CSCI 1430 Final Project Report: U-Net, Deep CNN, and Conditional GAN for Auto-Colorization

JustSoMeKernels: Wanjia Fu, Shania Guo, Linghai Liu, Matianyu Zang
TA name: Alex Choi.
Brown University

Abstract

Image colorization has been used in a wide range of applications, including restoration of historical images and medical imagery. Among the available models, U-Net, Deep CNN, and Conditional GAN each has its own strengths and weaknesses. To compare and better understand them, this paper proposes and builds the three models for auto-colorization on two datasets, one on a large variety of complex objects and scenes, and a smaller one on fruits and vegetables. The visualization and loss results are qualitatively and quantitatively analyzed and discussed, respectively. This paper offers insight and guidance on the potential causes of different model performance on auto-colorization.

1. Introduction

Colorizing black-and-white (grayscale) images has a wide range of applications in various fields. It helps to revive historical images, enhance medical imagery, and can even be used for artistic purposes. Colorizing these images helps to discover more potential information.

Finding an effective and efficient technique of auto-colorizing grayscale images is still a challenging task in the field of computer vision today. Some popular approaches include using Convolutional Neural Networks (Deep CNN) and Generative Adversarial Networks (GANs), which have been widely tested and adopted (Figure 1). Among them, the U-net model, the Deep CNN model, and the Conditional GAN model are three examples that caught our attention for their highly-rated performance on the auto-colorization task. Thus, our aim for this project is to propose, build and compare these three models and their performance on the same dataset. By comparing the training process and the results of each model, we hope to gain insights into their strengths and weaknesses and provide guidance on the best approach to use for this task. In addition, we hope to gain a better understanding of the various factors that affect the performance of a model on auto-colorization tasks. We use two different datasets, the first is a 5000-training-image dataset of grayscale and RGB images on a wide range of subjects and scenes, including animals, scenery, daily objects, etc. The second dataset is a 723-image dataset with grayscale and RGB images of fruits or vegetables from 20 different categories. Each image has a centered object on a white background. The second dataset is a collection of images with more limited subjects, and therefore we expect the models' performance to be better on this dataset.

Figure 1. Examples of results from existing auto-colorization models

2. Related Work

In this project, we are inspired by the following papers:

Deep Koalarization [2] In the study presented in this paper, the authors have developed an innovative approach to image colorization utilizing an encoder-decoder architectur while leveraging features extracted by large

vision models pre-trained on ImageNet, e.g., Inception-ResNet-v2. The extracted feature is combined with the output of the encoder, fused into a hybrid output. This is then funneled into the decoder, which generates the final colorized image. The authors used a subset of images from ImageNet.

U-Net [4] This paper introduces the U-Net architecture, a unique configuration of convolutional networks designed for highly efficient image processing with skip connections of intermediate results. This model is comprised of two main components: the first half, which employs max pooling layers between convolution blocks to downscale the image, and the second half, which uses upsample layers to expand the image to its original size. What sets U-Net apart is its strategic transfer of information from the first half of the network to the second half, a process that mitigates information loss from the downscaling process. While the authors of the paper applied U-Net to biomedical image segmentation, the model's design makes it particularly well-suited for other tasks as well, including our colorization task.

Conditional GAN [3] The Conditional Generative Adversarial Network (Conditional GAN) is an advanced variant of the traditional GAN. This innovative model enhances the network's capabilities by conditioning it with supplementary information, such as class labels, thereby allowing for a more controlled and targeted generation process. The authors of the paper employed the model to generate MNIST digits and tag vectors on MIR Flickr 25,000, and the model is also effective for image colorization.

Dataset [1] Our study takes advantage of the Natural-Color Dataset (NCD), a comprehensive collection curated by the authors of this paper. The utilization of this particular dataset enables us to tailor our models to the unique subsets of images pertaining to fruits and vegetables.

