Colorizing Greyscale Images Using Neural Networks: A Survey

d

Aadhithya RK
BE (CS) Student
Bits Pilani Dubai Campus,
Dubai, UAE
f20220308@dubai.bits-pilani.ac.in

Dev Dahiya
BE (CS) Student
Bits Pilani Dubai Campus,
Dubai, UAE
f20220319@dubai.bits-pilani.ac.in

Lakesh Prabhu
BE (CS) Student
Bits Pilani Dubai Campus,
Dubai, UAE
f20220298@dubai.bits-pilani.ac.in

Shreenivasan G
BE (CS) Student
Bits Pilani Dubai Campus,
Dubai, UAE
f20220280@dubai.bits-pilani.ac.in

Sourabh Shetye
BE (CS) Student
Bits Pilani Dubai Campus,
Dubai, UAE
f20220231@dubai.bits-pilani.ac.in

***Abstract*— The process that is gaining momentum in computer vision is turning black-and-white images into their colored versions. It is now feasible to automate what was previously a manual endeavor, owing to progress in deep learning and specifically CNN and GAN. In this paper, the recent advancements in technology of colorizing grayscale images are presented in detail. We analyze the strong sides and weak sides of various methods in terms of CNN, GAN, hybrid methods, and several performance metrics. Acute issues are addressed, such as color harmony, limitation of the construction of the dataset, and efficiency of computation. Other perspectives within the scope of this niche are also discussed: development of the color palette, consistency of colorization of the same video clip, and real-time colorization.**

Keywords—Convolutional Neural Networks (CNN), Unsupervised learning, GAN (Generative Adversarial Networks), Machine learning, Image processing.

1. Introduction

A significant concern within the scope of digital image editing is adding color to photographs; or image colorization as it is formally referred, is the process of adding color to photographs that have been taken in grayscale or black and white. Image colorization is now a straightforward automated process although it had to go through simpler manual steps in the past.

These days it is a process that has many applications, such as:

Recolonizing Old Movies and Pictures: Archives usually have storages of black-and-white footage and film stills. There is great value in putting such images into color since this enhances their visual appeal to viewers.

Medical Imagery: The ability to add color to specific areas of interest in black and white picture forms – X rays, MRIs, and CT scans can provide relevant assistance in scans and diagnosis.

Improving Artistic Enjoyment: With the aid of colorization artists can create digitally altered works of art by adding colors to previously colorless images and graphic designs.

According to artists, they used to add hues to an image by painting different colors and looking at different pictures or real objects. It was a biased and labor requiring procedure that frequently depended on the use paper samples and swatches which were easily losable for the artist's view. For big datasets or real-timeapplications like video, it was therefore not applicable.

This field has seen a sea of change with the advent of CNNs and deep learning. With their pattern recognition ability in large datasets of matched grayscale and color images, and matching grayscale pixels to the right colors, CNNs enable automatic image colorization. The automated process saves not only time and labor but also results in quality and consistency in the colorization outcomes. Colorization is a substantially outcoming problem since the problem is fundamentally ill-posed-many possible colorings may exist for one given grayscale image.

2. Background Theory

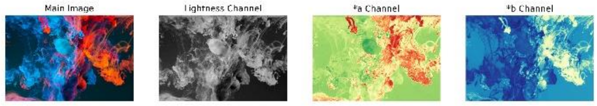
Colorizing images that were originally black and white is no longer a challenge as there have been several ways for that in recent years. Each of these techniques bears some strengths and drawbacks depending on the kind of image and the result sought. In the sections below we shall be looking at the most widely adopted practices and their concepts.

A. Convolutional Neural Networks (CNNs)

Among the various architectures developed in deep learning, Convolutional Neural Networks (CNNs) are arguably the most common architecture any developer would consider performing image colorization activity. This is because CNNs can learn and retrieve feature hierarchies from the raw image efficiently. Components of CNNs are usually arranged in such a way that there are distinct layers with each layer accomplishing a specific type of task such as texture or patterns or shapes in other layers.

