iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** IX **Month of publication:** September 2021

DOI: https://doi.org/10.22214/ijraset.2021.38218

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IX Sep 2021- Available at www.ijraset.com

Image Colorization Using Autoencoders

Rachaell Nihalaani[1], Simran Mansharamani[2], Juhi Janjua[3]
[1] Undergraduate Student, [2]Graduate Student, [3]Assistant Professor, Department of Computer Engineering, Thadomal Shahani Engineering, College, Mumbai-50, Maharashtra, India

Abstract: Modification of art may be viewed as enhancement or vandalization. Even though for a long time many were opposed to the idea of colorizing images, they now have finally viewed it for what it is - an enhancement of the art form. Grayscale image colorization has since been a long-standing artistic division. It has been used to revive or modify images taken prior to the invention of colour photography. This paper explores one method to reinvigorate grayscale images by colorizing them. We propose the use of deep learning, specifically the use of convolution neural networks. The obtained results show the ability of our model to realistically colorize grayscale images.
Keywords: Deep Learning, Convolutional Neural Network, Image Colorization, Autoencoders.

I. INTRODUCTION

The process of taking a grayscale (black and white) image as input and giving out a colorized image as the output which represents the semantic colors and tones of the input is called Image Colorization [1]. It basically assigns color to black and white images making it enhanced and more appealing. Previous methods for image colorization either were based on annotation and significant human interaction or produced desaturated colorization. It is a highly unsettled problem which requires a real-valued luminance image to be mapped to a three-dimensional color-valued one, that has no unique solution [2]. We have proposed the solution using Autoencoders which achieves optimized results. Autoencoders are a type of feedforward neural networks with the same input and output. In brief, the input is compressed into a lower-dimensional code representation and is reconstructed to give the output [3]. The rest of the paper is assembled as follows. Section 2 reviews related literature. Section 3 provides a description of the dataset used in this paper. The methodology used is explained in Section 4. Section 5 analyzes the results obtained while Section 6 draws the conclusion to the paper as well as describes future scope.

II. LITERATURE REVIEW

There are various algorithms that can be used for image colorization.
To fully generalize the colorization procedure, [4] used conditional Deep Convolutional Generative Adversarial Network (DCGAN) and compared the results between the generative model and traditional deep neural networks. They came to the conclusion that the CIFAR-10 dataset under GAN had the highest accuracy among other networks. The convolutional neural network based model was used by [5] and they combined it with Inception ResNet V2 classifier to colorize images. They found that the colorized image was not as accurate with the ground truth with respect to color due to their poor choice in loss function. [6] shows that the combination of a deep CNN and a carefully selected objective function used for colorization can produce results that are difficult to distinguish from real color photos. They achieved this by training the CNN with various losses. An effective color-CycleGAN solution was proposed by [7] with their objective being to generate reasonable color instead of reinstating the original color and they successfully achieved a performance better than several stat-of-the-art methods. [2] uses the ChromaGAN model which is based on an adversarial strategy that captures perceptual, semantic and geometric information. Their experimental results show that their adversarial technique provides realistic color images.

III.DATASET

The dataset used in this paper is the Landscape Color Image [8] dataset obtained from Kaggle. This dataset is appropriate for our framework as it was created with the main objective of creating an autoencoder network that can colorize grayscale images. This dataset contains images of landscapes such as streets, buildings, mountains, glaciers, trees etc. It also has the corresponding grayscale images in another folder. The dataset consists of upwards of 7000 grayscale images and another set of their corresponding colored images. In total, we have upwards of 14000 images. Now, the reason there are both, the grayscale and color versions of a single image, is to compare the color output of our autoencoders with the color version of the image, for similarity.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IX Sep 2021- Available at www.ijraset.com

IV. METHODOLOGY

A. Autoencoder

Autoencoder is a special deep learning architecture. To put into perspective, autoencoders are a class of convolutional neural networks (CNN). They are a kind of neural network which learns representation of raw data in compressed form. It consists of two sub-models or networks - the encoder and decoder. First, the encoder learns input data as a reduced dimensional representation through a series of CNN and down-sampling. Then the decoder attempts to regenerate the data from these representations through the use of CNN and up-sampling. A well-performing decoder is one that is able to regenerate identical data as the original input data. After the model is trained, decoder is discarded but encoder is saved for later use [9].