3. Method

We are given a grayscale image with size $1 \times H \times W$ and the goal is to reconstruct the corresponding RGB image of the same size $3 \times H \times W$.

3.1. Adaptation of Deep Koalarization

The diagram of this model is displayed in Figure 2.

Encoder The encoder consists of a series of convolution layers with kernels of size 3. The number of channels increases from 1 to 64, 128, 128, 256, 256, 512, 512, and back to 256. The strides are set as 2 between layers with the same number of channels, otherwise set to 1 as default. ReLU is used as the nonlinear activation function between each convolution layer.

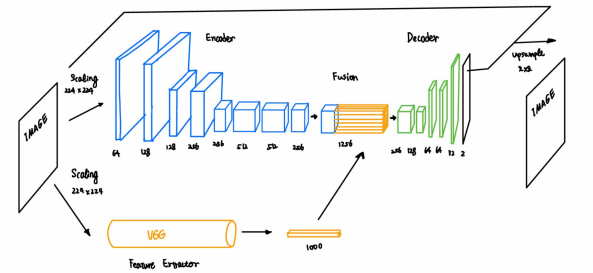

Figure 2. Model Architecture Diagram for Deep-Koalarization

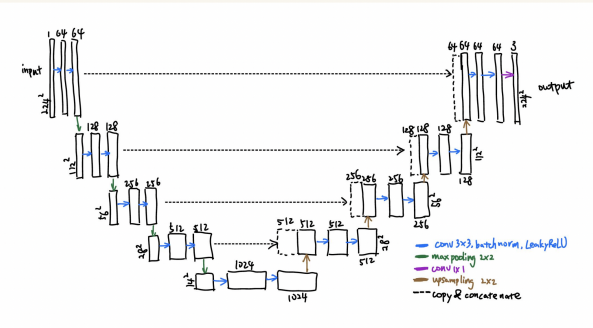

Figure 3. Model Architecture Diagram for U-net

Fusion The fusion layer combines image features extracted by a pre-trained VGG16 model with the encoded input. The length of the VGG16 feature is 1000 for each image, and this feature vector is concatenated to each pixel of the encoded input, thus producing a total of 1256 channels.

Decoder The decoder first uses a convolution layer with kernels of size 3 to map the fused input to 128 channels. The following layers consists of alternating layers of upsample (with scale factor of 2) and convolution (with kernel size of 3). The number of channels goes down from 128 to 64, 32, and 3. ReLU is used as activation, except for tanh used before the last upsample layer.

Loss Mean Squared Error from the ground truth RGB image is used as the loss for backpropagation.

3.2. U-net

The diagram of this model is displayed in Figure 3.

We followed (almost) the same U-net as when it was proposed [4], except that the output has three channels (RGB) rather than the 'a' & 'b' channels in 'LAB' Color Space. The input grayscale image undergoes convolution layers followed by batch normalization, LeakyReLU, and MaxPooling until encoded into 1024 channels. Then the encoded input undergoes upsample. And, with skip connections, the upsampled result is concatenated with intermediate results during encoding, and then passes through convolution layers. This

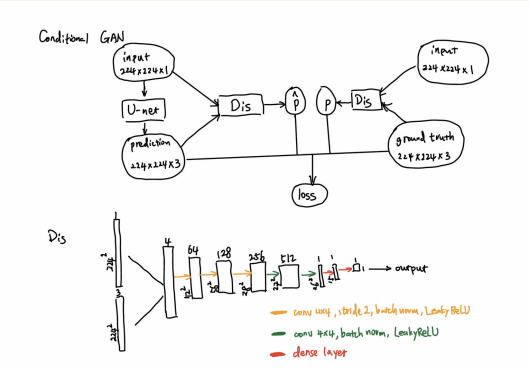

Figure 4. Model Architecture Diagram for Conditional GAN

process is repeated in the decoding process until we finally return to an image of the same height and width, but with 3 channels (RGB). Our implemented U-net also use Mean Squared Error as its loss function.

