A Review of Gray-scale Image Recoloring Methods With Neural Network Based Model

Yuyang Wang[1], *Student Member, IEEE*, Tianli Ren[1†], Zhuoyi Zhang[1‡], Zheyu Chen[1§],
Jinfeng Xu[1¶], Zixiao Ma[1‖], Pai Zhu[2**], Mingjia Zhang[1††]
[1]Beijing-Dublin International College,
Beijing University of Technology, Beijing, China & University College Dublin, Dublin, Ireland
[2]Department of Computer Science, Northeastern University at Qinhuangdao
[*]yuyang.wang821@gmail.com, [†]rtl.weshare@gmail.com, [‡]zhuoyi.zhang@ucdconnect.ie [§]czy5722025@163.com,
[¶]jinfeng.xu@ucdconnect.ie, [‖]zixiao.ma@ucdconnect.ie, [**]201917196@stu.neu.edu.cn, [††]mingjia.zhang@ucdconnect.ie

2022 7th International Conference on Image, Vision and Computing (ICIVC) | 978-1-6654-6734-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIVC55077.2022.9886935

***Abstract*—Gray-scale image recolor or computer-assisted color restoration is a computer-assisted process of adding color information to grayscale photographs, which is crucial in perceiving visual information by humans. An important feature of this technique is the ability to restore the original style of old images. Neural network-based black and grayscale image coloring technology is gradually becoming a prevalent topic in current research with the development of computer vision and data science. Models with DNN, CNN, GAN, and DCGAN were successively proposed in the recent few years, indicating a fresh-new hot spot of image processing. Several popular image recoloring methods have been discussed in this essay, including a review of recent research and then a detailed analysis of the current deployment method. Finally, the essay compares the existing application, demonstrating several real-world examples and providing information and statistics on how to apply this technique to the industry. We also summarized current limitations and future trends of this research field to provide an outlook for other researchers.**

***Keywords*—Image Color Restoration, Image Processing, Machine Learning, Computer Science.**

I. Introduction

The first color photograph was taken in 1861 by Maxwell (1831-1879) in England. However, due to the fact that the film was expensive, color photographs were prevalent around the 1960s, and the vast majority of photographs were black until the 1960s. Although no recolored photograph can replace an old black and white photograph, a recolored photograph will give a whole different perspective and experience. Photos are people's memories, and these memories can be made even better if the grayscale image could be colorized.

Image recoloring is an image-to-image translation problem that maps a high dimensional input to a high dimensional output. Nazeri argued that it can be seen as a pixel-wise regression problem where structure in the input is highly aligned with structure in the output. [1]

Models for the recoloring of grayscale images began back in the early 2000s. In 2002, an algorithm were proposed which recolored images through texture synthesis. Image recoloring was achieved by matching the texture and luminance of an existing image and the original image. However, this method requires significant user intervention which makes it unable to be an ideal way of recoloring. [2] Nowadays, with the development of generative network, using two sub-nets challenging in the model (generator and discriminator) [1] were proposed (See II-B). The image generated using this technique does not need human intervention, which is an more appropriate way of image recoloring.

II. Recoloring Technique

The recoloring process based on black and white images can be seen as a pixel-based point-to-point prediction process. The input and output are high dimensional data, which is the difficulty of this problem. [3] The mapping function of the high-dimensional data is the parameter to be learned by the generative model. The central problem of this research is that the number of color channels for the input data is 1 and the number of color channels for the output data is 3. Therefore, the problem becomes a prediction task from the lower dimension to the higher dimension. Therefor processing of the color channel is the key point of this problem. In addition to this, the commonly applied color incorporation method cannot be used directly on this problem using the conventional form of direct prediction, which is because in restoring process, the color does not vary drastically from pixel to pixel, so a method that takes care of the contiguous pixel is required to synthesize the distribution of pixels around a particular pixel in order to acquire a correct and reasonable coloring.

