DSA5204 Final Report

Colourful History - Image Colorization based on GAN

Type of project: Application

Group Members:

[redacted]

Background

It is an important part of the image processing system to colorize the grey-scale image. As a kind of computer-aided technology, image coloring has a wide range of applications in the field of video processing and old photo restoration. The current image colorization techniques mainly based on color transfer and color-based expansion of two methods.

Welsh et al. [9] first proposed an image coloring method based on color transfer. This method needs a color reference image to transfer the color information of the reference image to the image to be colored. Welsh considers searching for similar pixels in the brightness channel of the image, transferring the chromatic information corresponding to similar pixels, and finally getting the coloring result similar to the color of the reference image. However, pixels with the same brightness can only have one color, which greatly reduces the types of colors that can be colored, and the coloring results will be greatly limited by the reference image. In 2004, Levin et al. [10] designed a simple coloring method based on local color expansion. Their method is based on a simple premise: adjacent pixels in space-time with similar intensity should have similar colors. Based on this premise, users only need to annotate images with some colored graffiti, and those indicated colors can automatically propagate in space and time to produce fully colored images. Different from previous methods, Yatziv et al. [11] proposed a coloring method based on color blending in 2006. This algorithm uses the principle of brightness weighting and chrominance mixing, and can obtain high quality static image and video coloring results with low complexity and acceptable computational cost. However, these methods need significant user intervention, so as to make the color of the colored image meet people's expectations. In recent years, with the continuous development of deep learning technology, the current image coloring methods mainly use CNN, GAN and other deep learning methods.

2 Technical Approach

Our technical approach can be mainly divided into four parts. Firstly, we choose the proper dataset and preprocess the data, which mainly contains resizing image, splitting dataset, data augmentation and converting channel. For the modelling part, we mainly use GAN together with its variants, Unet GAN, Res-Unet GAN, Dense-Unet GAN, attention-Unet GAN, and traditional CNN for comparison. L1 loss is used as our loss function. Meanwhile, some training tricks are added in this part. For the evaluation part, we use MAE, Accuracy, PSNR, SSIM four metrics to measure and compare the performance of our models. Finally, we colorized some old pictures with our best trained models.

3 Reproduction and Extension

3.1 Data and Pre-processing

We choose the Flickr2K dataset, which contains 2650 images of animals, landscapes and people for our project. For the data pre-processing part, to make the code self-explanatory, we firstly resize the image to 256*256, then we split the dataset, and flip the training set horizontally. We convert the RGB image we read to Lab color space and separate the grayscale(L) channel and the color(ab) channels as our inputs and targets for the models respectively.

3.2 Modeling

3.2.1 Loss function

Our model is based on GAN. In addition to using the basic loss function, we add another part to the traditional loss function (see formula (1)(2)(3)). The main function of new added part is to make the fake image more similar to the real one. At the same time, L1 loss is better than L2 loss in the image field, from the experiments in paper [8], the performance of the model will continue to improve when we use L1 loss on the basis of using L2 loss. At the same time, L1 loss is more widely used in the field of image. It can make local internal color similar, and different segmentation area have different color. At the same time, due to the diversity of color choices, the model tends to produce gray images without L1 loss.

$$L_{GAN}(G,D) = \min_{\theta}\max_{\varphi} E_{x\sim p}\left[logD_{\varphi}(x)\right] + E_{z\sim p0}\left[log\left(1 - D_{\varphi}\left(G_{\theta}(z)\right)\right)\right] \tag{1}$$

$$L_{L1}(G) = E_{x,y,z}\left[\left||y - G(x,z)|\right|_1\right] \tag{2}$$

$$L_{total=arg \min_{G}\max_{D} L_{GAN}(G,D)+\lambda L_{L1}(G)} \tag{3}$$

3.2.2 Conditional GAN

Our goal is to get a color image, which means that the input of the model cannot be an image without content or meaning, instead it needs to have actual content. Specifically, it cannot be a black-and-white image with content. Such input has a certain distribution. Our colorization task is also based on this distribution. The advantage of this model is that the distribution to be simulated by the generator is relatively simple, which is better conducive to the convergence of GAN.

As can be seen from the data in the training, the color filling of the image is not groundless. But according to some edges and some textures, this is also based on the existing information of the input image to make the model get better effect.

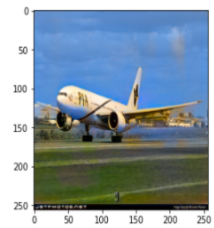

Fig.1. It shows that edge and texture separate the color, image is from the training process

3.2.3 Generator models

In the model part, we mainly modify the generator, from the simplest UNET to Res-UNET to dense-UNET to attention-UNET.