3.3. Conditional GAN

The diagram of this model is displayed in Figure 4.

Generator We used our implemented U-net as the generator: given a grayscale image, it returns a colored image.

Discriminator The discriminator takes in both a grayscale image and its corresponding colored image, be it the ground truth or generated by the generator, and predicts a probability that the colored image *does* correspond to the grayscale image.

In the discriminator, the two input images are first concatenated into four channels, then undergo 3 convolution layers, each followed by Batch Normalization and LeakyReLU. The last convolution layer maps from 512 channels to a single channel, and the result is fed into an Multi-layered Perceptron (MLP) with a hidden size of 15 and output size of 1. This number is then normalized using `sigmoid` so that it falls between 0 and 1.

Loss Let $x \in \mathbb{R}^{1\times H\times W}$ and $y \in \mathbb{R}^{3\times H\times W}$ be a pair of grayscale and RGB image. Let $\hat{y} = \texttt{Gen}(x)$ be the generated RGB image. $p = \texttt{Dis}(x, y)$ and $\hat{p} = \texttt{Dis}(x, \hat{y})$ be the output probabilities of the discriminator. The GAN loss is defined as

$$\mathcal{L}_{GAN} = \mathbb{E}_{x,y}\left[\log p + \log(1-\hat{p})\right]$$

We also incorporated $\mathbb{L}_1$ loss from y to $\hat{y}$, so

$$\mathcal{L}_{total} = \mathcal{L}_{GAN} + \lambda\mathbb{L}_1(y, \hat{y})$$

where $\lambda > 0$ is a hyperparameter (set as 50).

Numerical Stability During training, we encounter severe numerical instability when using $\mathcal{L}_{total}$ as defined above. Hence, we introduced a buffer term ε to $\mathcal{L}_{GAN}$:

$$\mathcal{L}_{GAN} = \mathbb{E}_{x,y}\left[\log(p+\varepsilon) + \log(1-\hat{p}+\varepsilon)\right]$$

where $\varepsilon > 0$ is a hyperparameter (set as 10^{-2}).

4. Results

4.1. Technical Discussion

4.1.1 Data

We use two different datasets.

Dataset 1 - has 5000 grayscale and RGB images on a wide range of subjects, including animals, scenery, daily objects etc, and about 700 grayscale and RGB image pairs for testing.

Dataset 2 - a 723-image dataset with grayscale and RGB images of fruits or vegetables from 20 different categories. Each image has a centered object on a white background.

Observation On the first dataset 1, U-net performed the best. Our adaptation of Deep Koalarization results in some colorization in addition to blurring the image. CGAN does not perform quite well on the testing dataset, but reached fair performance after replacing its generator with our trained U-net.

In order to obtain better colorization results, we switched to the dataset 2 with a smaller number of training images. This training data is also less complicated, and it contains 20 categories rather than too large a variety of scenes and objects.

On dataset 2, deep koalarization produces completely white images, which we believe might be attributed to the large patches of white in the original gray-scale images and ground-truth color images. Some green vegetables are colorized as red by U-net and CGAN, which might result from the uneven distribution of the colors of vegetables in the training dataset.

4.1.2 Qualitative Discussion

Dataset 1 The performance of all three models are not satisfactory, the models learnt limited colors such as green, brown, and blue, and are not able to color complex objects. We can conclude that the models gain basic understanding of the underlying patterns (Figure 5).

U-net Our U-net model is trained for 150 epochs. It contributes to some colorization but not very satisfactory.

Figure 5. Visualization result on dataset 1.

Deep Koalarization Our deep koalarization model is trained for 150 epochs. It learns some simple patterns, such as the fact that the sky is blue, the grass is green, and the horse is brown. This can be seen in the third row of pictures of a man with a surfing board, where the sky above the ocean is clearly colored blue by deep koalarization. However, it also makes a few mistakes. For example, in the fourth row, deep koalarization identifies the wall as grass and colors it green.