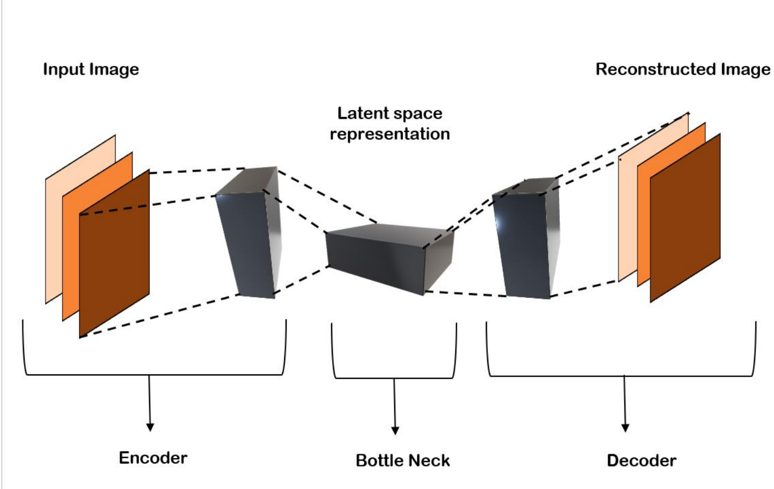


Fig. 1 Autoencoder Architecture

B. Implemented Methodology

In our model, we have used python and have implemented the image colorization task using the following steps. Once all required python libraries such as tensorflow, keras, numpy etc. have been imported, we begin the exploratory analysis of the data. Details of the dataset used [8] have been provided in the previous section. Next step would be data pre-processing and visualisation. As this Landscape color image dataset [8] has grayscale images as well as corresponding color images, we display them side by side in order to visualise our expected output. Pre-processing of data includes resizing or normalising images and converting them to an array. Post this process, data is ready to be used in our model.

Next, data is split into training data and testing data. Since our data is now in the form of an array, we 'slice' the array to achieve splitting. Next step is to define our Encoder model which uses CNN and downsampling. On the same lines, we define our decoder model which uses CNN and upsampling. Attempts such as concatenation of encoder and decoder layers, have been made to prevent feature loss. Next, the model is fit on our training data, followed by evaluating the model on the test data. The final step is to display the model output which consists of the grayscale image, the generated color image as well as the original image for comparison.

