Image Colorization using Deep Learning: A Review

Aayushi Beniwal, Sai Anurag Neelisetty, and Uttej Reddy Pakanati
abeniwal@uwaterloo.ca, saneelisetty@uwaterloo.ca, urpakanati@uwaterloo.ca

***Abstract*—Image colorization is an image-to-image translation problem. It can be considered a pixel-wise regression problem where structures in inputs and outputs are highly aligned. Automatic colorization is an active area of research because of its wide variety of applications. There are several techniques to solve the problem, and the colorization networks used are classified into several categories. These different categories are discussed in the introduction. This paper aims at studying the state-of-the-art algorithms implemented for the image colorization problem. The work reviews and summarizes three deep learning models with plain network architecture. The inputs to these networks are natural images, and the output is a single image. The primary focus of the papers is on automatic colorization without human intervention.**

***Keywords*—Colorization review, convolutional neural networks, deep neural networks, generative adversarial networks, image colorization**

I. Introduction

Image colorization is the process of assigning a color to each pixel of the target grayscale image. Colorization of images is a demanding problem due to the differing conditions of imaging that need to be dealt with a specific algorithm. This is a complicated problem because two out of the three image dimensions are absent; although the scene semantics may be beneficial in many cases, for example, the grass is generally green, clouds are mostly white, and the sky is blue. Nevertheless, such semantic priors are highly rare for many artificial and natural objects, e.g., shirts, cars, flowers, etc. With the swift progress of deep learning techniques, various image colorization methods have been proposed, and state-of-the-art performance on existing datasets has been reported.

Colorization networks differ in many significant aspects, including the network depth, network architecture, learning strategies, loss functions, etc. As shown in Figure 1 [1], these colorization networks have been categorized into several groups based on the input type, user guidance, structural differences, etc. The plain networks have a simple, straightforward architecture that stacks layers without skip connections. Deep colorization, Colorful colorization, and Deep depth colorization are few techniques which use plain network architecture. In this paper, we focus on how different deep learning models with plain network achieve colorization. Their architecture, experimental results, limitations are discussed and are compared with other state-of-the-art colorization methods.

II. Deep colorization

Deep colorization is one of the many methods of coloring a grayscale image. The paper [2] formulates image colorization as a regression problem, and deep neural networks were employed to solve the problem. Image colorization techniques can be divided into two categories: scribble-based colorization and example-based colorization. The scribble-based methods typically require significant efforts from the user to provide large scribbles on the target grayscale images. The example-based method typically transfers the color information from a comparable reference image to the target grayscale image.

Deep colorization is an example-based colorization method that has two major steps: In the training step, the global descriptors of the reference images (e.g., tree, building, sea, mountain, etc.) are first extracted, these images are then grouped into different clusters, and the semantic histogram of each cluster is computed. Then the feature descriptors at sampled pixels and corresponding chrominance values are calculated. A deep neural network is then constructed and trained using the training set. In the testing step, the global descriptor and the semantic histogram of the target grayscale image are computed, and then the nearest cluster center is found. Then, the feature descriptors at each location are extracted and then sent to the trained neural network, and corresponding chrominance values are obtained. The chrominance values are then refined and combined with the grayscale image to get the color image.

A. Architecture

Deep neural networks (DNNs) usually consist of one input layer, multiple hidden layers, and one output layer. In this model, the number of neurons in the input layer was set to the number of dimensions of the feature descriptor extracted from each pixel location in a grayscale image, and the output layer has two neurons. In the hidden layer, the number of neurons was set to half of that in the input layer. Every neuron in the hidden or output layer is linked to all the neurons in the previous layer, and each connection is associated with a weight. The output of the neurons in the output layer is the weighted combination of the outputs of the neurons in the previous layer. ReLU activation function was used to speed up the convergence of the training process.