A. CNN Based Models

The CNN model is the basis of the GAN based models (which will be discussed in II-B) because each CNN model has the most important component with the GAN model which is the convolution operations, therefore, each pixel in the output can be seen as a regression problem of the input pixel. CNN has been widely used in the image classification problem. However, this technique can be also applied to solve the task of image color restoration. Some research [4] create two CNN networks for recoloring task, which includes two variations of the user interaction recoloring network shown in *Fig.1*. Both options forecast a colorization using the blue layers. Additionally, the Local Hints Network use red layers

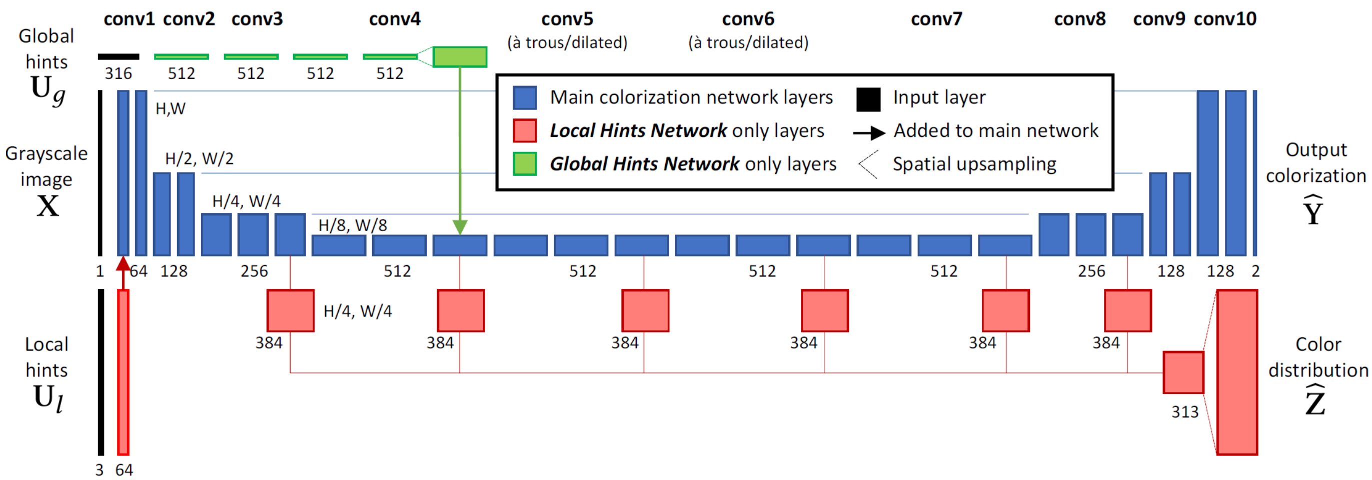


Fig. 1. CNN based Network Architecture [4]

Fig. 2. Image Generated by CNN based Network [4]

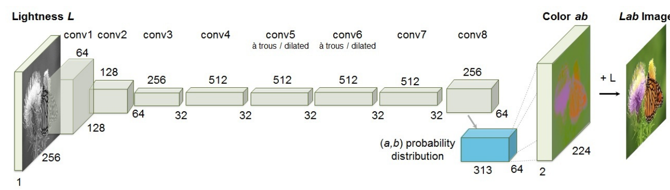


Fig. 3. Color Distribution Based CNN Model [5]

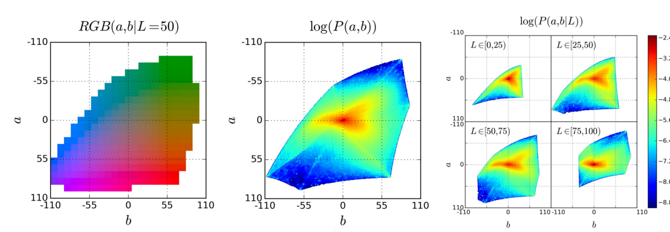


Fig. 4. Quantized (a, b) color space with a grid size of 10. [5]

to include user points and anticipate color distribution. The Global Hints Network incorporates the green layers, which convert global input using 1×1 conv layers, into the primary recoloring network. Each box represents a conv layer, with the vertical dimension being the spatial resolution of the feature map and the horizontal dimension representing the number of channels. Through sub-sampling and up-sampling, changes in resolution are accomplished. When resolution is lowered on the primary network, the number of feature channels doubles. In up-sampling convolution layers, shortcut connections are introduced. A series of generated result has been shown in *Fig.2*

The network anticipates user-intended behaviors based on learning semantic similarities, which is an advantage of this research. Nonetheless, the network might sometimes be too optimistic and generate undesirable non-local consequences.