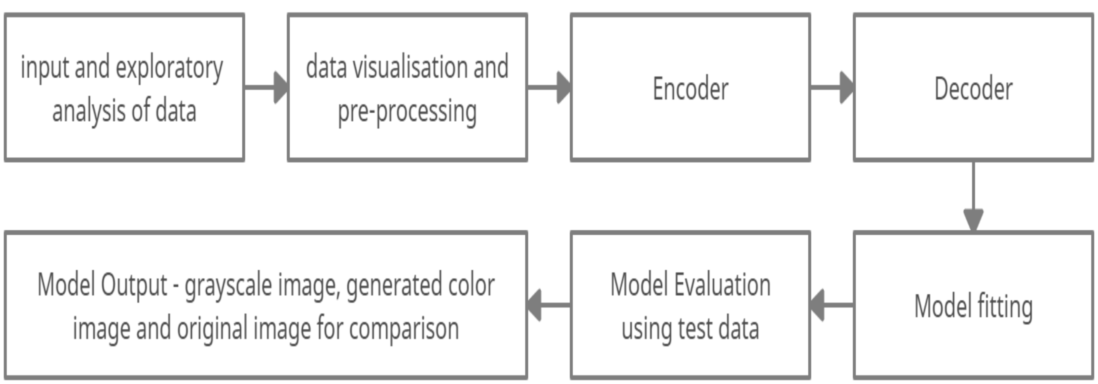
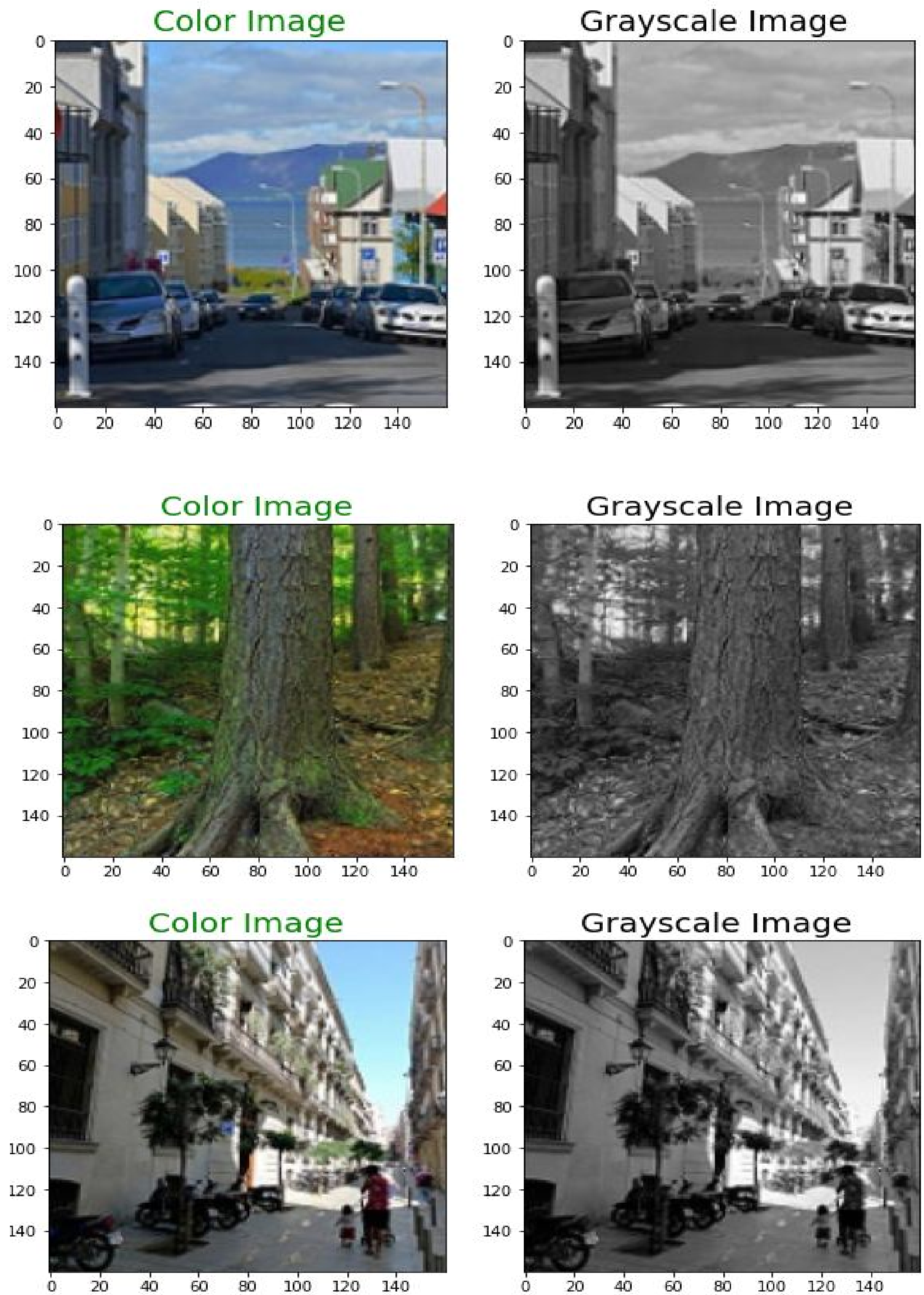


Fig 2. Proposed Framework

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IX Sep 2021- Available at www.ijraset.com

V. RESULT AND ANALYSIS

Before the results are shown, in order to get familiar with our dataset, we displayed a few grayscale and corresponding color images present in the dataset. This also helped us analyse the expected color scheme, vibrance and other factors that needed to be understood.



ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IX Sep 2021- Available at www.ijraset.com

