CSCI5210: Assignment 1

Shujun WANG
Student ID:1155100742
sjwang@cse.cuhk.edu.hk

Colorization

1 Introduction

Convolutional neural networks have a strong capacity to learn different pattens and attain good performance in many vision tasks. Satoshi *et al.* [Iizuka *et al.*, 2016] proposed a joint end-to-end network architecture for global and local image pairs learning to achieve Automatic Image Colorization and Simultaneous Classification, which showed significant improvements compared against the state of the art.

In this project, a fully automated data-driven approach is used for colorization of grayscale images without image classification based on their previous work. We only utilize partial neural network of the joint network [Iizuka *et al.*, 2016] which also has a good performance.

We will introduce the whole experiment process in the next section. The pre-process for data will be described detailed firstly. After that, we will explain the architecture of the convolutional neural network. Then training details will also be elaborated in the following. In the end, we will show the results and discuss the them with initial color images and compare the performance with the outputs from joint model [Iizuka *et al.*, 2016].

2 The Proposed Algorithm

2.1 Data Pre-processing

In this project, we utilize a small dataset containing alomost $41,000$ color images with size 256×256. To get the inputs and labels of the neural network, we can easily transfer images in RGB color space into LAB space using OpenCV Library, where L channel is our input and AB channels are the output exactly. Due to the input size of our neural network is 224×224, random crop is used for data augmentation. Other procedures such as random rotation 90/180/270 degrees, random flip horizontally and vertically have a good effect on data expansion. We remove some grey images mixed in the training dataset and those images with similar mean value among RGB channels.

2.2 Network Architecture

Our network architecture overview is illustrated in Figure 1. It consists of an encoding path and a decoding path. A 3×3 convolution with same padding, followed by a ReLU activation, a Batchnormalization layer consist of a convolutional Unit. Activation layers and Normalization layers are not shown in Figure 1. The text labeled on the top of Convolution Unit represents the number of channels in each layer. In encoding path, a 3×3 convolution layer with stride 2 is used for downsample instead of pooling layer. The number of feature maps are doubled to increase. The decoding path is composed of a convolution Unit and an upsampling layer using Nearest-neighbor interpolation to expand feature resolution. We use ReLU as activation function for the whole network architecture except the last layer with Sigmoid activation. The input of our network only has one channel which is the grey image needed to be colorized. The output has two channels which are A and B channels in LAB color space. For the whole network, we downsample 3 times during encoding path and only upsample 2 times in decoding path. Hence, we need to upsample the output of the network one more time then transfer LAB channels with same size into RGB color space.

2.3 Training Details

All the convolution layers are initialized by a truncated normal distribution centered on 0 with $stddev = sqrt(2/(fan\_in + fan\_out))$, where fan_in is the number of input units in the weight tensor and fan_out is the number of output units in the weight tensor. Weight of each convolution layer is regularized by $2 - norm$ with a penalty of 0.001. SE(Square Error) is computed between the output and target output combined with weights normalization as the loss. Since without getting the mean of loss, a small learning rate is utilized in this project. We use Adam optimizer with $\beta\_1 = 0.9, \beta\_2 = 0.999, \epsilon = 1e - 08$, decay $= 0.001$ and initial learning rate of $1e - 7$ which will be updated by multiplying 0.9 every 10 epochs. We use 4 GPUs and batch size of 64 to train the network, training almost 20 hours.

3 Results and Discussion

Evaluation is done on $4,000$ diverse images. We show colorization results on the test dataset in Figure 5. The first two columns are color images and grey images respectively. The images shown on the third column are generated by our methods, while the last column shows results of Satoshi *et al.* [Iizuka *et al.*, 2016]. It is clear that the colorization results of our model are quite "natural". The staffs with blue and green color are easier to learn such as blue sky, oceans, grass and