While another research [5] modifies the output layer, where there is normally a sigmod or other classifier, they applied a (a, b) probability distribution to output *Fig.3*, which allows the model to give color distribution of recolored image. This model will be input with a series of sample images to learning the mapping of the color and finally use the leaned probability to estimate new images. In other words, the research treat the problem as polynomial classification. They quantize the output space into bins with grid size 10 and keep the $Q = 313$ which are in-gamut.

For a given input $\mathbf{X}$, the model learn a mapping $\widehat{\mathbf{Z}} = \mathcal{G}(\mathbf{X})$ to a probability distribution over possible colors $\widehat{\mathbf{Z}} \in [0, 1]^{H \times W \times Q}$, where $\mathbf{Q}$ is the number of quantified a, b values. This is the core logic of using CNN to colorize images. [5] The distribution of (a, b) are demonstrated in *Fig.4*

The most important contribution of this research is that they replaced the CNN model's output layer with a probability distribution function instead of using it directly for prediction. Thus, the direct effect of outliers on the output results is avoided, which improves the training efficiency and model accuracy.

For the evaluation, the team conducted a Perceptual realism (AMT) experiment, in which the model algorithm fooled

Fig. 5. Generated Result of Color Distribution Based CNN Model [5]

participants on 32% of trials, which is a good result for the image colorizing task. A sample of generated images has been shown in *Fig.5*.

B. Generative Adversarial Networks (GANs)

In GANs, there are two parts of the network – generator and discriminator. The generator and discriminator are both convolutional neural networks (CNNs) (See II-A). In mathematics, the generator is represented as $G\left(z;\theta_G\right)$ and the discriminator is represented as $D\left(z;\theta_D\right)$, where z is the noise. The model will converge when the classifier can tell the real class of the generated data. This optimize process can be expressed as *Eq.1* and *Eq.2*, where x is colored image.

$$\min_{\theta_G} J^{(G)}\left(\theta_D,\theta_G\right)=\min_{\theta_G}\mathbb{E}_z[\log(1-D(G(z)))] \tag{1}$$

$$\begin{aligned}\max_{\theta_D} J^{(D)}\left(\theta_D,\theta_G\right)=\max_{\theta_D}(&\mathbb{E}_x[\log(D(x))]\\ &+\mathbb{E}_z[\log(1-D(G(z)))])\end{aligned} \tag{2}$$

The network purposed by the University of Ontario's research team [1] has an asymmetric shape, which has n encoding units and n decoding units. Using a conditional Deep Convolutional Generative Adversarial Network (DCGAN), they seek to generalize the recoloring technique completely the baseline of their model has been shown in *Fig.6*.

The design was updated to be a conditional GAN rather than a conventional DCGAN; they also adhere to the guidelines and only offer noise in the form of dropout, which is applied to many layers of their generator. The structure of generator G is identical to that of the baseline. For discriminator D, they employ a similar design to that of the baseline's contractive path: a sequence of 4×4 convolutional layers with stride 2, with the number of channels doubling after each down-scaling.

The team measures the result by using mean absolute error (MAE) ans they computed the error pixel by pixel, treated the color RGB as vectors, and calculate the distance between them. The result shows their model achieved 5.1 MAE which is better than the 7.1 of the baseline U-Net.

Another model is called HistoGAN [6], a color histogram-based technique for managing the hues of GAN-generated pictures. They concentrate on color histograms because they

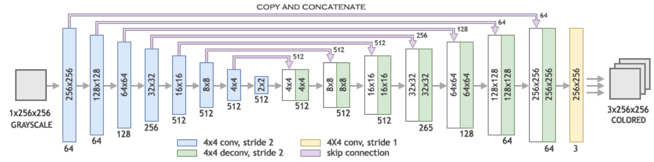


Fig. 6. U-Net Baseline [1]