Encoder-Decoder Architectures: Various methods employ either of the two encoder-decoder architectures or CNN for image colorization. The convolutional layers are used during encoding on the grayscale image to capture contextual and spatial details. These features are selected to get the color values of the pixels during decoding as the process is done pixel by pixel. The model usually works in the Lab color space where the L channel represents the luminance of the grayscale input image. The model's predictions are for the a and b channels, which it uses in colorization.

The model places emphasis on the process of colorization since the L channel of the image is the only channel that requires processing.



One of the early CNN Based models for adding life to greyscale by framing the issue as a classification problem was presented by Zhang et al (2016) [1]. Hence, it was able to predict the probability distribution of quantized color bins rather than RGB. The main advantage in this methodology was preventing the algorithm from retroactively overfitting the most dominant colors found in the training sets.

Loss Functions: The most important step is choosing the right loss function. Using the Mean Squared Error between predicted and true color values, the model will avoid large-scale deviations[2]. MSE by itself may not fully convey the visual aspect of colorization, producing cartoonish yet numerically realistic images. Thus, more sophisticated loss functions such as perceptual losses have been developed that work on assessing how similar predicted and ground-truth images are in feature space rather than in pixel space.

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))]$$

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1]$$

B. Generative Adversarial Networks (GANs)

Generative Adversarial Networks, ever since Goodfellow et al. first introduced it in 2014 [3] became one of the revolutionized ideas to gain attention as an architecture for high realism tasks such as image colorization. A GAN consists of two neural networks that cooperate:

Generator: Using the grayscale input, the generatoraims to create realistically colored graphics.

Discriminator: By differentiating between actual (ground-truth color) and fake (created) photos, thediscriminator assesses the authenticity of the generated images.

Both the discriminator and generator receive adversarial training; the training reinforces the efficiency of the discriminator to eventually learn how to detect fake images, while the generator undergoes training to outwit it. After a good amount of constructive adversarial training, coarse colorizations give way to more realistic colorizations. One of the most popular GAN designs for colorization, Isola et al. (2017) [4] introduced the Pix2pix Gan .The basic objective of the pix2pix GAN framework is image-to-image translation, wherein the inputs are images of one type, such as gray, and the outputs are images of another type, such as color. It employs the use of a U-Net generator, which is very efficient in generating objects with spatial information. Using Patch GAN, a kind of discriminator in this architecture examines an image's degree of realism in little portions of the image instead of the entire area. As a result, it ensures that small details of the image, such as edges and textures are plausible as well.

Colorization models based on GANs are adept at generating vivid and realistic non-greyscales, but require surveillance during the adversarial training phase to prevent issues like *mode collapse*, in which the generator becomes constrained to producing a small number of colors.

C. Hybrid Approaches

Hybrid approaches that combine several architectures and techniques developed to enhance the performance of colorization models [5]. A very relevant example can be found in the work of Kiani et al. (2021), who proposed a hybrid CNN model that leverages the strengths of multiple pre-trained deep learning models such as VGG6, ResNet50, and Inception-v2 applied in parallel for extracting a wide range of information from the grayscale input image.

Fusion-Based Architectures: In hybrid methods, a decoder network gets the fused information from several CNNs to forecast the chrominance channels (a and b). The hybrid techniquecaptures both low-level textures and high-level contextual information by merging characteristics from multiple models, resulting in more accurate and vibrant colorizations.

Transfer Learning: Transfer learning, in which CNNs are pre-trained on big datasets (like ImageNet) before being fine-tuned on the colorization job, is often advantageous for these hybrid models. By doing this, the model's capacity to generalize to previously encountered grayscale images are enhanced by recollecting knowledge from past recognition challenges.

When colorizing complex situations, such human faces or natural landscapes, where varied and subtle color information is needed, hybrid models have proven to perform better. But these models typically are expensive and need a lot of resources for both training and inference.