Condition GAN Our conditional Gan model is trained for 100 epochs. Its colorization visualization is similar to that of U-net.

Dataset 2 We observe that the performance is much better. They successfully categorize the objects and give out vibrant colors that almost correspond to the ground truth. Among them, CGAN with our pre-trained U-net performs the best overall (Figure 6).

U-net Our U-net model is trained for 100 epochs. It can be considered as being trained from scratch. It colorizes the color red the plums, the inner parts of the lemons, and a tiny region of the broccoli.

Deep Koalarization Our deep koalarization model is trained for 100 epochs. The visualization results are not shown in Figure 6, because the output images are all completely white. We currently attribute this to the fact that a large portion of the original gray-scale images and ground-truth images are white.

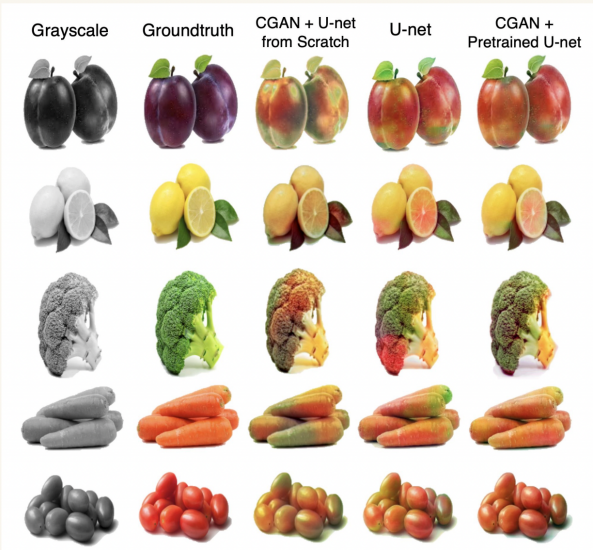

Figure 6. Visualization result on dataset 2.

CGAN+U-net from scratch Our conditional Gan model is trained for 100 epochs. The generator of CGAN doesn't take any pretrained weights from the U-net model, but rather is trained from scratch. The result is that roughly half of the broccoli (in Figure 6) is colorized with red, and a large part of the carrots is also colorized with green.

CGAN + pretrained U-net This time, the conditional Gan model takes pretrained weights from U-net model, and the colorization performance is better than the previous options. The majority of broccoli is green, and the carrots and tomatoes are red. One problem across all these model choices is that the plums have been colorized as red. This might be because the model has identified the plums as apples, since their shapes look quite similar.

4.1.3 Quantitative Discussion

Our adaptation of Deep Koalarization does not converge, so the loss is omitted. For conditional GAN in the plots, the generator parts are U-net shown in the middle panel. If the CGAN's are trained from scratch, the loss would be much lower, but the contributions are not mainly by minimizing the L1 loss.

We trained Koalarization and U-net on the 5000-image dataset for 150 epochs in total, but we first trained for 50 epochs and continued training for the rest 100 epochs, so there may be some spikes around epoch #50. For all models (Deep Koalarization, U-net, and CGAN) on dataset 2 and all CGAN models, we trained for 100 epochs and record training losses only.

U-Net The U-Net trained on the 5000-image dataset 1 con-

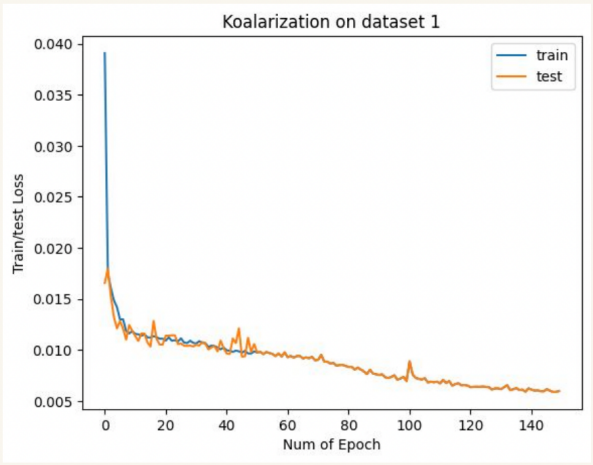

Figure 7. Loss plot for deep CNN on dataset 1.