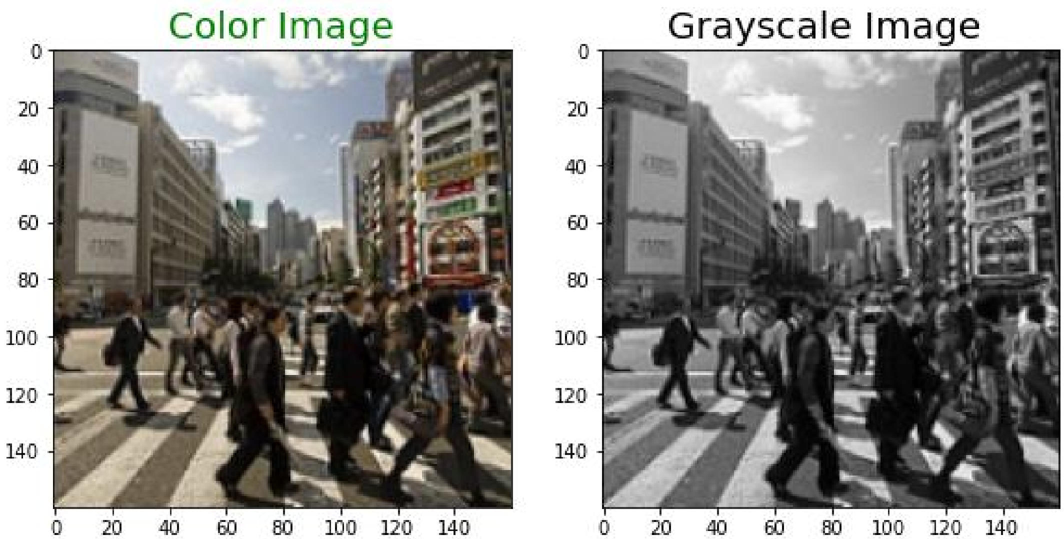
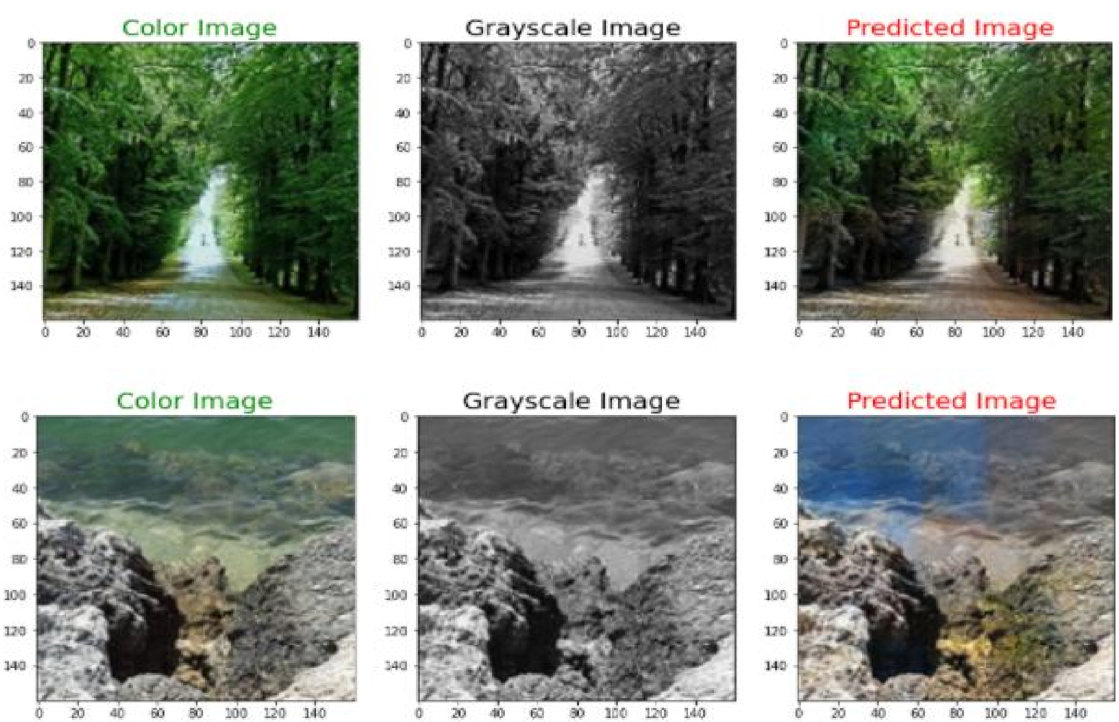


Fig 3. Few instances of Landscape Color Image Dataset

As seen in the image above, the model is able to accurately judge the colors in the image. However, they do have a patchy pigmentation or a blur kind effect.

In an autoencoder architecture, the size of the input information size is reduced first, accompanied by following layers. This constitutes the encoder part of the architecture. The decoder part is initiated where the model learns linear feature representation, and gradual increase in size. At finish, the input and output sizes are equal. While is architecture has the upside of preserving the output size, it has the downside of linearly compressing the input. This compression causes a bottleneck, due to which the transmission of all features becomes less possible. Also, default loss in encoder-decoder based image reconstruction had been l2 loss. This function is known to poorly correlate with image quality. L2 loss generally makes the image blurry since minimizing it means maximizing the log-likelihood of Gaussian.