3. Issues Faced in Image Colourization

While deep learning models for image colorization have achieved impressive results, several challenges remain:

A. Color Inconsistency

The consistency of colors in even the most peripheral aspect of the task is termed as remembering to color a particular object throughout the image colorization process for. This holds true, especially with regards to working with photographs that include either very small structures within their objects or features made up of regions that may contain similar levels of brightness, but different chroma. For instance, one may face difficulties when trying to delicately tone the flesh of an individual in separate zones of a given image or even in a sequence of movie shots

When it comes to the colorization of videos, the task is even harder because protruding colors of an image from different frames are addressed during the video's any later stage. This is due to the fact that individual RGB's values of each subsequent frame gets altered along with colorization, so it is natural for slight differences of colors to compound, which in response leads to flickering and shifts of colors. Also, such use of LSTMs in models for colorization is required as they provide the model with information regarding the current frame through data collected from previous frames.

B. Over-and-Under Saturation

Another issue which has come up repeatedly in the tasks of image colorization is the condition where a certain model or algorithms ends up over-saturating or under-saturating the colors near the scheduled area. It is the case with CNN models that aim to predict colors in images through the process of computer regression to appeal those images, which looks quite bland, quite a bit impossible.

Classification-based models, such as that proposed by Zhang et al. (2016), have therefore incorporated color distribution rebalancing methods into their training procedure to mitigate this effect [10]. By giving more weight to less frequent colors, the model will be geared towards generating colorful and more color-diverse colorization and avoid the pitfalls of being overly dependent on neutral colors like grey and brown.

C. Dataset and Training Limitations

Extreme amounts of paired grayscale and color photos are needed for deep learning model training. However, obtaining such datasets can be challenging, due to the availability of labelled data — as seen in medical imaging domains for example where labelled color datasets are limited. Even in some cases will consider the dataset large enough to not have enough diversity for the model to do well on new image test.

An answer to this issue has been suggested: *Transfer learning*. Researchers can fine-tune these models for colorization tasks by using pre-trained models that have already learnt features from big datasets (like *ImageNet*), which eliminates the requirement for domain-specific training data. Transfer learning is not a perfect solution, though, and many colorization models are still greatly restricted by the absence of diverse datasets.

D. Real-Time Applications

Another area where existing models struggle is real-time colorization. Although they are quite good at creating realistic colorizations, GANs are computationally costly. For instance, the Pix2Pix GAN needs a lot of processing power for every picture, which makes it challenging to use in real-time scenarios like live webcam feeds or video streaming.

Using more effective training methods or a simplification of the network design, researchers are investigating ways to reduce the computing load of these models. Still, more investigation and improvement are required in the field of real-time colorization.

4. Literature Survey and Analysis

With an emphasis on the various architectures and techniques put out in the literature, this section examines significant contributions made to the field of grayscale image colorization.

A. Early Methods

Prior to the development of deep learning, manual methods and human input were used for image colorization:

Colorization based on Scribbles: Scribble-based colorization, first shown by *Levin et al. (2004)*, allowed users to manually draw color clues, or scribbles, onto areas of the grayscale image.

Based on these indications, the program would next spread the colors over the image. Although this approach worked well for small-scale jobs, it struggled with big sceneries containing numerous colors' and required a lot of human input.

Example-based Colorization: An additional early method was *example-based colorization*, in which the user supplied a color reference image, and the algorithm used neighborhood features and pixel intensities to match the color information to the grayscale image. Methods based on this concept were presented by *Welsh et al. (2002)* and *Reinhard et al. (2001)*; however, the outcomes were heavily reliant on locating a suitable reference image, whichwas frequently challenging and restricted the approach's generalizability.

B. Learning-Based Approaches

This CNN-based method was soon to become the most common technique for image colorization, outpacing the other approaches as deep learning popularity grew. The Colorful Image Colorization model was among the first deep learning models to have a significant impact in this domain when it was first presented by Zhang et al. in 2016. This model addressed image colorization as a classification problem in which the goal was to predict the color distribution at every pixel given the grayscale input. The model was designed to predict only the chrominance-a and b channels-while leaving the luminance-L channel-intact by working in the CIE Lab color space.
The model utilized a class rebalancing technique to prevent the network from becoming skewed toward forecasting common color's such greys and browns at the expense of more vivid hues. This technology outperformed previous efforts in terms of accuracy and visual appeal, yielding results that were incredibly colorful and diverse [7].