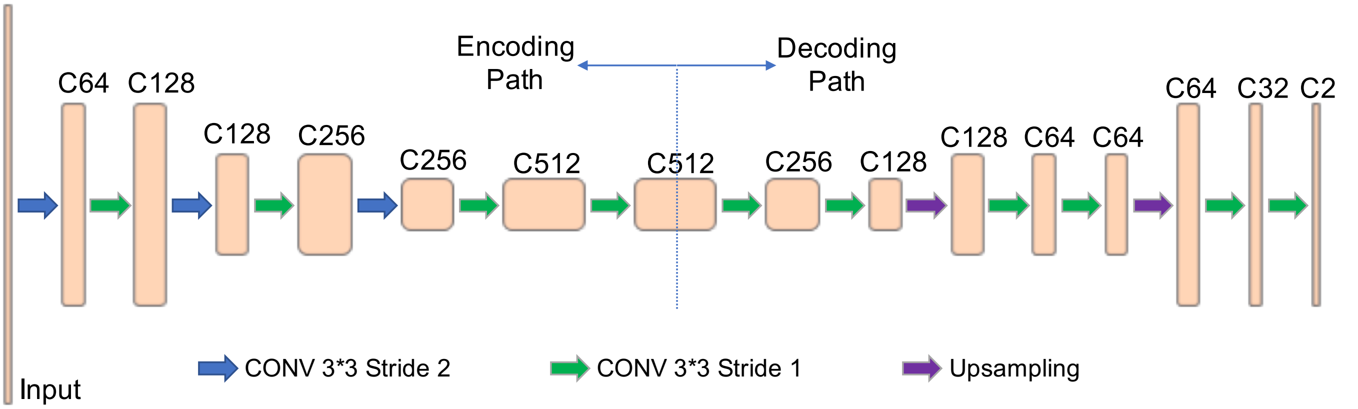

Figure 1: Overview of our model for automatic colorization of grayscale images.

trees. However, some bright colors are difficult to transfer especially for those with small and irregular shape. Since the output of our model has the size of 112×112, there exists a certain loss while training neural network, and has a bad effect on learning things with small area. Compared to the results from Satoshi *et al.* [Iizuka *et al.*, 2016], our model is not very stable when small staff is processed.

Image-to-Image Translation

1 Introduction

Phillip *et al.* [Isola *et al.*, 2017] proposed a common framework to solve the problem which is related to automatic image-to-image translation as a task of translating one possible representation of a screen to another. Conditional Generative Adversarial Networks(GANs) plays an important role in this image translation problem.

In the project, we try to reproduce the results using the methods proposed by Phillip *et al.* [Isola *et al.*, 2017]. The facades dataset is used to test whether the model is feasible.

2 The Proposed Algorithm

2.1 Data Pre-processing

CMP Facades dataset [Tyleček and Šára, 2013] is used for architecture label to photo translation. Facades dataset contains 400 training image pairs, 100 validation image pairs and 106 testing image pairs. For each image pair, an initial nature image of different building is corresponding to a colorful sketch. Each image is RGB format with size 256×256. For generator, the sketch image is considered as the input and original building image is considered as the target. The function of discriminator is to distinguish whether the input is true.

2.2 Network Architecture

For generator network, we use a U-net architecture which is used to generate realistic images of the original building according to the sketches. This network architecture is similar as network mentioned earlier. To get more detailed local information and circumvent the bottleneck for information, skip connections are utilized in generator. The generator architecture is shown in Figure 2. Each block in Figure represents a Convolution-BatchNorm-ReLU layer. Figure 3 shows the architecture of discriminator network.

In many cases, L1 loss could accurately capture the low frequencies but failed on high frequency information. To better model high level loss, patch discriminator which could classify whether patches of image is real or fake is proposed. We concatenate all the patch discriminator results to get the final judgement. For each patch discriminator architecture, we use a similar network as the generator encoding stage, but change the input size from 256 to 64 to realize function of patch discriminator. Binary cross entropy loss plays an important role in optimizering discriminator. All convolutions in encoder and in the discriminator are 3×3 spatial filters applied with stride 2 to downsample feature maps by a factor of 2. 3×3 convolutions with a stride of 1 and 'same' padding are used for decoding in generator. All ReLUs in the encoder and discriminator are leaky, with slope 0.2, while ReLUs in decoder are normal ReLU. Mean Absolute Error(MAE) is used to optimize generator. We use Adam optimizer with a 0.001 learning rate.