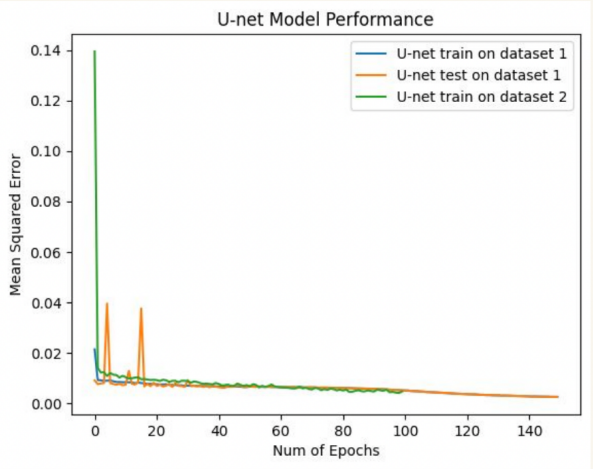

Figure 8. Loss plot for U-Net on dataset 1 and 2.

verges well, though in the test dataset, the loss experinces a few spikes around 5 and 18 epochs. For U-Net trained on the 723-image fruit dataset, the training loss is smaller than that for the first training dataset (Figure 8).

Deep Koalarization The Deep Koalarization trained on the 5000-image dataset converges well (Figure 7), but when trained on the 723-image fruit dataset, its loss oscillates a lot towards final epochs and doesn't converge. So its loss (Figure 10)is not plotted together with its loss for dataset 1.

Conditional GAN The conditional gan trained on the 5000-image dataset converges, but the test loss is larger than the training loss. When trained on the smaller dataset with 723 images, the training loss declines rapidly, and ultimately reaches a smaller loss than when the Gan is trained on the larger dataset. (Figure 9)

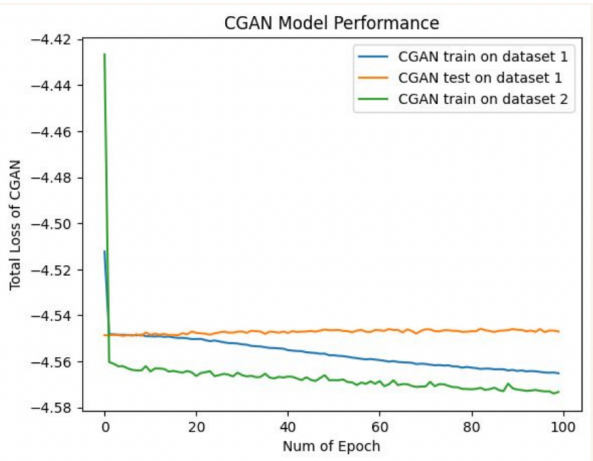

Figure 9. Loss plot for Conditional GAN on dataset 1 and 2.

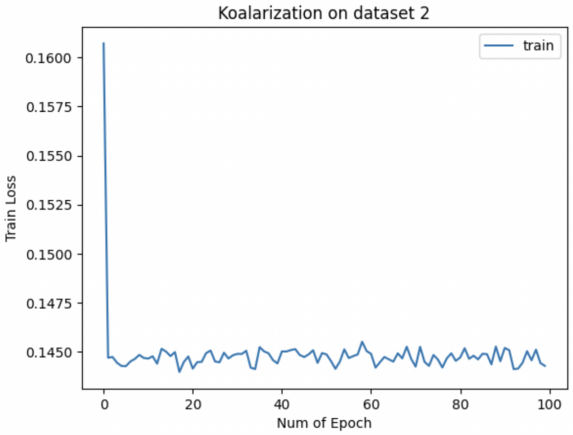

Figure 10. Loss pot for deep CNN on dataset 2.