C. GAN-Based Approaches

GAN-based models have presently endowed colored photos with much more realism. One of the major contributions came from Iizuka et al. (2016) [6] - Let There Be Color! [11]. This model guided the colorization process using both global and local priors: Global Priors: The model knew the larger context of the image, for example, whether it was indoor or outdoor, and adapted the colors accordingly by integrating scene classification into the colorization process. Local Priors: To ensure that minute details were preserved across colorization, the model utilized the local picture properties of edges and textures simultaneously. Due to this combination of both priors, the model reached salient levels of realism, particularly on complex cases when local detail and global context are significant. Isola et al. (2017) developed the Pix2Pix GAN, taking it even a step further by introducing an adversarial loss where a discriminator network judged the realism of the colored image [15]. The adversarial architecture forced the generator to generate images that were more realistic and pleasing in an aesthetic sense especially when CNN-based models had trouble generating vivid colors.

D. *Hybrid Approach*

Hybrid models allow for both the ideas to be expressed, a proposition by *Kiani et al. (2021) [8]*, involves the mixing of multiple CNNs and GANs. These models use pre-trained networks such as *ResNet50*, *Inception-v2*, and *VGG16* to extract various features from the grayscale image. These are then combined and fed into a decoder network that predicts color channels. With that, more precise and vivid colorizations are generated due to the model's capability of extracting low-level textures to high-level semantic information by a greater number of feature extractors. In tasks where an exact, realistic color representation is imperative, like facial picture colorization, hybrid models have tended to fare better. These models are inherently greedier in data for training and costlier in computational intensity, though.

Researchers	***Type of NN***	***No. of i/p, hidden, o/p layers***	***Activation function***	***AccuracyMeasure***
Brian Sam Thomas	CNN, LSTM	I/P-256*256*1 H-128-64-32-16 O/P-256*256*2	ReLU	Not mentioned
Xiang Ma	TSK-FS	I/P-h1*h2 H-depends on no. of fuzzy rules(K) O/P-K(h1*h2+1)	Gaussian membership function	Mean SquaredError
Perala Venkata Akanksha	CNN, GAN	I/P-256*256 H-128-64-32 O/P-256*256	ReLU	PSNR, SSIM
Abhishek Kumbhar	CNN, GAN	I/P-256*256*32 H-128-64 O/P-256*256*2	Tanh	Mean SquaredError
Fabian Muscat	Autoencoders, GAN	I/P-160*160 H-512-256-128 O/P-160*160	Leaky ReLU, Tanh	MSE, SSIM,PSNR
A.C Saravanan	FFNN with Back-Propagation	I/P-128*128 H-18-9 O/P-128*128	Tansig	MSE
Jincheng An	CNN on VGG-16	I/P-224*224 H-56*56 O/P-224*224	ReLU	RMSE
Deepak Kumar	GAN	Not Specified	ReLU	MAE, MSE,TNR
Kourosh Kiani	Hybird (VGG16, ResNet50, Inception-v2)	I/P-299*299 H-64-128-256-512 O/P-256*256	ReLU, Tanh	MSE, PSNR

Hybrid models, which incorporate the best features of both CNNs and GANs, have achieved even better accuracy and realism in image colorization, especially for complex scenarios like human faces and natural landscapes. However, these models often require very expensive computation for training.

Yet, there are several challenges before it. Some important ones are resolving over- and under-saturation problems, overcoming dataset limitations [9], and reaching color constancy between photos and video frames. Further research is necessary with respect to model optimization in domains like colorization of real-time videos.