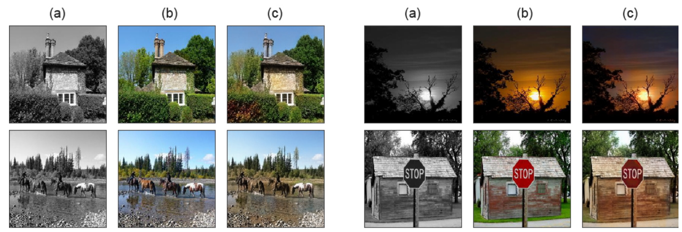


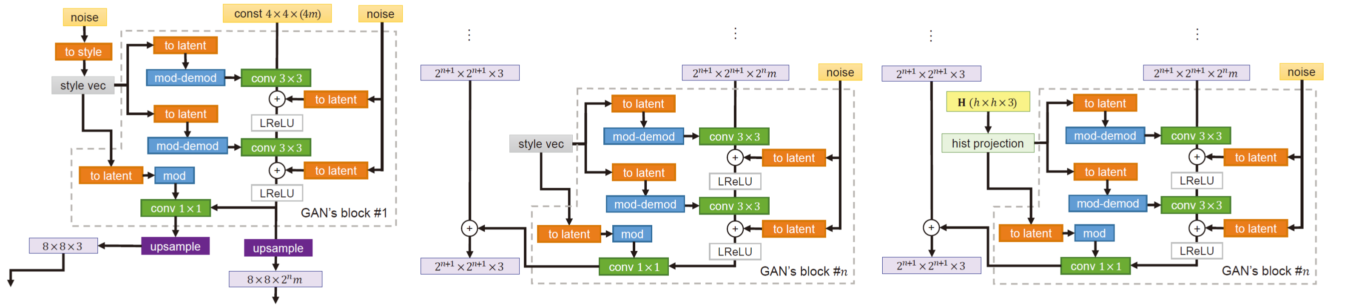
Fig. 7. Recoloring Result of GAN Based Model [1]

offer an easy approach to represent picture color while retaining domain-independent semantics. In particular, they provide an efficient adaptation of the current StyleGAN architecture for controlling the colors of GAN-generated pictures using a target color histogram feature shown in *Fig.8*. The authors then explain how to expand HistoGAN to recolor actual photos. They jointly train an encoder network and HistoGAN for picture recoloring. The recoloring model, ReHistoGAN shown in *Fig.9*, is an unsupervised technique taught to encourage the network to maintain the original image's content while modifying the colors depending on the provided target histogram. They demonstrate that their histogram-based strategy provides a superior method for controlling the colors of GAN-generated and actual pictures, while providing more attractive outcomes than existing alternatives.

C. NoGAN (A highly effective way of training)

NoGAN is technically not a model, but an effective way of training GAN models (See II-B), which was purposed by the DeOldify *(A GAN-based image color restoration project)*. Especially effective when using GAN to deal with tasks related to images and videos. It combines the benefits of traditional models and GAN models. This project has received 14.2k stars since its release. Examining the difference between this model and how this project solved this image recoloring problem is necessary.

The main idea is to pre-train the generator and discriminator separately and then perform a final adversarial training step as in a standard GAN. [7] In this case, the generator is initially trained with some grayscale images (i.e. feature loss) and then uses the generated fake images to train the discriminator as a binary classifier. The generator receives the grayscale images as input and uses its perceptual losses to generate more colorful images; the discriminator is then trained to distinguish the original color images from the recovered ones. Finally, regular GAN training is performed to train the discriminator and generator alternately.



(A) Simplified version of StyleGAN's first block (B) Simplified version of StyleGAN's last block (C) HistoGAN block

Fig. 8. StyleGAN and HistoGAN Block [6]

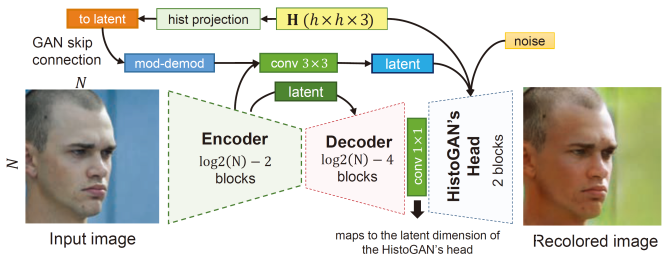


Fig. 9. Recoloring-HistoGAN (ReHistoGAN) [6]