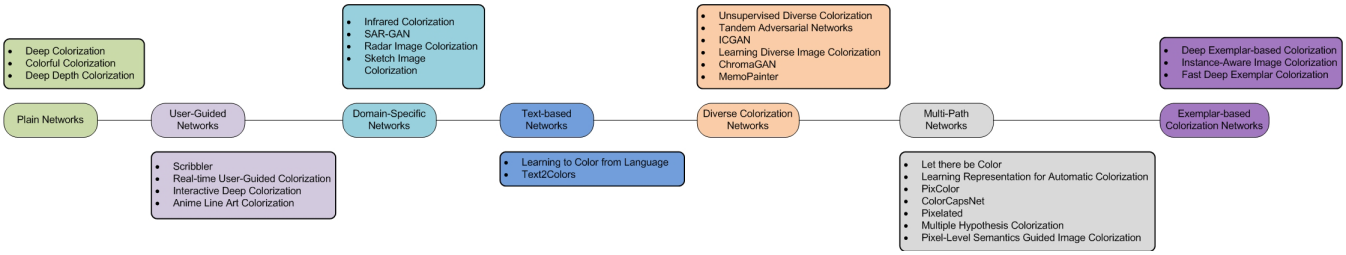

Fig. 1. Classification of the colorization networks based on structure, input, domain, and type of network.

Classical error back-propagation algorithm was used to train the network and the weights between a pair of neurons are learnt.

B. Feature Descriptor

Feature design was crucial to the success of the deep colorization method. There were enormous candidate image features that affected the efficacy of the trained model (e.g., SIFT, SURF, Gabor, Location, Intensity histogram, etc.). Numerous experiments were conducted to test various features, and only the features that have practical impacts on the colorization results were kept. The adopted features were separated into low, mid, and high-level features and are concatenated to construct the feature descriptor. To ensure artifact-free quality, a joint bilateral filtering method [3] was used to smooth/refine the chrominance values with the target grayscale image as a reference and with that most of the visible artifacts were successfully removed.

C. Adaptive Image Clustering

The reference images were clustered adaptively on distinct layers by a standard k-means clustering algorithm. After completing the training of DNN for the cluster on a layer, the negative Peak Signal-to-Noise Ratio (PSNR) is computed from the colorization result and the ground truth image for each reference image as error. If the error was lower than a threshold, the reference image was removed from the set. As a result, the top layer contained all the reference images, whereas the lower layer contained few images. To address the problem of reference images in the searched cluster globally similar but semantically different from the target images, a sematic histogram is incorporated to search for the globally and semantically identical cluster.

D. Evaluation

The PSNR distribution of 1519 test images with/without image clustering was compared, and it was noticed that the image clustering technique has improved the colorization accuracy and reduced the visible artifacts significantly, especially for the objects with large color variances.

E. Differences from the state-of-the-art methods

The paper [4] proposes an automatic colorization framework like the deep colorization model, and the following are the differences – 1) While [4] takes 251.709 and 712.149 seconds to colorize 256 × 256 and 512 × 512 images respectively, the deep colorization model takes only 6.780 seconds and 17.413 seconds to colorize 256 × 256 and 512 × 512 images respectively. 2) Model in [4] requires an appropriate scene histogram in the refinement step. However, no spatial prior is required for the deep colorization model. 3) The deep neural networks learn the mapping function automatically instead of designing the objective function carefully by hand or searching for massive hyper-parameters like [4]. [1] also states that numerous experiments demonstrated that deep colorization method outperformed the state-of-the-art algorithms both in terms of speed and quality.

F. Limitations

Deep colorization is fully automatic and thus usually is more robust than the conventional methods. Nevertheless, it depends on machine learning techniques and has its own constraints. For instance, the model should be trained on a vast reference image database that contains all potential objects which is not possible in practice. For instance, the deep colorization model in [1] was trained on natural images and thus is invalid for the synthetic image. Color to grayscale transformation will result in the loss of color information. However, this is a limitation to all state-of-the-art colorization techniques.