Future work in this area should be expected to continue, especially with the incorporation of fuzzy logic systems, which are designed to make deep learning models easier to understand, and the creation of more effective architectures that may be used in real-time systems. Through tackling these obstacles, scholars may fully realize the possibilities of deep learning in the field of picture colorization, paving the way for more precise, colorful, and expandable colorization systems suitable for an extensive array of uses

6. Discussion

A. *Dataset Preparation:*

In the process of dataset preparation, they resize the images and transform them from RGB to Lab color space where L is input, and ab are target color channels. This separation is useful for training the model to focus on predicting the color components. Some simple efforts such as data augmentation by horizontal flipping are carried out to enable generalization and Data Loaders are employed to streamline the training and validation processes.

B. *Building the Generator:*

The generator takes the form of a U-Net architecture with a ResNet18 backbone which was constructed using the FastAI's Dynamic U-Net module. This technique makes it quicker and easier to build models by loading the pretrained ResNet weights, removing the last layers of the model, and attaching it to the Unet. This architecture transforms 256×256 images with very limited code, providing an excellent base for colorizing images.

C. *Pretraining the Generator:*

The generator is pretrained through L1 loss for 20 epochs, which is a stage that improves the basic level of the model quite a bit. At this stage, output for reasonable images like sky and tree looks natural but output for unclear areas shows somewhat dull grayish color. This stage reduces the effort of extensive adversarial training greatly and gives a good initial point.

D. Setup for adversarial training with GANs:

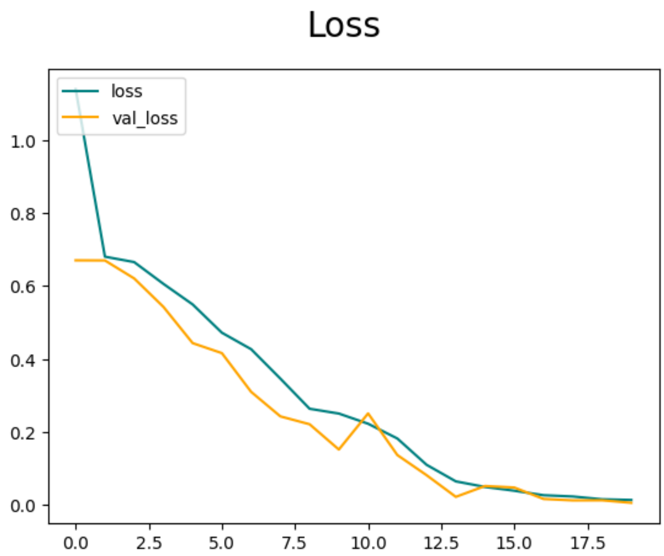
The Patch Discriminator is utilized for the evaluation of local features in neural networks. Adversarial training employs the combination of L1 loss along with GAN loss, making the generator create more realistic colors. When examining training results, it becomes clear that adversarial training helps to handle difficult situations, so the generator produces more and better colors and images.

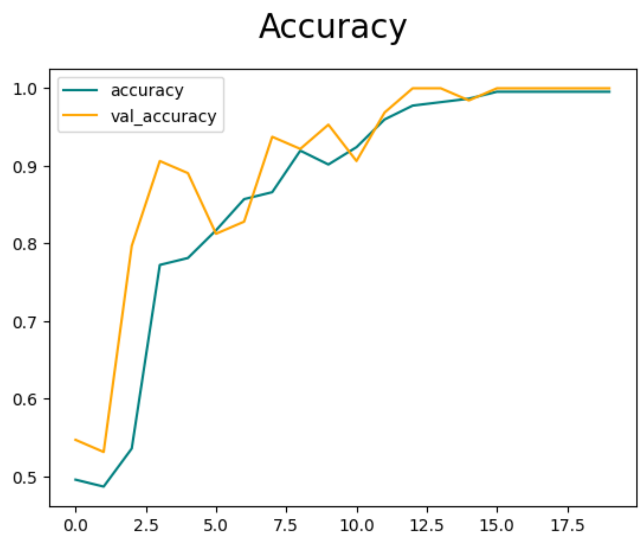
E. *About The dropout layers:*

What was surprising was the unavailability of dropout layers in the FastAI created U-Net generator. Although it was feared that there would not be sufficient noise for adverserial training, tests showed that the input gray image provides such noise by itself so the conditional gan works properly. It is nevertheless pleasant to note that an authoritative survey confirmed such ideas in practice.