But good colorization isn't how well the colour matches to the original image but how plausible it looks to us humans.[10]



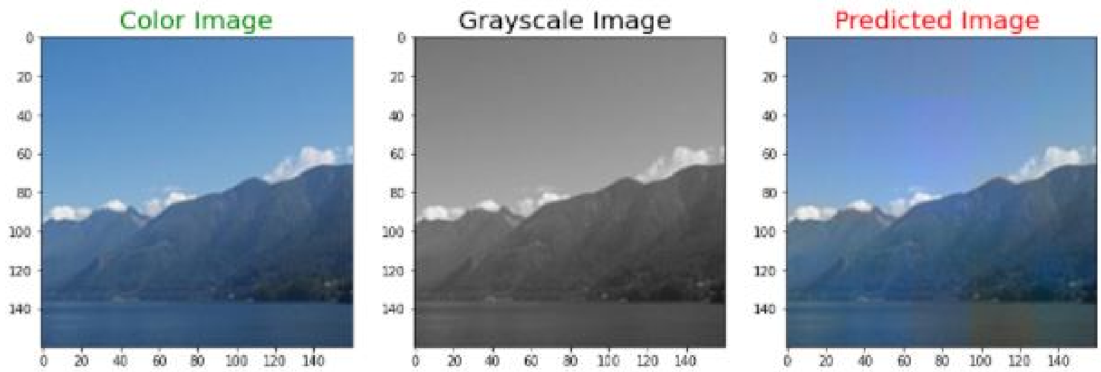


Fig 4. Model Output - grayscale image and predicted image along with original color image for comparison

VI. CONCLUSION AND FUTURE SCOPE

In this paper, a reliable method for colorizing landscape grayscale images has been presented using CNN and Autoencoders. These neural networks are able to distill the salient features of an image, and then regenerate the image based on learned features.It is now known that for colorization, a combination of deep CNN and an objective function that is well-chosen produce the finest results. Being trained to color only, it surprisingly learns a representation useful for object detection, classification and segmentation that perform strongly when compared to other self-supervised pre-training methods. We expect that the results will improve if the network is trained further.