III. Image Colorization using Generative Adversarial Networks

The paper [5] discusses a technique known as Generative Adversarial Network, designed by Goodwell and his colleagues in 2014 [6]. Generative Adversarial Networks, shortly called GANs, have become popular and are widely applied to various problems. In this paper, a Deep Convolutional Generative Adversarial Network (DCGAN) was used to solve the problem of image colorization, public datasets Places365 and CIFAR-10 were used.

Colorizing grayscale images started back in the early 2000s. Welsh et al. [7] proposed an algorithm in 2002 that colorizes images based on texture synthesis. In 2004, Levin et al. [8] proposed another algorithm with a cost function that penalizes the difference between a pixel and the weighted average of its neighboring pixels. These proposed solutions are not fully automatic, required user intervention, and are not considered ideal. Isola et al. [9] proposed a

method that uses a special type of network, general adversarial networks (GANs), and compared the differences in colorization between traditional convolutional neural networks and GANs. This paper [5] draws inspiration from [9] and aims to improvise by generalizing the procedure to high-resolution images and suggesting strategies to speed up the training process and stabilize it.

A. Generative Adversarial Networks:

GANs are composed of two networks – a generator and a discriminator. The generator produces fake images, and the discriminator classifies whether they are real or fake. The discriminator is given both an actual image from the dataset and a generated image from the generator. The generator learns and produces realistic images to fool its opponent – the discriminator and hence the name adversarial networks.

Both the networks are convolutional neural networks as we are dealing with images. The cost functions of generator G and discriminator D are discussed. The cost function of G is designed so that the probability of D being mistaken is maximized, and the cost function of D is designed such that the probability of assigning the right label is maximized.

Input to the generator is a randomly generated noise vector. But for the image colorization problem, the input has to be a grayscale image, so the general GAN cannot be used. A special variant called Conditional Generative Adversarial Network (CGAN) was introduced. In a CGAN, both the generator and the discriminator are conditioned on some information such as class labels. The generator is added with an additional parameter like the class label, hoping to generate such an image. The discriminator is also provided with the label to distinguish real images better. As the inputs of both D and G are changed now, the cost functions are also modified.

B. Architecture

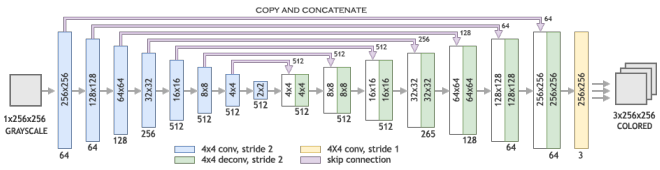

Fig 2. U-Net architecture

Baseline Network: U-Net is a symmetrical architecture model introduced in 2015. It was first used for biomedical image segmentation. It contains a contracting path resembling an encoder and an expansion path resembling a decoder. The contracting path follows typical CNN architecture. This model architecture was considered for the baseline model in the paper. L*a*b* color space was used for the colorization process.

GAN Network: Based on the guidelines for Deep Convolutional GAN (DCGAN) in [10], the generator and the discriminator networks were designed with CNNs. It was modified to be a conditional GAN instead of a traditional DCGAN. The architecture of the generator was the same as the baseline, and for the discriminator, the contracting path of the baseline model was followed. The discriminator architecture contained several 4*4 convolutional layers with stride 2. The number of channels was doubled after each downsampling. Leaky ReLU was the activation function and batch normalization were also used. Hyperparameters like the Adam optimizer and learning rate were mentioned. There are specific problems with GANs like non-convergence and mode-collapse. Model parameters destabilize, oscillate, and do not converge. The generator collapses and does not provide varieties of output. Training constraints and strategies to overcome these problems include, introducing batch normalization, using alternative activation functions such as leaky ReLU, one-sided label smoothing, and reduced momentum were discussed.