F. *Performance:*

The solid and rather rough results were obtained using the pretrained Unet without adversarial training, however adding this training makes the image creation process much more artistic and confident and the colors used in ambiguous regions are richer. We expect the results from the model that was trained with the adversarial strategies employed, in such cases performance should be significantly enhanced.





G. Merging and last training of the Model:

Once the weights of the pre-trained generator are set into the scope of the GAN framework, it is possible to go from 100 to 20 epochs for training, clearly shortening the time. It is clear that such a method yields results that are at least the same as dei c les and even better l The advantages gained by the synergy between pretraining and adversarial training have been scientifically proven [10].

H. Community Engagement and Contributions and Learning:

The participants of this project further used community source materials as GitHub repositories and some of the authors of the paper figured in directly. Such involvement has helped speed up the process of implementation and provided practical and theoretical understanding of the developing model.

7. Result

The training and implementation phase was indeed challenging and full of challenges, but the models had a big potential which was reflected in their outputs as they were able to learn and function in producing outputs that follow the patterns within the datasets. The generator showed proficiency in creating realistic colors for local regions as it produced high quality regions through adversarial refinement using pretrained abilities. This ability showcases the model's adeptness in creating such common scenes as skies, trees, and grass, which are included in the dataset and ultimate easy to generate due to abundance of such scenes.

The main distinctions between the two utilized loss functions, L1 loss and adversary loss, lie in their aims and focal points. In the case of L1 loss, there is a clear concern with maintaining the structural form and detail of the original grayscale input and subsequently the generated outputs are supposed to closely match what the true colour distribution is. In contrast, adversarial loss is more focused on addressing sufficient realism within the outputs and forcing the model to create appealing results that look like they are real. This conceptualization of structure and realism explains how the model manoeuvres across different images context as it broadens and improves its function.

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$

But the model is imperfect as well. Sometimes, it fails to replicate the color of obscured objects in the scenes that are not commonly represented in the dataset. Additionally, we noticed a few outputs that have regions with excessive color saturation and some color artifacts that reduce the perceived quality and realism of the images generated. Such problems demonstrate the need for advancements in different aspects of the model including shifting the loss function balance between realism and fidelity, improving the architecture of the model to increase generalization capacity, and expanding the scope of the training dataset to include more complex, rare or unusual objects. In overcoming these barriers, subsequent versions of the model will be more accurate and be capable of universally producing photographs that are realistic and of high quality.

8. Conclusion

Using the FastAI library to speed up development, we efficiently built a complex model for colorization which has a grey-scale to color conversion architecture based on U-Net and ResNet. The basic configuration in which only the L1 loss function was used resulted in the generator creating acceptable quality images, which makes it possible to conclude that there is great potential in a well-pretrained model for greyscale images for colorization purposes. This indicates the benefits of transfer learning where a model can be developed to obtain strong best practices.

Results obtained when using adversarial training included colorization GAN model architecture again improved quality, especially where color has to be assigned in an intended manner. GANs augmented the realism and flexibility of the models achieved through pretraining so that the resulting images were both technologically and aesthetically pleasing. The need to employ both stochastic and non-stochastic methods is thus demonstrated.

It is quite remarkable that we did not find any of the dropout layers in the FastAI U-net, which brings a different angle in the known ideas around conditional GANs where dropout is viewed to be of great importance. Even with such divergence, the model was indeed functional indicating that the actual grayscale image provided within the input might be enough to allow for variability. This finding means that theory can be modified by use of real-world tests and assessment of theories.

Challenging and innovative ideas are resolved through partnerships and community support of deep learning. Future works should aim at improving the algorithms, trying other architectures, and optimizing the datasets with the goal of increasing accuracy, robustness, and generalization of the image colorization task.