First, we will introduce the differences in the structure of each generation model. UNET can be regarded as encoder-decoder structure, shown in figure 2. Through connection between the encoding part and the decoding part, we can use not only the information obtained from the encoded hidden vector, but also some information obtained from the encoder, such as, in the encoder part, each part has different responsibilities: they can extract edge, texture and target. By combine these parts, it can achieve very good results in image coloring and medical target recognition.

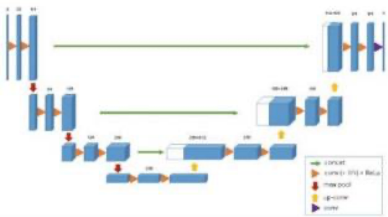

Fig.2. The structure of UNET

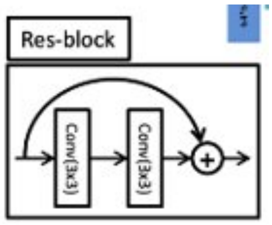

Fig.3. The residual-block

RESNET replaces the convolution layer of the encoder with residual block, shown in figure 3. Inspired by resnet, we can increase the number of layers of the convolution layer. Just concatenate input of each convolution layer to the input of the subsequent convolution layer, we get dense UNET, shown in figure 4.

Finally, we also use the most popular attention mechanism. During the up-sampling, we use the attention gate (figure 5) to control the input. By using attention matrix, we can determine the importance of each input. This makes our multiple inputs not only connected, but also added the weights. This also makes attention-UNET be the optimal generator.

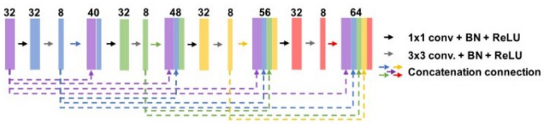

Fig.4. The structure of dense block

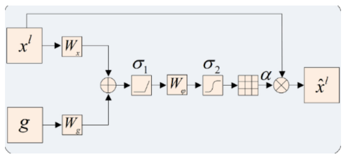

Fig.5. The structure of attention gate

3.3 Training Strategies

- Load data on-the-fly: We use data generator to load data to save memory.
- Transfer RGB to L-ab: We only predict 2 channels (a and b) instead of three to make training more stable.
- Conditional GAN: We use gray scale image with some structure as input instead of random noise.
- Data augmentation: We 'increases' our dataset to reduce the generalization gap by adding noise, rotating, etc.
- Structure of Neural Network: We modify the neural network by using LeakyRule, Batch Normalization layer and deconvolution layer. The most effective strategies for the

performance of model is the loss function we used and batch-norm layer, we can see the performance from figures below:

Fig.6. The first fig shows that the first several epochs will generate grey scale images with traditional loss function and without batch-norm layers. The second fig shows that if we change the loss function, it will add some color when the training starts. The third fig shows that if we use batch-norm layers, it will also add colors when the training starts. But the color is almost same in wide range. The fourth fig shows that if we combine both, in first epoch, it can add color precisely.

4 Results and Implementation

4.1 Result comparison

To visualize the performance, we used our trained model to fit the testing set splitting from Flickr2K Dataset. Our results are much greener and yellower than original photos. This deviation may be due to the combination of our training set, large proportion of photos are green and yellow and neural network like GAN may learn the most frequent colour in the training set which not universal. Among our 5 models, GAN with Attention Unet performed best, photos colorized by this GAN look more realistic and have less colour distortion.

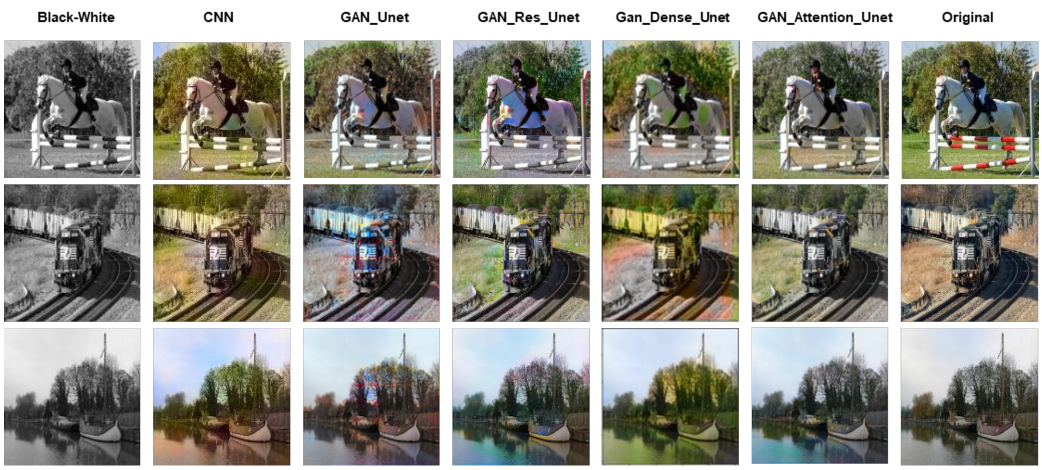

Fig. 7. Example results from our Flickr2K test set.