C. Experiment Results:

Mean Absolute Error was used as a performance metric. Compared with images generated by CNNs, GANs produced vibrant images with clear visual improvement. One drawback with GAN was highlighted. It produced colors that are only present in the dataset that it was trained on. This shows that the dataset has to include all varieties of colors for a particular object. Also, images with high fluctuations are colored green, and the reason again was because of the images present in the dataset. Few colored images experienced the sepia effect, which may have been because of insufficient training. Training results are tabulated for U-Net (baseline model) and GAN networks. GAN model produced better results.

The paper was concluded by discussing other performance metrics for quantitative evaluation. Metrics such as root mean square error (RMSE) and peak signal-to-noise ratio (PSNR) would enable a robust process to quantify performance.

IV. COLORFUL IMAGE COLORIZATION

In the paper [11], grayscale photographs are taken as input and classified using class rebalancing at the time of training to boost colors. After training over a million images the feed-forward CNN is applied for testing. Colorization Turing test is then used to evaluate the algorithm. Further, a cross-channel encoder is used, which acts as a self-supervised feature learning increasing the performance. They anticipated the distribution of probable colors for each pixel to correctly describe the multimodal nature of the problem. At last, final colorization is produced by taking the annealed- mean of the distribution.

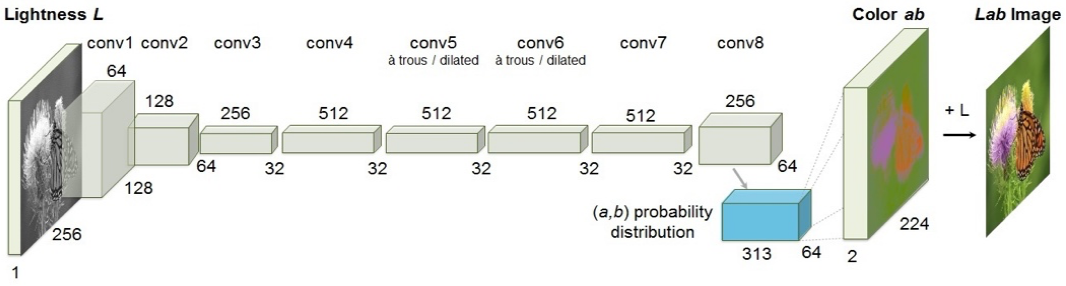

Fig 3. The network architecture for Feed Forward CNN

The contribution in this paper can be broadly divided into two areas. First, automatic image colorization is achieved by designing an objective function that takes care of the multimodal uncertainty of the colorization problem and captures a wide diversity of colors by training on millions of color photos. Second, the self-supervised representation learning method for colorization achieves state-of-the-art results.

A. Architecture:

The network used Feed Forward CNN where each convolution layer refers to a block of 2 or 3 repeated convolution and ReLU layers, followed by a batch normalization layer as shown in Figure 3. It has no pooling layers. All resolution adjustments were accomplished via spatial downsampling or upsampling between convolution blocks. Using this architecture, the model was trained to map from a grayscale input to the distribution of quantized color value outputs.

B. Approach:

Objective Function: CIE lab color space was considered. For the given input lightness channel, the objective is to learn a mapping to the two associated color channels a and b. A natural objective function, Euclidian loss between the ground truth and predicted colors was proposed in [2] but using this gave grayish and desaturated results. Hence, the problem is treated as a multinomial classification. A mapping to a probability distribution over possible colors was learnt for a given input. To compare predicted against the ground truth, they defined a function Z which converts ground truth color Y to vector Z, using a soft-encoding scheme and used multinomial cross-entropy loss.

Class Rebalancing: Because of the existence of backgrounds, the distribution of ab values in natural photographs is heavily skewed towards values with low ab values. The class imbalance problem was handled by reweighting the loss of each pixel at train time based on the pixel color rarity.

Class Probabilities to Point Estimates: By re-adjusting the temperature of the softmax distribution in the annealed-mean operation resulted in enhanced predicted distribution above all pixels.