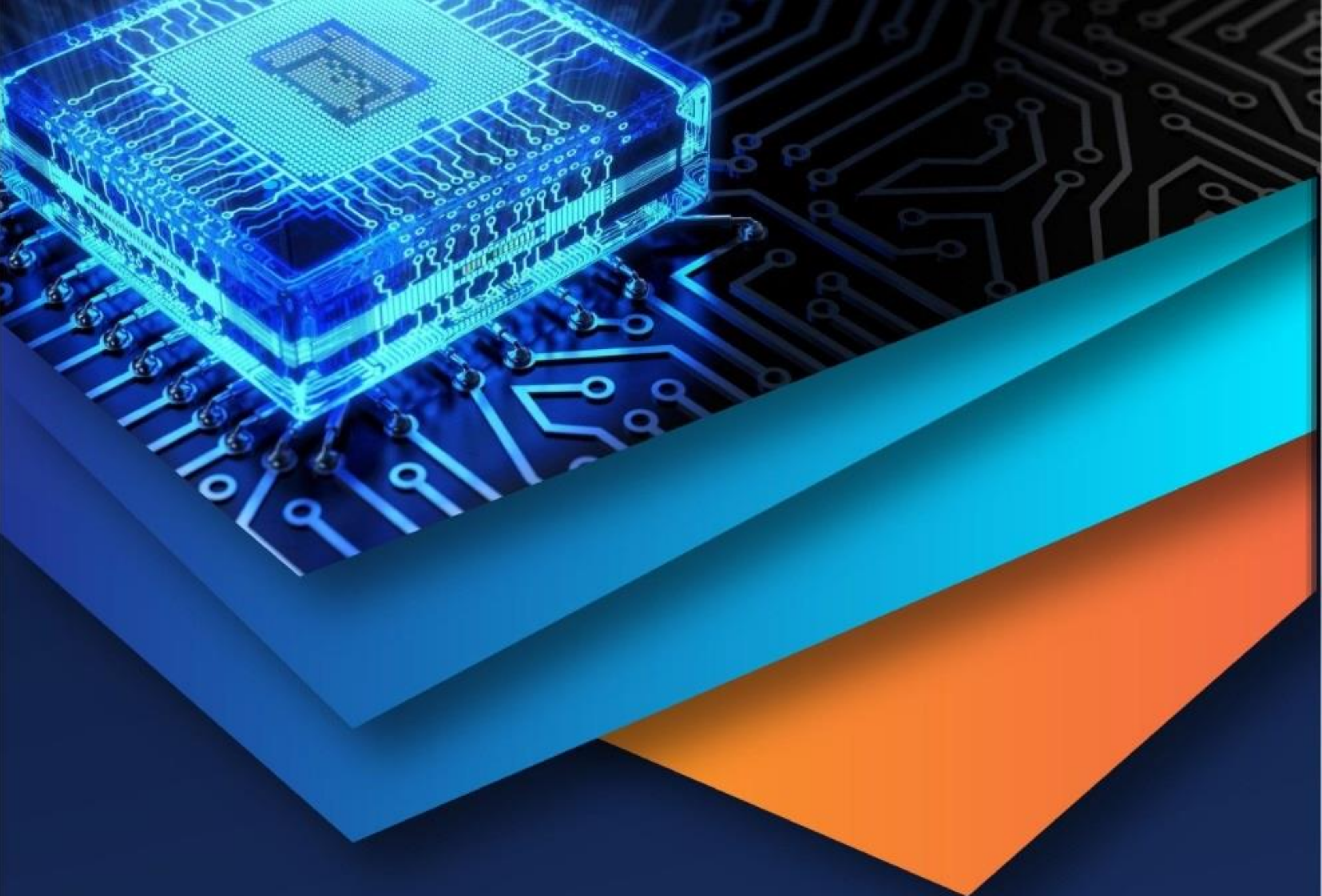
Building on our model, we could also use other methods such as conditional generative adversarial networks such as Pix2Pix or even deep convolutional generative adversarial networks (DCGANs) to compare and evaluate and choose the best model.

VII. ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our professor Juhi Janjua for her constant support and guidance. We would also like to extend our gratitude to our Head of Department, Dr. Tanuja Sarode, and our Principal, Dr. G.T. Thampi.

REFERENCES

[1] A. Rosebrock. "Black and white image colorization with OpenCV and Deep Learning." pyImageserach.com https://www.pyimagesearch.com/2019/02/25/black-and-white-image-colorization-with-opencv-and-deep-learning/ (accessed Sep. 15, 2021)
[2] P. Vitoria, L. Raad and C. Ballester. (Jan. 2020). ChromaGAN: Adversarial Picture Colorization with Semantic Class Distribution. [Online]. Available: https://arxiv.org/pdf/1907.09837.pdf
[3] A. Dertat. "Applied Deep Learning - Part 3: Autoencoders." towardsdatascience.com https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798 (accessed Sep. 15,2021)
[4] K. Nazeri, E. Ng and M. Ebrahimi. (Jun. 2018). Image Colorization Using Generative Adversarial Networks. Presented at AMDO. [Online]. Available: *https://link.springer.com/chapter/10.1007/978-3-319-94544-6_9*
[5] A. Avery and D. Amin. (2018). Image Colorization. [Online]. Available: https://cs230.stanford.edu/projects_spring_2018/reports/8290982.pdf
[6] R. Zhang, P. Isola and A. A. Afros. (Sep. 2016). Colorful Image Colorization. Presented at ECCV. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-46487-9_40
[7] Y. Xiao, A. Jaing, C. Liu and M, Wang. (Sep. 2019). Single Image Colorization via Modified CycleGan. Presented at ICIP. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8803677
[8] B. Mamba, "Landscape color and grayscale images." (December 15, 2020). Distributed by Kaggle. https://www.kaggle.com/theblackmamba31/landscape-image-colorization
[9] J. Brownlee. "Autoencoder Feature Extraction for Classification." machinerlearningmastery.com https://machinelearningmastery.com/autoencoder-for-classification/ (accessed Sep. 12, 2021)
[10] S. Nayak. "Convolutional Neural Network, CNN based Image Colorization using OpenCV." LearnOpenCV.com https://learnopencv.com/convolutional-neural-network-based-image-colorization-using-opencv/ (accessed Sep. 12, 2021)



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY