing has shown a particularly remarkable growth. Within the field of image translation, neural networks are used for many applications such as style conversion, character generation, and black and white image or sketch colorization. This paper focuses on colorization. The pix2pix program uses neural networks for colorization. It is specialized in sketch to image conversion and colorization. This paper explains the basic idea of colorization to illustrate the effectiveness of these technologies.

1. INTRODUCTION

Colorization is a form of image conversion. A neural networks finds a corresponding pattern between an input image and pattern filter which a computer learned from a large amount of data sets. For this process, three networks are important: convolutional neural networks (CNN), a semantic segmentation networks, and a generative adversarial networks (GAN). A CNN has wide applications and is specialized for finding a features from many data sets. A semantic segmentation networks are specialized for finding boundaries between objects and their locations. A GAN is an innovative network that can help colorize more accurately than the other two networks. These three networks are utilized together for automatic colorization. In study by Isola et al [3], they shows how these networks are effective in automatic colorization and image conversion. Study by Iizuka et al [2] show a different approach to coloring black and white image without GAN.

2. BACKGROUND

2.1 Colorization

Automatic colorization is used to add color to images that do not contain colors such as black and white photos. Traditionally, it used to require manual work, but with the development of deep learning, it became possible to add color automatically. Although there are various colorization methods, an early paper by Welsh, et al shows that deep learning can be used to colorize images. [8]

This method colorizes a black and white image based on a reference color image. First, a computer selects a small subset of pixels in the color image as a sample. The pixels are tiny squares which comprise of an image. Next, it checks each pixel of a grayscale image and selects the best matching sample in color image. Finally, the best match pixel color is transferred to a black and white image.

Welsh et al say "the problem of coloring greyscale images has no inherently correct solution. Due to these ambiguities, human interaction usually plays a large role in the colorization process." [8] Therefore, designating a color conversion area by manual work is used to be necessary for getting high accuracy result. The semantic segmentation can make a big contribution on this point. It can accurately detect edges between objects without manual work, so it can reduce ambiguity and precisely separate colors.

Moreover, a neural networks (Described in the next section) can learn the relationship between a luminance image and a color image. A neural networks can extracts features from a color image. In other words, the neural networks can learn which color is used in which place. Once learned, a black and white image can be colorized by using the result from the neural networks.

2.2 Artificial Neural Networks

An Artificial Neural Network (ANN) is a network imitating nerve cells and their connection in a human brain. A neural network consist of an input layer, an output layer, and several hidden layers. An input layer receives data and passes it to next layer, a hidden layer does some calculation and passes results to the next hidden layer or output layer, and output layer produces a result value. Each layer consists of many nodes. Each edge has a weight indicating a strength of a connection between one node and another node. Training adjusts the weights to correct errors between the output result and the desired output. The classification is that classifies data into categories. For training, a network receives a labeled data as training data to understand what an object is. A labeled data is answer data. For instance, if a data item is an image of a cat, its label is a cat. Then the network minimizes an error. We get a random result from a network at first time. This result is not a desirable output because a computer does not know which data is more important than other data, so we need to adjust the weights. A gap between result and desirable output is called error. We calculate error by error functions and propagate an error from the backside of network to beginning of network. Next, we change problematic weights and reduce an error and get close to desirable result by changing the importance of data to adjust a weight. This process is called back propagation.

UMM CSci Senior Seminar Conference, December 2018 Morris, MN.

It continues until the result is sufficiently close to the labels, so the error is minimal.