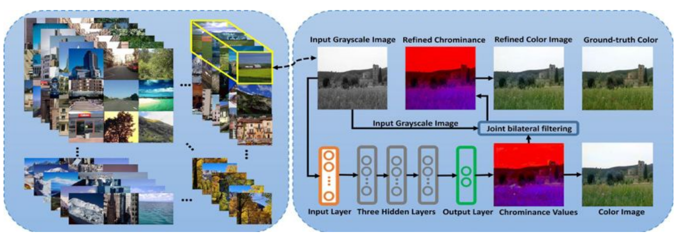


Fig. 10. Colorizing Overview(DNN) [8]

D. Deep Neural Networks (DNN)

DNN-based coloring is an entirely automated coloring technique. It redefines coloring as a matching issue, makes greater use of global picture data, and incorporates adaptive image clustering approaches. This strategy employs a post-processing method based on combined bilateral filtering to ensure distortion-free results. *Fig.10* depicts a visualization utilizing a vast reference picture collection and a coloring approach based on deep learning. The process of recoloring involves two stages: (a) Training deep neural networks (DNNs) for each class requires reference photos categorized into different groups by adaptive image clustering algorithms. (b) The coloring technique and DNN architecture. Given a grayscale, the closest class is automatically researched and the DNN is trained appropriately. At each pixel, feature descriptors are retrieved to be used as neural network input. The weights gained from the reference picture database are linked to the connections between neuron pairs. The output is the chromaticity of the corresponding pixel, which may be mixed directly with grayscale to produce the relevant color value. The chroma computed by the training model may be rather noisy in low-texture regions, but this noise may be minimized by a bilateral joint filter (the input grayscale image is used as a guide). [8]

III. Deployment

At present, there are two major deployment schemes for neural network models, which are client-side deployment and server-side deployment, and each of the two approaches has its advantages. The server-side deployment is characterized by strong computational power and high stability but can cause a long waiting time for users due to the network latency. [9] In contrast, client-side deployment avoids the network transmission process and reduces server maintenance costs, but cannot perform complex operations due to limited device performance. In the following, this essay provides an overview of the two different deployment scenarios. The comparison of hash rates for two different deployment scenarios are shown in the *Table.* I.

A. On Device Processing

One of the most popular cross-platform solutions for deploying the model to the device is using TF-Lite. [10]

1) Advantages:

- Lightweight. Reduce the size of the runtime library and model, reduce memory consumption, and work with more devices.
- High performance. Deeply optimized for mobile and IoT devices, it accelerates machine learning with a wide range of hardware, such as the latest ARM CPU commands, GPUs, DSPs, and NPUs. [11]
- Cross-platform and high compatibility. Support Android, iOS, embedded Linux and MCU, and other platforms, and support a variety of ”one-time switch, deploy anywhere”.
- Easy to use. It provides rich platform-related APIs, a rich model library, complete examples and documentation, and a rich toolchain to lower the threshold of developers.

TABLE I
HASHRATE DIFFERENCE BETWEEN THE TWO DEPLOYMENT MODES

Type	GPU Version	Year	Collaboration	FP32 (TFLOPS)	FP16 (TFLOPS)	Memory bandwidth (GB/s)
Terminal device	Adreno 430	2015	No	0.333	0.666	25.6
Terminal device	Adreno 540	2017	No	0.568	1.135	29.8
Terminal device	Adreno 630	2018	No	0.727	1.454	29.8
Central server	Tesla V100	2017	Yes	14	112	900
Central server	RTX Titan	2018	Yes	16	130	672

TABLE II
COMPARISON OF DEOLDIFY RELATED WORKS

Type	Time Cost	Price	Complexity
Open API	8.33s	Free	Medium
MyHeritage	18.24s	Need Subscription	Easy to use
Google Colab	4.32s	Free	Require Professional Knowledge

- Extensible. Modular, easy to customize, easy to expand to more hardware support, customize more operators

2) Disadvantages:

- A limited number of operators supported to perform complex mathematical operations.
- Even with software optimization, mobile computing performance is still low compared to GPU servers and is not suitable for a large batch of complex computing.
- Machine learning API support in the current Android system coverage is low, some low versions of Android can not run the model.

B. Server Side Processing

The server-side deployment is much more complex than the on-device model deployment [12], and the online ML service should have at least 4 parts of services to give the client a correct result in a reasonable time.

1) Web API Server: Web API service provides a communication interface between the server and client, and it should be responsible for receiving the request and sending back the result while checking the user's permission.

2) GPU Compute Services: Nearly all the machine learning models are running on GPU. A GPU cluster is needed to provide services.