C. Evaluation:

The network was trained on the 1.3M images from the ImageNet training set [12], validation was applied on the first 10k images in the ImageNet validation set and tested on separate 10k images in the validation set [13].

The model was tested with various loss functions. Quantitative performance metrics fail to capture visual realism and therefore qualitative methods are to be used. The techniques used were: 1) Perceptual realism: An experiment Amazon Mechanical Turk (AMT) was performed involving human participants to identify the image with fake colors when the real and fake images were shown side by side. The algorithm fooled 32% of the trials on participants. This percentage was significant compared to many algorithms but not as good as the model in [13]. 2) Semantic interpretability (VGG classification): Here, the fake colorized images are passed through a VGG network [14] that had previously been trained to predict ImageNet classes from genuine color photos. If the classifier works well, the colorizations are accurate enough to be useful in determining object class. After ablating colors from the input, classifier performance dropped from 68.3% to 52.7%. The re-colorizing using the full method helped improve the performance to 56.0%. However, the Larsson et al. [13] achieved 59.4% which is the highest. 3) Raw Accuracy (AuC): When L2 metric was used to fine tune color classification network, performance was better compared to optimizing it from scratch.

V. Conclusion

Under deep learning models with plain network architecture, we thoroughly reviewed these three approaches. The deep colorization model has surpassed the state-of-the-art algorithms [1] in terms of both quality and speed with few limitations. Adaptive image clustering method and joint bilateral filtering for artifact-free colorization quality makes deep colorization a best model. The Colorful colorization method used a feed forward CNN along with a chosen objective function to produce images that are very realistic. The representations the network learned can also be used for object segmentation, detection and classification and is not

just restricted for the image colorization. Generative adversarial networks performed better in producing realistic colored images compared to the traditional CNNs. GANs have become the go to architecture for such colorization problems. But there are issues with GANs and strategies that are mentioned above could help in achieving the best performance.

REFERENCES

[1] S. Anwar, M. Tahir, C. Li, A. Mian, F. S. Khan and A. W. Muzaffar, "Image Colorization: A Survey and Dataset," *CoRR,* vol. abs/2008.10774, 2020.

[2] Z. Cheng, Q. Yang and B. Sheng, "Deep Colorization," *CoRR,* vol. abs/1605.00075, 2016.

[3] G. Petschnigg, M. Agrawala, H. Hoppe, R. Szeliski, M. Cohen and K. Toyama, "Digital photography with flash and no-flash image pairs," *ToG,* 2004.

[4] A. Deshpande, J. Rock and D. Forsyth, "Learning large-scale automatic image colorization," pp. 567–575, 2015.

[5] K. Nazeri, E. NG and M. Ebrahimi, "Image Colorization using Generative Adversarial Networks. Articulated Motion and Deformable Object," pp. 85-94, 2018.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial nets. In Advances in Neural Information Processing Systems," vol. 27, 2014.

[7] T. Welsh, M. Ashikhmin and K. Mueller, "Transferring color to greyscale images," *In ACM TOG,* vol. 21, pp. 277-280, 2002.

[8] A. Levin, D. Lischinski and Y. Weiss, "Colorization using optimization. In ACM transactions on graphics (tog)," *ACM,* vol. 23, pp. 689-694, 2004.

[9] P. Isola, J.-Y. Zhu, T. Zhou and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2016.

[10] A. Radford, L. Metz and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015.

[11] R. Zhang, P. Isola and A. A. Efros, "Colorful Image Colorization," *CoRR,* vol. abs/1603.08511, 2016.

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla and M. Bernstein, "Imagenet large scale visual recognition challenge. International Journal of Computer Vision," vol. 115, p. 211–252, 2015.

[13] G. Larsson, M. Maire and G. Shakhnarovich, "Learning representations for automatic colorization. European Conference on Computer Vision," 2016.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," vol. arXiv:1409.1556, 2014.