2.3 Convolutional Neural Network

The purpose of convolutional neural network(CNN) is classification. For example, use a cat image as input, and a computer identifies cat or dog based on learning result of many data set. A computer has difficulty that understanding an image. A computer needs to acquire it from datasets using CNN. A CNN is a combination of many convolutional layers, pooling layers, and fully connected layers. (This will be explained in more detail in next section) For example, VGG-16 which is one of the best types of CNN has 13 convolution layers and 3 fully connected layers. Each convolution layer identifies feature information from an image and returns feature maps. A feature is a pattern of image. A CNN searches for matching patterns for an input image. The output of this process is called a feature map. After passing through several convolution layers, each feature map is shrunk by a pooling layer. Finally, every feature map vote for one classification in fully connected layer. Through this iteration, it becomes possible to classify images. This can help colorization as describe in section 4. [5]

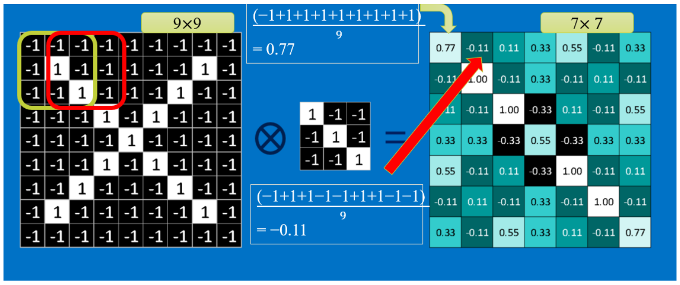
2.3.1 *Convolutional Layer*

A convolution layer detects a feature from a small area of input, and then an output of a convolution layer is passed as an input for pooling layer or another convolution layer. As an illustration, we analyze whether Fig 1 is a cross mark. A convolutional layer applies filters to an image to get a feature map. A filter is a pattern of a cross image which was acquired from many data set by training CNN. Each filter has elements which are necessary to construct a cross. A filter is generally at least 2 × 2 pixel and is applied to a small area of an image. It compresses images to obtain feature values. A feature value is obtained by dividing a sum of value that multiplies filter value and image value by a number of filter pixel. Next, a filter is slid to next area of image to execute the same process. The spacing of the position to apply a filter is called stride. Each stride has a small overlap with neighboring strides because it is more efficient to get better feature values. A CNN repeats this process until it is applied to an entire image. The results of applying the filter to many subsets of the image are collected together into a feature map like the right hand side of Fig 1

Eventually, a convolution layer generates feature maps from compressed features values by an amount equal to the number of filters.

2.3.2 *Pooling Layer*

A pooling layer transforms a feature map into a more manageable form. Compressing a feature map based on feature values is called pooling. It applies at least a 2 × 2 window to the entire feature map to get the highest value in a window. By this process, a CNN can clarify a feature in an image and discard unnecessary information. Moreover, a pooling layer reduces calculation cost.



9×9

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

$\otimes$

1	-1	-1
-1	1	-1
-1	-1	1

=

$\frac{(-1+1+1+1+1+1+1+1+1)}{9}$
$= 0.77$

$\frac{(-1+1+1-1-1+1+1-1-1)}{9}$
$= -0.11$

7× 7

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

Figure 1: Convolution Layer Calculation. Leftside is an input image, rightside is a feature map, and middle is a filter. [5]

