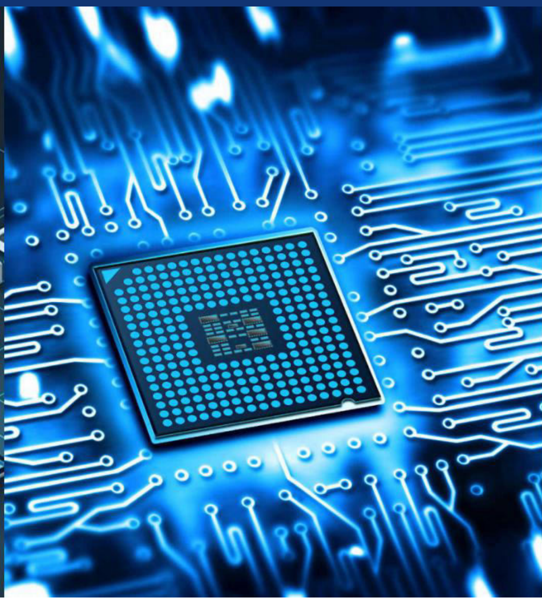
IJIRCCE

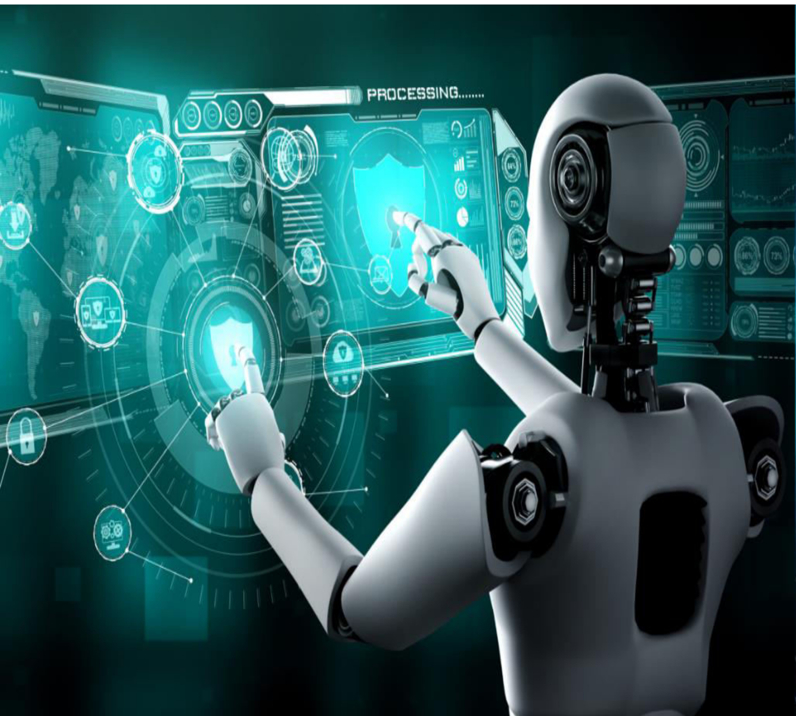
ISSN(O): 2320-9801

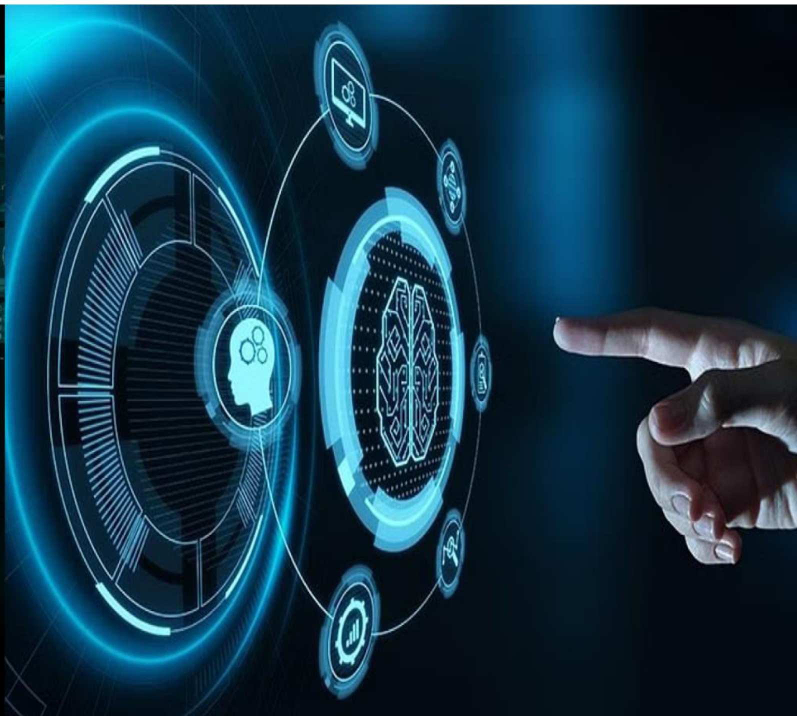
ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 8.771