3) Storage Services: The uploaded files should be able to store somewhere temporally for processing.

4) Database Services: User authorization should be done by querying the database and the application-related information should also be stored in a database.

IV. APPLICATIONS

Currently, some similar applications have been released in the application market or presented in the form of websites. Nearly all the architecture they use is a full server-side processing and client-side display (See III-B). Some of them are basically prototypes, and only a few of them have well-designed user interfaces.

A. Web Sites

1) Deep AI DeOldify Project: This project offers three different ways of using the model.

The Comparison of its related work has been shown in *Table.* II

TABLE III
COMPARISON OF IMAGE COLORIZATION MOBILE APPLICATIONS

App Name	Developer	GUI Design	Sales(1 Month)	Rating	Price
Colorize Images	Colorize Images	Basic	1,000,000+	4.0	Watching Ads for Free processing
Colorize	Photomyne Ltd.	Fancy	500,000+	4.6	Monthly Subscription
AI Colorize....	KallosSoft	Plain	10,000+	No Enough Data	Monthly Subscription

OpenAPI provides a relatively fast and free API service, but its use still requires configuring tokens and constructing requests, thus raising the threshold of use to some extent.

MyHeritage is a commercial site that provides a relatively complete set of image restoration work, but its subscription fee is high and not suitable for small-scale use by ordinary users.

Google Colab provides a machine learning model notebook, which simplifies most of the model loading part, but still requires a certain degree of expertise to use.

B. Mobile Applications

Nearly all the on-sale applications are using server-side image processing pipelines (See III-B), the difference has been shown in *Table.* III

1) Colorize Images (Author: Colorize Images): This application uses a server-side deployment scheme and its interface design is relatively simple. However, since its charging model is to use ads to redeem the credit, it is more popular among users. Because of the simple interface design, the rating is relatively low.

2) Colorize (Author: Photomyne Ltd.): This app uses the same model deployment method as the previous application. The software is designed with the user's needs in mind and has a comprehensive video tutorial for use, resulting in a higher number of downloads and ratings.

3) AI Colorize & Restore Old Photo: Fix Damaged Image (Author: KallosSoft): This application is optimized mainly for the editing of images. After coloring the image with this software, the application can continue cropping the image, filtering, or drawing.

C. Media Field

Medical images provide valuable anatomical information for clinical operations, and color images enhance monochromatic medical models. It can improve the contrast of anatomical structure and facilitate accurate segmentation.

Visual information can provide significant assistance to doctors in analysis and disease diagnosis. Colour images can provide better support for professionals when the structure of the image remains the same, as shown in the *Fig.11*. Automatic detection of disease areas based on colour images may be possible in the future.

V. LIMITATION AND FUTURE DEVELOPMENT

The most significant difficulty of recoloring is not how to color but is to choose which color to draw on the canvas. For many photos that were shot a few decades ago, it is difficult to restore the original color atmosphere by relying on stills, posters, and other limited information hints.

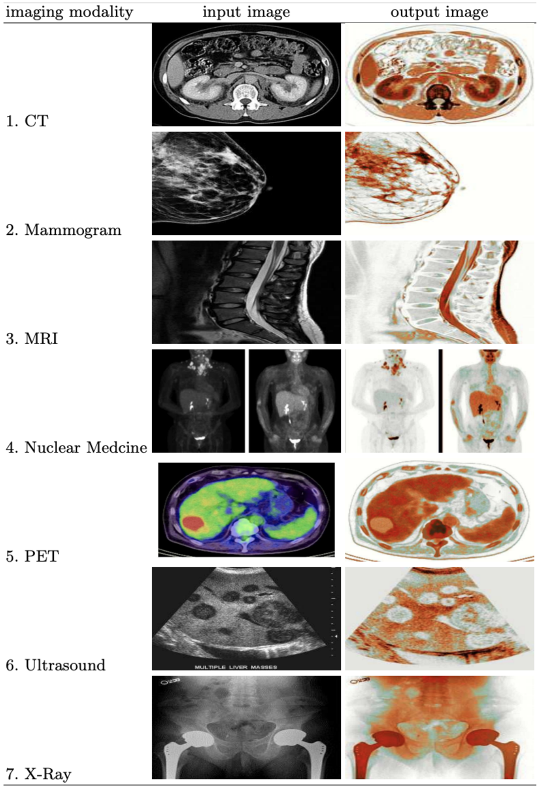


Fig. 11. Input Images and Colorization Result [13]