4.2. Socially-responsible Computing Discussion via Proposal Swap

Accuracy of our auto-colorization We acknowledge that it would be a great idea to convert our auto-colorized images back to gray-scale images, and to test the colorization accuracy by comparing these gray-scale images to the original gray-scale images. This would require more time and resources to complete, and it is also involved with cyclic GAN, which is more complicated. Currently we do lack a formal way of testing the accuracy, and have only been able to quantify in terms of the losses but not the actual colorization visualization results.

Potentially misleading representations of cultural events As mentioned in Section 4.1.2, our models are not performing too satisfactorily when it comes to complicated objects and scenes with rather complex structures and patterns. Thus, we believe that the results and models in this paper are not yet good enough to

colorize cultural artifacts. But if time permits, we will improve our model and increase their accuracy. This improved accuracy, in turn, will also make it less possible for marketers to misrepresent cultural events.

Colorization may distort the actual event and harm integrity Taking this point into consideration, we will mark all images that are produced by our auto-colorization models as colorized from original gray-scale images. We will also ask for permission from the photographers and owners of black-and-white images before we apply auto-colorization.

5. Conclusion

Our project demonstrates the potential of deep learning techniques in image coloring. Through our exploration of different model architectures and datasets, we have not only achieved promising results but also compared the performances among models.

Image colorization is a highly subjective, challenging task, and there is still much space for improvement. Future work could focus on refining the model architecture, finetuning it on more representative datasets, or even integrating user inputs for more customized results.

The image colorization technique can be applied in various fields, such as film restoration, medical imaging, etc. It underscores the transformative potential of deep neural networks in the field of computer vision.

References

[1] Saeed Anwar, Muhammad Tahir, Chongyi Li, Ajmal Mian, Fahad Shahbaz Khan, and Abdul Wahab Muzaffar. Image colorization: A survey and dataset, 2020. 2

[2] Federico Baldassarre, Diego González Morín, and Lucas Rodés-Guirao. Deep koalarization: Image colorization using cnns and inception-resnet-v2, 2017. 1

[3] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. 2

[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)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. 2

Appendix

Team contributions

Wanjia (Julia) Fu Built U-Net model with Linghai Liu, and contributed to the development of Conditional GAN, with a primary focus on the generator, discriminator, and the combination of the two networks. Contributed to the ”Observation/Insights” part of the presentation poster, and completed the abstract and results of this report.

Zixuan (Shania) Guo Contributed to the development of the Deep-koalarization model and the Conditional GAN model, with a primary focus on the implementation and optimization of the training and testing functions. Help compile, categorize, and assemble the outcomes derived from the models, and generate visual representations (images) to display them. Formatted and edited the poster, and contributed to the ”Motivation”, ”Problem”, ”Results” etc. parts of the presentation poster. Created the Deep-Koalarization model diagram and completed the Introduction part of this report.

Linghai Liu Focused mostly on model architectures, methodology, and training. Answered questions and difficulties by other team members. Wrote the U-net and participated in debugging Deep Koalarization and conditional GAN. Wrote shell scripts, set up virtual environments, and trained the models on the computing clusters. Wrote ‘method’ part of project report and drew model diagrams on the poster.

Matianyu (Yuki) Zang Played a role in establishing the model architecture, which encompasses the Encoder, Decoder, and Feature Extractor components of the Deep-koalarization model, as well as the Discriminator in the Conditional GAN model. Dedicated effort to quantitatively evaluate the model performances via visualizing the training and testing loss through graphical plots. Contributed to the drafting of the ’Related Work’ and ’Conclusion’ sections of the project report.