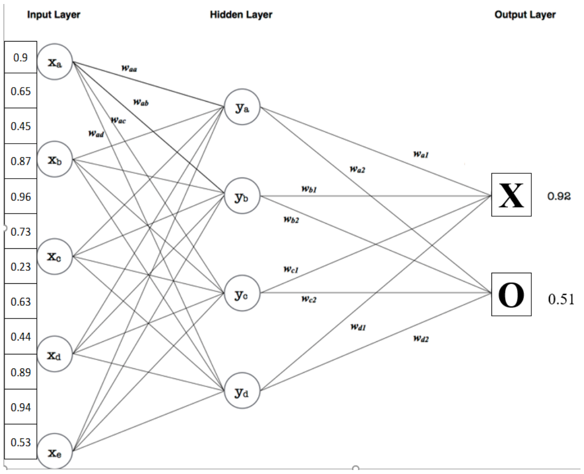


Figure 2: Fully Connected Layer. `https://bit.ly/2JaVDOw`

2.3.3 *Fully Connected Layer*

At the end of CNN, every feature map vote on the classification of the image. In the example in Fig 2, they vote for either a cross mark or a circle mark to get confidence value. Eventually, every feature map ends up with 1 × 1 pixel which is a single value. These values are an input of the fully connected layer, which predicts circle or cross based on weight and activation function. An activation function helps to organize value. In Fig 2, they use Sigmoid function as activation function. If an input value is small, an output value goes to 0. If an input value is large, an output value goes to 1. An hidden layer applies activation function to value which is *inputvalue* × *weight*. However, we do not get the desired outcome on the first try since the contribution of each node are not known: a cross may be classified as a circle, and vice versa. Each node has different importance. Some node is good for finding cross mark, some node is good for finding circle mark. Therefore, we backpropagate network to update the weights and reduce errors.

2.4 Fully Convolutional Neural Networks

The Fully Convolutional Neural Network (FCN) is an approach to semantic segmentation. A semantic segmentation is defined as a high definition prediction task. As an illustration, a network classifies a bike and a human (Fig 3). Semantic segmentation models draw a contour line indicating the boundary of each object. It is similar to classification It predicts image to get confidence value of an input image. FCN does classification to each pixel in image. FCN can be used for classification as in Fig 3. When used for sematic segmenation, FCN perform classification to each pixel in an image. However, in a conventional method, only fixed size images could be used, and a speed of calculation was slow. An FCN is able to solve this problem. [7]

2.4.1 Advantage of FCN

An FCN is a method of end to end semantic segmentation. Normally, a CNN will output a classification result, but an FCN will output a 2-dimensional heat-map result. Most of the structure of FCN is similar to CNN. The only difference is that an FCN adopts a convolution layer instead of a fully connected layer at the end of a network. This method can take any size of input and is faster than a CNN.

2.4.2 Up-sampling

An up-sampling layer restores a feature map which lost detail information due to the pooling layers back to half the size of the original image. In other words, it may be seen as an inverse of a pooling layer and convolution layer.

An up-sampling layer receives feature information values from a feature map and sets the values into a next feature map empty frame. Next, based on these values which are already determined, the upsampling layer fills in values in every position of the result.

2.4.3 Skip Connection

A skip connection passes detailed information to upsampling layer. By repeated a pooling, an image becomes coarse and loses detail information. Altogether, the pooling processes discard detail information to clarify the pattern of an image but lose position detail information, so an output becomes a low resolution. By linking information from an earlier layer to an upsampling layer, it is possible to recover the lost detail information and to create a highly accurate prediction map. In Fig 3, the output of the bottom part of an up sampling network is very coarse, but the output of the top part of an upsampling network has more detail than the bottom part because the skip connections allow the network to restore detail.

2.4.4 U-Net

The U-Net is an encoder-decoder structured semantic segmentation model. The encoder uses a CNN which encodes the features of an image using over 1000 feature maps. A decoder expands and combines feature maps by deconvolution and upsampling until the feature maps become one feature map half size of an input image. This is a derivative model of an FCN. A U-net has more upsampling layers and skip connections than an FCN to improve the quality of an image. A top level of network has more detail than a bottom level of network. (See Fig 3) A skip connection is applied to each up-sampling layer. [6]

2.5 Generative Adversarial Network

The Generative Adversarial Network (GAN) consists of a generator network and a discriminator network. A generator network generates an image which imitates a real image. A discriminator takes datasets image and generated an image as inputs and determine wether an input image is real or not. Both networks are trained at the same time. A generator tries to generate a more accurate and high quality image, and discriminator tries not to be fooled by the generated images. [1]

2.5.1 GAN Structure

A GAN consists of discriminator (D) and generator (G). Both networks are multilayer perceptron which is basic model of artificial neural network. A generator creates imitated images from noise. It tries to create a more realistic and accurate image. A discriminator takes a real image dataset and generated images as input and determine whether a given image is genuine or not. (Fig 4) A generator learns to generate images which the discriminator may erroneously recognized as an image coming from a real image dataset. A discriminator learns to determine whether an image is real image or image that created by the generator.

GANs use a value function V(D, G) defined by Equation 1 represents an error function for GAN. X is distributed according to $P_{data}(x)$ which is the distribution of real images. Z is distributed according to $P_z(z)$ which is the distribution of random input noise. $E_{x\sim P_{data}(x)}[\log D(x)]$ is expected value of $logD(x)$, and $E_{z\sim P_z(z)}[\log(1-D(G(z)))]$ is expected value of $log(1-D(G(z)))$. A goal of this network is to minimize this formula with respect to G and maximize it with respect to D. G creates an image $G(z)$ based on Z. D determines whether an image is real or generated. The goal is $D(x)$ approaches to 1, which D wants to be high since it wants to always recognize a real image as a real image. $D(G(z))$ approaches to 0, which D wants this to be low because $G(z)$ is generated image, and G wants this to be high because it generates more accurate imitated image. If D can discriminate well, the value of $D(x)$ increases and $log(D(x))$ also increases. Conversely, the value of $D(G(z))$ decreases and $log(1-D(G(z)))$ increases. On the other hand, if G generates an image which D cannot correctly determine, the value of $D(G(z))$ increases and $log(1-D(G(z)))$ decreases.

$$\min_G \max_D V(D,G) = E_{x\sim P_{data}(x)}[\log D(x)] \\ + E_{z\sim P_z(z)}[\log(1-D(G(z)))]. \quad (1)$$

2.5.2 Conditional GAN

The discriminator uses additional information to distinguish a generated image and real image, and additional information is used in the training of a generator too. Simply put, additional information is mixed with each input of a generator and a discriminator. This information creates parameters which guide the generator and discriminator to generate the desired form of output. For example, when a GAN recognizes a face, adding smile expression information to the generator and discriminator will force a generator to try to generate smile face, and a discriminator will try to determine whether an image contains a face with smile. [4]

3. IMAGE TO IMAGE TRANSLATION

In recent years, the software pix2pix [3] has attracted attention in the field of image generation and colorization. This software uses conditional GAN (cGAN) and U-net. This cGAN uses U-Net as a generator, and use Patch GAN as discriminator. This cGAN generator network takes a real image as additional input data along with noise to generate an imitated image.(Fig 5) Moreover, this software uses L1 to improve a quality of image. [3]

3.1 Methods

L1 and L2 are commonly used general purpose error functions that are useful for capturing the rough scale structure

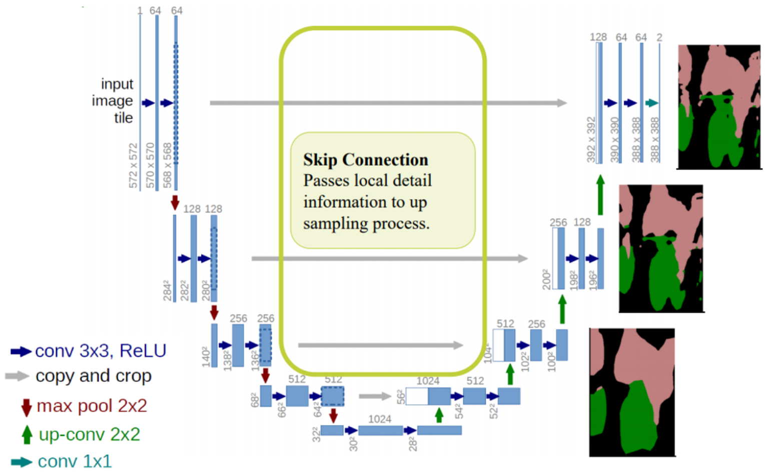


Figure 3: U-Net diagram based on [6]. Bike and human image example from [7].

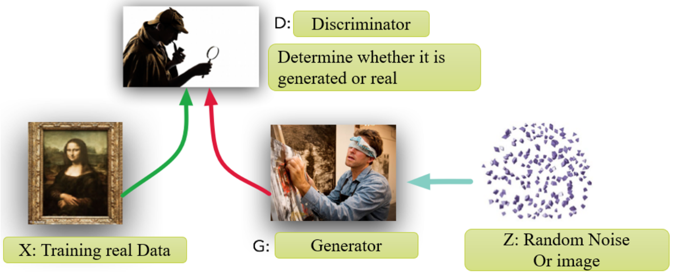


Figure 4: Structure of GAN. `https://bit.ly/2ttFZtv`

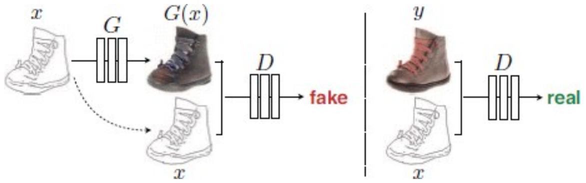


Figure 5: Structure of conditional GAN . [3]

of an image, but typically lose detail, and resulting image become blurred and low resolution. (See [3] for more details on L1 and L2. See Fig 6 for example of images generated with L1.)

Both methods are called regularization. It helps to avoid overfitting, which is when a result become too specific to the training data. The regularization minimizes the size of weights. The L1 and L2 have different approaches to it.

Since GAN aims at a result which can not be distinguished from an observed image, it does not generate a blurred image. A GAN can capture the more detail features. pix2pix takes advantage of L1 and GAN to generate a more realistic images. With L1, the GAN discriminators no longer needs to focus on the global structure of an image, so discriminator only needs to focus on detail part of image. This idea is called PatchGAN. It classifies N $N \times N$ size of patches to determine whether an image is real or imitated.

In order to realize translation of image, high-resolution inputs and outputs are significant. A large amount of low-level information needs to be shared between input and output. The U-net meets the requirement of this condition because it has a lot of skip connection between pooling layer and up-sampling layer.

3.2 Set up for Experiment

In [3], the authors experiment in various image fields: Semantic labels to photo, Architectural label to photo, Map to aerial photo, and black and white to color photo, etc. Here I will focus only on image conversion and colorization. The research group adopts “real vs fake” perceptual studies on Amazon Mechanical Turk (AMT) for the evaluation of experimental result. The authors show an image to humans via AMT and evaluate percentage of people who consider a generated image to be real. In this experiment, 50 testers each classify 50 images. A tester can see the real images and imitation images for 1 second, then the tester attempt to identify the real images. The best score from the generators perspective is 50% which is highest score if discriminators were choosing randomly.

3.3 Utility of cGAN

Fig 6 and Table 1 experiments show how cGAN + L1 method is effective in colorization. A neural networks converts labeled black and white images to colorized pictures to determine which method can produce detailed and high resolution image. Fig 6 experiment is to generate a street scene image from a labeled input image. When only L1 is used, the output image is blurred and low resolution. When L1 + cGAN is used, the output image becomes clear and has more details.

Next, the authors adopt converting photo to map and map to photo test to validate cGAN effectiveness.(Table 1) In this test, L1 and L1 + cGAN network convert photo to map and map to photo. In both tests, L1 + cGAN gets a higher score than only L1 method. Many AMT users felt that the L1 + cGAN result is more realistic.

Table 2 shows the effectiveness of the authors method in colorization. They compare three methods: L2 norm, Zhang technique, and L1 + cGAN. Zhang 2016 method is specialized for colorization. By contrast, cGAN is not specialized,

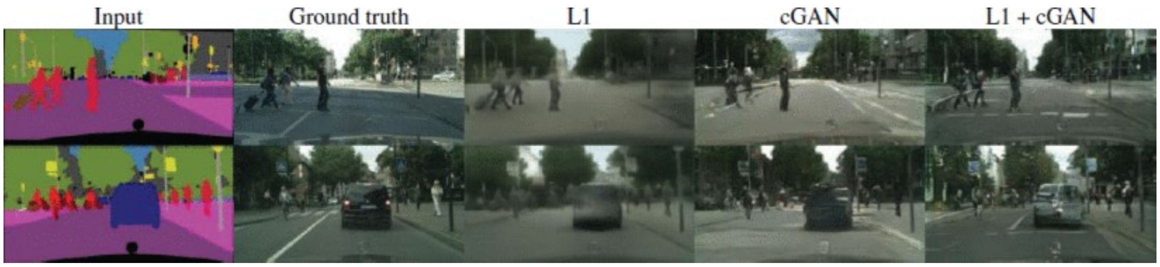


Figure 6: Labeled Image to Street Scene. Input is a segmented image [3]

	Photo to Map	Map to Photo
L1	2.8%±1.0%	0.8%±0.3%
L1 + cGAN	6.1%±1.3%	18.9%±2.5%

Table 1: Image conversion AMT score. [3]

	AMT tester labeled real
L2 Regression from [9]	16.3%±2.4%
Zhang et al, 2016 [9]	27.8%±2.7%
L1 + cGAN	22.5%±1.6%

Table 2: Colorization AMT Score. [3]

but it can be widely used in various field. In the AMT test, L1 + cGAN deceived people 22.5% of the time, and Zhang technique deceived people 27.8% of the time. Both scores are greater than L2 method. Even if L1 + cGAN method is lower than Zhang 2016, L1 + cGAN can be used in other fields, so the AMT scores are reasonable.(Table 2)

3.4 Utility of U-Net

U-Net can bring low level information to up-sampling layer to catch detail information. Fig 7 shows the result. The bottom images are generated by U-Net, and the top images are generated by a encoder decoder network which is created by removing the a skip connection from U-Net. Clearly, you can see that U-Net produces sharp image with L1 + cGAN. U-Net catches the shape of car and tree. Although it is blurred, it catches the shape of car even if in case only L1 is used.

4. GLOBAL AND LOCAL COLORIZATION

The conventional methods of colorization require input by a user guide or reference image which is similar to greyscale image. Since these methods can not easily obtain a desired result, trial and error are necessary. The authors model [2] does not require that user input, but instead requires a large amount of data set to learn the relevant global and local features.(Fig 8) [2]

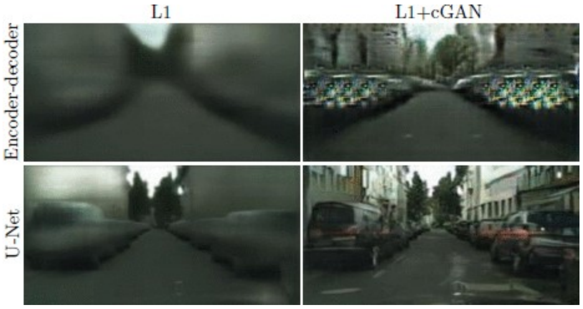


Figure 7: Compare U-net and encoder decoder network. [3]

4.1 Methods

The author methods consist of 6 network: low level feature network, middle level feature network, global feature network, fusion layer, classification network, and colorization network.

The low level feature network consists of 6 convolution layers. This network does not use pooling layer, but instead uses a convolution layer with an increased stride to reduce the size of feature map. Therefore, this network does not need to use a skip connection in the up sampling layer. It shares local information with the global feature network and middle level feature network.

The middle level feature network consists of 2 convolutional layers to obtain middle level feature information. This network does not include a fully connected layer A stride is 1×1, so the output of the network is the same size as the input and includes 256 dimensional feature maps.

The global feature network consists of 4 convolution layers and 3 fully connected layers to obtain 256 global information vectors. This network next passes it to the fusion layer and classification network.

The fusion layer combines information from the global feature map and middle level feature map. This network concatenates 256 dimensional global feature vectors with 256 middle level feature map to obtain new feature map. The output of this layer is 2D feature map which is same as output of fully convolutional neural networks.

The classification network reinforces global context. The colorization network use MSE criterion (Measure the poor prediction accuracy) for training. The MSE crierion compare the original image and generated an image to obtain a difference of value for each pixel. Then squaring each pixel to remove negative values. and compute the mean of all of these squares. However, it lacks global context which distinguish between indoor or outdoor. The authors adopt classification network to provide this context. This network guides a training of global feature network. The colorization network is able to understand outdoor and indoor context based on result of classification network.