In studios where experts manually restore the old color of the image, they are making a secondary creation but not the original one. Unsupervised coloring or computer-aided recoloring is a convenient technique, but without local cues and other methods to optimize coloring, the coloring results may cause these problems. [14] For large-scale commercial image color restoration projects, the high cost of manual tagging is significantly high, so it is difficult for this type of technique completely replace manual recoloring. But some researchers are already making progress in dealing with this problem, for example, real-time user-guide image recoloring has been purposed. For public use of recoloring technique, users can't label a specific area of an image, but the quality of the restoration is not as high as the commercial use, little inconstancy is acceptable. Thus, the unsupervised coloring technique is applicable in this scenario.

VI. Conclusion

The greatest value of color photographs is the spiritual shock of returning to the historical scene. [15] As color photos correspond to our visual perception, images with color resonate more directly with our hearts than black and white photographs. At present, the coloring model based on GAN, CNN, and DNN is becoming stable, and the coloring result can reach the expectation of most users. The model can be deployed using client-side deployment or server-side deployment, which means that the technical route is clear. At the same time, existing applications have high ratings and downloads, which proves that this area has a broad prospect and is worthy of future development work.

References

[1] K. Nazeri and E. Ng, "Image colorization with generative adversarial networks," *CoRR*, vol. abs/1803.05400, 2018.

[2] Z. Zhu, M. Toyoura, K. Go, K. Kashiwagi, I. Fujishiro, T.-T. Wong, and X. Mao, "Personalized image recoloring for color vision deficiency compensation," *IEEE Transactions on Multimedia*, vol. 24, pp. 1721–1734, 2022.

[3] Q. Zhang, Y. Nie, L. Zhu, C. Xiao, and W.-S. Zheng, "A blind color separation model for faithful palette-based image recoloring," *IEEE Transactions on Multimedia*, vol. 24, pp. 1545–1557, 2022.

[4] 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," *ACM Transactions on Graphics (TOG)*, vol. 9, no. 4, 2017.

[5] R. Zhang, P. Isola, and A. Efros, "Colorful image colorization," in *Colorful Image Colorization*, vol. 9907, 10 2016, pp. 649–666.

[6] M. Afifi, M. A. Brubaker, and M. S. Brown, "Histogan: Controlling colors of gan-generated and real images via color histograms," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7937–7946.

[7] F. Mameli, M. Bertini, L. Galteri, and A. Del Bimbo, "A nogan approach for image and video restoration and compression artifact removal," in *2020 25th International Conference on Pattern Recognition (ICPR)*, Jan 2021, pp. 9326–9332.

[8] S. B. Cheng Z, Yang Q, "Deep colorization," *Proceedings of the IEEE international conference on computer vision.*, pp. 415–423, 2015.

[9] P. B. Dasgupta, "A parallelized approach for colorizing grayscale images," *International journal of computer trends and technology*, vol. 67, no. 10, pp. 69–72, 2019.

[10] L. Shuangfeng, "Tensorflow lite: On-device machine learning framework," *Journal of Computer Research and Development*, vol. 57, no. 9, p. 1839, 2020.

[11] Z. HANLI and J. ZHIJIAN, "Grayscale image colorizing method based on gpu acceleration," 2017.

[12] M. Zaharia, A. Chen, A. Davidson, A. Ghodsi, S. A. Hong, A. Konwinski, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe *et al.*, "Accelerating the machine learning lifecycle with mlflow." *IEEE Data Eng. Bull.*, vol. 41, no. 4, pp. 39–45, 2018.

[13] M. U. G. Khan, Y. Gotoh, and N. Nida, "Medical image colorization for better visualization and segmentation," in *Annual conference on medical image understanding and analysis*. Springer, 2017, pp. 571–580.

[14] S. Boutarfass and B. Besserer, "Improving cnn-based colorization of b and w photographs," in *2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS)*, 2020, pp. 96–101.

[15] G. R. Kuhn, M. M. Oliveira, and L. A. F. Fernandes, "An efficient naturalness-preserving image-recoloring method for dichromats," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1747–1754, 2008.