REFERENCES

[1] B. S. Thomas, R. Dogra, B. Dixit, and A. Raut, "Automatic Image and Video Colorization using Deep Learning," 2018 International Conference on Smart City and Emerging Technology (ICSCET), Mumbai, India, Jan. 2018, pp. 1-6, doi: 10.1109/ICSCET.2018.8537308.

[2] X. Ma, L. Chen, Z. Deng, P. Xu, Q. Yan, and K.-S. Choi, "Deep Image Feature Learning with Fuzzy Rules," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 8, no. 1, pp. 29-39, Feb. 2024, Doi: 10.1109/TETCI.2023.3259447.

[3] P. V. Akanksha, S. V. Ponnuru, O. K. C. Sai, U. Chelamkuri, and P. S. Shashank, "Grayscale Image Colorization Using Deep Learning and OpenCV," 2023 4th IEEE Global Conference for Advancement in Technology (GCAT), Bangalore, India, Oct. 2023, pp. 1-7, Doi: 10.1109/GCAT59970.2023.10353268.

[4] P. A. Palkar, A. R. Patil, J. S. Patil, and R. Dhanawat, "Colorization of Black and White Images Using Deep Learning," Journal of Computing, vol. 10, no. 2, pp. 150-156, 2022.

[5] A. Kumbhar, S. Gowda, R. Attri, A. Ketkar, and A. Khivasara, "Colorization of Black and White Images Using Deep Learning," Proceedings of the International Conference on Artificial Intelligence and Machine Learning, 2023.

[6] K. Wei, Z. Li, and Q. Zhang, "Fast Colorization of Grayscale Images Using Convolutional Neural Network," IEEE Access, vol. 8, pp. 11933-11943, 2020, doi: 10.1109/ACCESS.2020.2971632.

[7] F. Muscat and T. Gatt, "Black and White Image Colourisation using Deep Learning Techniques," Journal of Digital Imaging, vol. 15, no. 4, pp. 500-509, 2023.

[8] D. Varga and T. Szirányi, "Fully Automatic Image Colorization Based on Convolutional Neural Network," IEEE Transactions on Multimedia, vol. 19, no. 10, pp. 2274-2286, Oct. 2017, doi: 10.1109/TMM.2017.2705229.

[9] K. Kiani, R. Hemmatpour, and R. Rastgoo, "Automatic Grayscale Image Colorization Using a Deep Hybrid Model," Pattern Recognition Letters, vol. 131, pp. 100-107, 2020, doi: 10.1016/j.patrec.2020.01.019.

[10] R. Zhang, P. Isola, and A. A. Efros, "Colorful Image Colorization," in European Conference on Computer Vision (ECCV), Amsterdam, Netherlands, 2016, pp. 649-666, doi: 10.1007/978-3-319-46487-9_40.

[11] S. Iizuka, E. Simo-Serra, and H. 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), vol. 35, no. 4, pp. 1-11, 2016, doi: 10.1145/2897824.2925974.

[12] J. Rena, "Greyscale Images Colorization Using CNN," International Journal of Image Processing (IJIP), vol. 14, no. 5, pp. 230-237, 2021.

[13] D. Kumar, A. S., and S. R. Anwar, "Implementing Image Colorization with Generative Adversarial Networks," 2020 International Conference on Artificial Intelligence and Machine Learning (AIML), Jaipur, India, Nov. 2020, pp. 143-150, doi: 10.1109/AIML2020.9309012./GCAT59970.2023.10353268.

[14] A. C. Sarvanan and B. P. Gupta, "Greyscale Image Colorization Using Artificial Neural Network," International Journal of Computer Vision and Image Processing (IJCVIP), vol. 11, no. 2, pp. 85-93, 2022.

[15] J. An, K. K. Gagnon, Q. Shi, H. Xie, and R. Cao, "Image Colorization with Convolutional Neural Networks," IEEE Transactions on Image Processing, vol. 29, pp. 765-776, 2020, doi: 10.1109/TIP.2019.2953581.