2.3 Training Details

The GAN network is trained respectively for generator and discriminator. In one iteration, we freeze generator network firstly to train discriminator network once. Then we freeze discriminator network to train generator network. Images with size of 256 are directly through into the generator network while training. We use real and fake images alternatively while training discriminator network and crop one 256 image into 16 image patches with size of 64 as the input of discriminator. We trained $30,000$ iterations almost 400 epochs with a fix learning rate of $1e-3$.

3 Results and Discussion

Evaluation is done on 106 sketch images. We show translation results on the test dataset in Figure 4. Images in first column are real images. Sketch images are illustrated in column two, which are used to generate output of generator. After training the whole cGAN, generator is utilized to yield outputs which

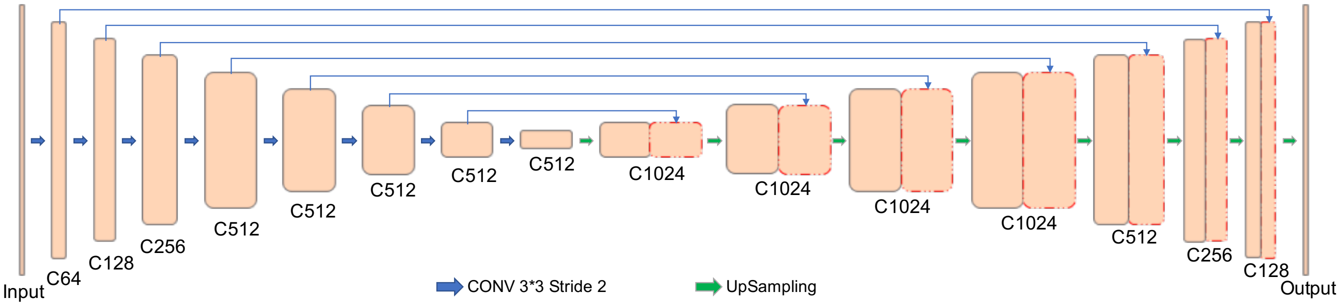

Figure 2: Architecture of GAN generator network.

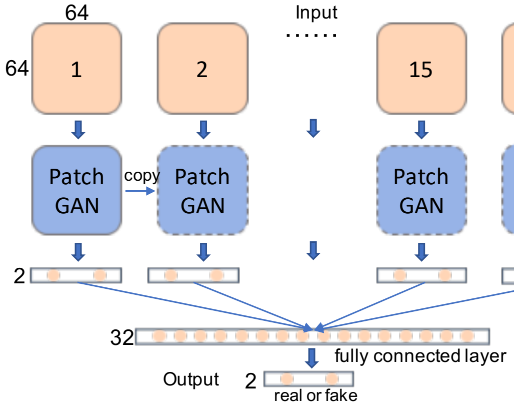

Figure 3: Architecture of GAN discriminator network.

are shown on the third column. Images in the last column are the outputs of the model trained by Isola *et. al* [Isola *et al.*, 2017].

As we can see from the results in Figure 4, outputs of our network could capture some global information and detailed information. It could distinguish glass buildings and brick building easily. However, due to the limitation of detailed description in the sketch images, our generator results are blurry in detail and do not continuously. The difference between results of us and Isola *et. al* is the batch size. They set the batch size from 1 to 10 according to different tasks. But in this project, we just samply set a batch size of 64. But here we did not train another model with a small batch size, which we can treat this as a future work.

References

[Iizuka *et al.*, 2016] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi 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):110, 2016.

[Isola *et al.*, 2017] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[Tyleček and Šára, 2013] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *German Conference on Pattern Recognition*, pages 364–374. Springer, 2013.