4.2 Colorization

The colorization network takes the fusion layer output as input to expands it until it is half the size of the original

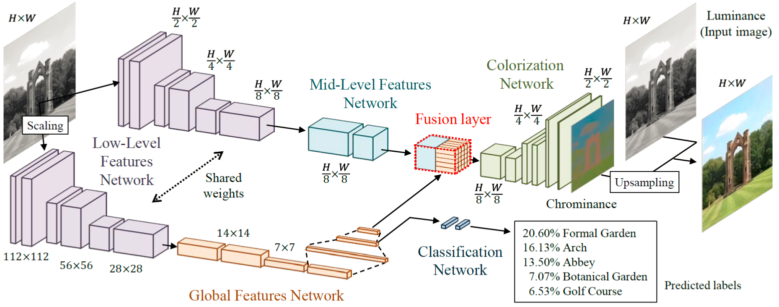
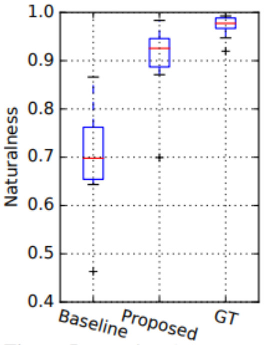


Figure 8: Structure of Joint end to end network. [2]



Approach	Naturalness (median)
Ground Truth	97.7%
Proposed	92.6%
Baseline	69.8%

Figure 9: Result of Global Network Experiment. [2]

input image. It applies up-sampling layer twice and deconvolution 4 times. An output of this network is chrominance (RGB values). At the end of a network, combine chrominance image with luminance image (Black and white image) to generate a colorized image. For training, this network adopts MSE criterion.

4.3 Experiment

The authors evaluate the effectiveness of global feature networks by compares a network without a global network and the network developed by the author. The network without global features is called baseline, and the author network called proposed. During training, network learned feature information from 2,448,872 training images. The training image set includes 205 type of scenes such as abbey, conference center, and volcano.

Fig 9 is the result of the experiment. The naturalness shows how an image is well colorized. 10 different users view 500 images per 3 categories to evaluate naturalness. GT is ground truth. The baseline result is 69.8% which mean this is not well colorized. In contrast, the proposed result is 92.6% which mean colorized very well. Therefore, the global network is effective for colorization.

5. CONCLUSION

The global local image colorization method is good for black and white image colorization. Image to image translation method can be used in various fields of colorization. Both methods have different excellence and advantages. A colorization with neural network method is improved year after year by an evolution of new technology. A GAN is a typical example. This technology dramatically improved accuracy in sketch field. Moreover, it succeeds in reducing the amount of data required for learning. This technology has versatility, so it will be used in other field and improve the quality of the generated result.

6. ACKNOWLEDGEMENT

Thank you to Nic McPhee, Elena Machkasova, and Mitchell Finzel for advising and feedback.

7. REFERENCES

[1] I. J. Goodfellow, J. Pouget-Abadie, B. X. Mehdi Mirza, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Generative Adversarial Nets*, 2014.

[2] Iizuka, Satoshi, 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. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, pages 110:1–110:11, New York, NY, USA, 2016. ACM.

[3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Image-to-Image Translation with Conditional Adversarial Networks*, pages 640–651, New York, NY, USA, 2017. IEEE.

[4] M. Mirza. Conditional generative adversarial nets. In *Conditional Generative Adversarial Nets*, 2014.

[5] B. Rohrer. How do convolutional neural networks work? In *How do Convolutional Neural Networks work?*, 2002.

[6] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *U-Net: Convolutional Networks for Biomedical Image Segmentation*, 2015.

[7] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Fully Convolutional Networks for Semantic Segmentation*, pages 640–651. IEEE, 2016.

[8] 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*, SIGGRAPH '02, pages 277–280, New York, NY, USA, 2002. ACM.

[9] R. Zhang. Colorful image colorization. In *Colorful Image Colorization*, 2016.