To measure the performance, we use MAE (mean absolute value) and 5% Accuracy [1] to measure how close our result with true image. And we use PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural similarity index measure) to measure the distortion of our colorized image.

Method	MAE	5% Accuracy	PSNR (Mean)	PSNR (Std)	SSIM (Mean)	SSIM (Std)
Unet	15.7048	0.6181	21.0652	2.8775	0.896	0.0575
ResUnet	13.7539	0.6780	22.4598	3.6226	0.9253	0.0522
DenseUnet	16.0282	0.6117	20.9791	3.0603	0.9074	0.0477
AttentionUnet	12.1038	0.6899	23.2658	3.2792	0.9347	0.0394
CNN	14.9176	0.5933	21.8557	2.5316	0.9103	0.0423

Table. 1. Test set results of our models.

GAN with Attention Unet perform best among 4 metrics, it has lowest MAE and highest 5% Accuracy, PSNR and SSIM. Meanwhile, CNN has lowest standard deviation of PSNR and SSIM, this may be due to the simple structure of neural network.

4.2 Old Photo Colorization

We use our best model, GAN with Attention Unet, to colorize old photos of some famous person like Chinese writer Lin Huiyin and English actress Audrey Hepburn as well as some old movies such as Modern Time and Roma. Our model performs great on sceneries but has a little distortion on human faces and bodies. It is due to the type of scenes in our training set. Most pictures in our training set are landscapes without people, the neural network may not learn well on features related to people.

Fig. 8. Old photos colorized by GAN with Attention Unet.

5 Conclusion and Future Work

5.1 Conclusion of our project

Coloring black and white images is one of the most exciting applications of deep learning. Therefore, in this project, we choose the topic of image coloring, which is basically on the GAN. We first choose and preprocess the dataset, then we apply the data to the GAN model and its variants such as Unet GAN, Res-Unet GAN, Dense-Unet GAN, attention-Unet GAN and so forth. In addition, traditional CNN is also used with the purpose of comparison. Moreover, we choose L1 loss in our project, which is more appropriate compared with the others. Then we train and adjust the model, when MAE, Accuracy, PSNR, SSIM four metrics are used to do some performance assessments towards our models. After the above parts done, we color some old pictures with attention-Unet GAN, which is the model of best performance.

5.2 Furure work

we have done within the limitation of time and capacity of computer, so there still reserve many things that can be improved. For example,

- Tuning parameters to get model with better performance. Training the model needs a lot of time so we just tune some of our parameters when training our models, in the future, we can try to tune all the parameters with brunches of data of different dimensions to make our models more robust.

- Change the discriminator. Because in our project we mainly use the single discriminator, so we think that we can explore more kinds of discriminators in the future, which is also interesting as well as attractive.

- Use lager dataset with different scenes. Now our models perform well in training landscape related images but perform relatively poor in the images of people. Maybe the reason is related to the dataset, so in the future we would like to improve our models through using lager dataset with different scenes like the scenes of people.

Reference

[1] Nazeri, K., Ng, E., & Ebrahimi, M. (2018, July). Image colorization using generative adversarial networks. In International conference on articulated motion and deformable objects (pp. 85-94). Springer, Cham.

[2] He, M., Chen, D., Liao, J., Sander, P. V., & Yuan, L. (2018). Deep exemplar-based colorization. ACM Transactions on Graphics (TOG), 37(4), 1-16.

[3] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial networks. arXiv preprint arXiv:1406.2661.

[4] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

[5] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

[6] Xiao, X., Lian, S., Luo, Z., & Li, S. (2018, October). Weighted res-unet for high-quality retina vessel segmentation. In 2018 9th international conference on information technology in medicine and education (ITME) (pp. 327-331). IEEE.

[7] Guan, S., Khan, A. A., Sikdar, S., & Chitnis, P. V. (2019). Fully dense UNet for 2-D sparse photoacoustic tomography artifact removal. IEEE journal of biomedical and health informatics, 24(2), 568-576.

[8] H. Zhao, O. Gallo, I. Frosio and J. Kautz, "Loss Functions for Image Restoration With Neural Networks," in IEEE Transactions on Computational Imaging, vol. 3, no. 1, pp. 47-57, March 2017, doi: 10.1109/TCI.2016.2644865.

[9] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. Transferring color to greyscale images. In ACM TOG, volume 21, pages 277–280, 2002.

[10] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In ACM transactions on graphics (tog), volume 23, pages 689–694. ACM, 2004.

[11] L. Yatziv and G. Sapiro, Fast image and video colorization using chrominance blending.In IEEE Transactions on Image Processing, vol. 15, no. 5, pp. 1120-1129, May 2006, doi: 10.1109/TIP.2005.864231.