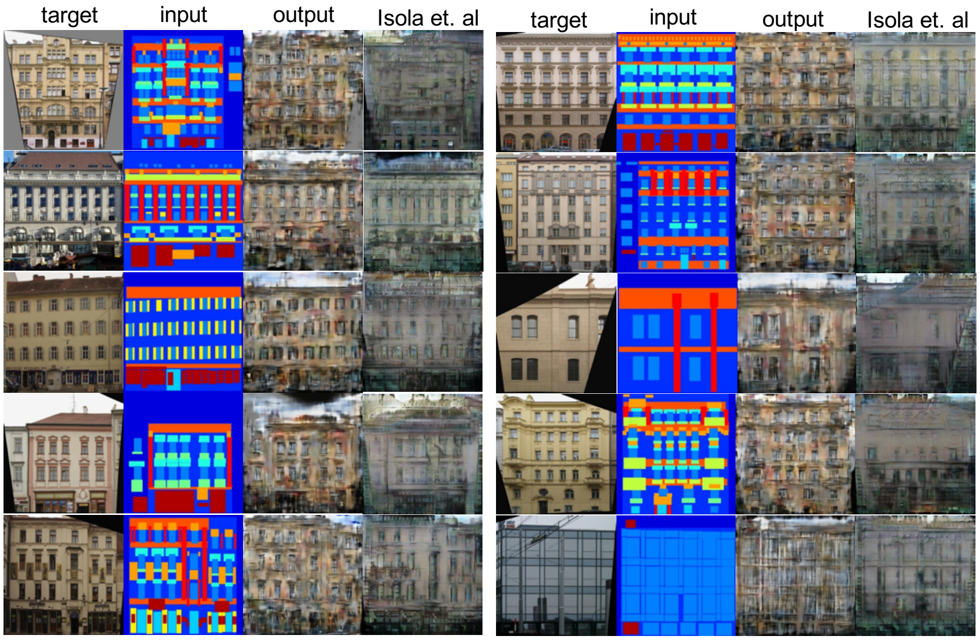

Figure 4: Translation results of cGAN.

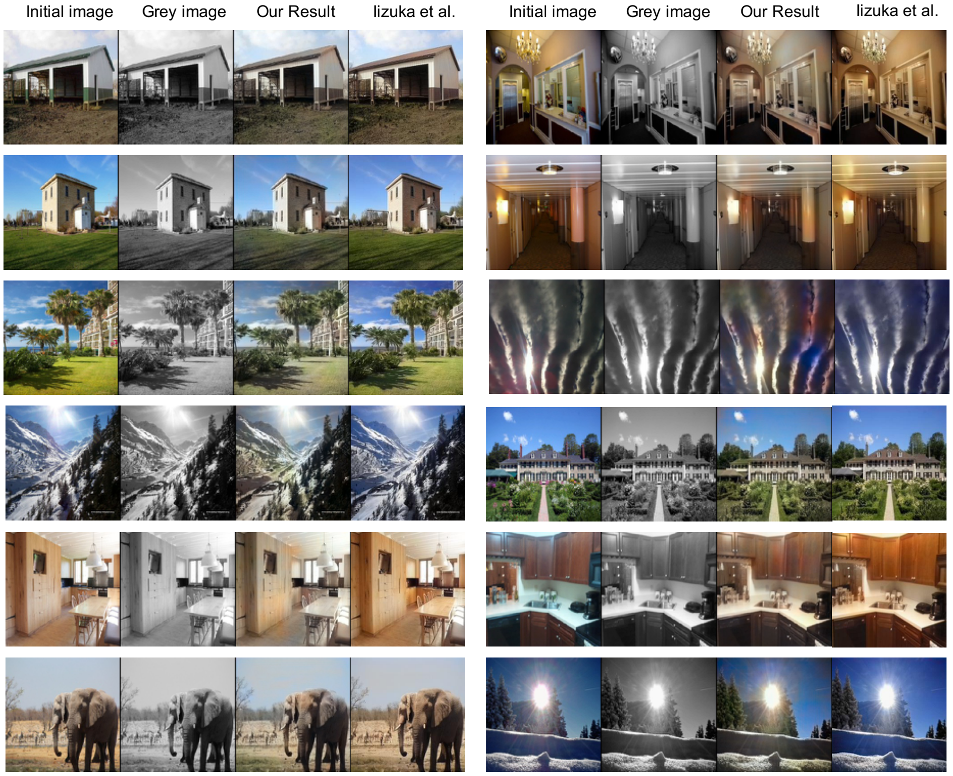

Figure 5: We show the results of our approach on some of the images from the test dataset.