Volume 13, Issue 5, May 2025

www.ijircce.com ijircce@gmail.com +91-9940572462 +91 63819 07438

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305017

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Noise Removal using Auto Encoders, Auto Decoders, and Grayscale Image Colorization using CNN

V. Yamuna[1], M. Hema Deepika[2], Ch. Charan Teja[3], D. Sumanth Kumar[4]

Asst. Professor, Department of ECE, N.B.K.R. Institute of Science and Technology, Vidyanagar, Tirupati District, Andhra Pradesh, India[1]

UG Students, Department of ECE, N.B.K.R. Institute of Science and Technology, Vidyanagar, Tirupati District, Andhra Pradesh, India[2,3,4,5]

ABSTRACT: This thesis investigates the application of Convolutional Neural Networks (CNNs) for the task of grayscale image colorization. Colorization refers to the process of adding color to grayscale images, a traditionally manual and time-consuming task. With the advancement of deep learning, particularly CNNs, it has become feasible to automate this process with considerable accuracy. This work presents a detailed exploration of the architecture, training procedures, datasets, loss functions, and evaluation metrics involved in developing an efficient CNN-based colorization model. Experimental results on benchmark datasets demonstrate the model's ability to generate realistic and visually pleasing colorized images.

KEYWORDS: Grayscale Image, Image colorization , CNN Algorithm, Machine Learning.

I. INTRODUCTION

For a long time, colorizing has been achieved by through direct painting; however, this requires significant time and cost. Nevertheless, people have continued to color greyscale images to capture the vividness of many old historical photographs. As more computational techniques have been developed, methods have been devised for automatic painting via computers. Image colorization techniques refer to techniques that add colors (RGB) to grayscale images or videos. When users see grayscale images, they can estimate the colors from the known information. However, it is impossible for humans to find the perfect color. Colorization is an ill-posed problem and does not offer a unique solution and thus focuses on finding plausible colors rather than accurate colors. Grayscale image coloring is an important task in many areas, such as the film industry, historical data review, and photography technology. In particular, as the mass production of webtoons increases around the world, the field of automatic cartoon coloring has drawn increased attention [25]. Significant progress has been made in this area but automatic image coloring remains a challenge. There are three methodologies in the colorization field, i.e., scribble-based, exemplar-based, and fully automatic. The first and second methods are classified as user-guided edit propagation methods and the third is a data-driven automatic colorization method. Deep learning significantly progressed over the past decade, especially in the field of image processing, where very good results are being obtained via convolutional neural networks CNNs). These techniques have recently been applied to colorizing methods and the results are shown to be very good. Prior to the emergence of deep learning techniques, the most effective methods relied on human intervention. In this paper, we propose a colorization technique using a fully automatic deep learning method. Most colorization methods utilize a structure called the encoder-decoder model; for this, we have modified FusionNet [18]. Owing to the poor performance of the model using the underlying MSE loss function, training was conducted by defining a novel loss function suitable for the coloring objective. We used the ImageNet dataset for training and validation and tested datasets of various images using the training model. The performance of the proposed model was quantitatively and qualitatively verified through various experiments.

II. EXISTING METHOD

The grayscale image coloring field is commonly divided into three categories: scribblebased, exemplar-based, and fully automatic methods. The first and second methods require human intervention and are called user-induced editing propagation methods. An end-to-end automatic method that minimizes human intervention is required. A fully

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

automatic method is introduced, which produces a color image that fits the grayscale image end-to-end, without human intervention for any image. In the scribble-based method, users provide colored scribbles, giving the model color-based hints. Scribble-based methods are interactive methods for coloring grayscale images by placing doodles. The input value that corresponds to the input image is the appropriate colored doodle drawing or point of color in the grayscale image. Levin et al. [15] introduced this method, which assumes that pixels adjacent to the space containing pixels with color information should have similar colors. In addition, other related methods have been proposed [9, 24, 16]. Zhang et al. [27] developed a model that randomly selects several pixels from a color image and assembles them as scribble input. Xiao et al. [23] developed the method of Zhang et al. [27] and created a model that used the overall color theme of an image as a global input. However, scribble-based methods are less efficient because they require a large amount of direct user input and selecting color schemes involves human subjectivity, allowing users to choose the wrong color. In the exemplar-based method, the grayscale and color images are used as inputs; the color image is an example image, with a color that is expected to appear as a result of a grayscale image. This image is called an exemplar or reference image.

Welsh et al. [22] developed a model that combines exemplar images and transforms colors by matching the statistics within the region adjacent to the input color image pixel. Kang et al. [13] proposed an improved method that uses an attention structure and webtoon images as a dataset. Various other methods have been devised [20, 5, 3]. These methods have significantly reduced user intervention compared to previous methods but are less efficient because they always require a reference image that is similar to the original color image of the grayscale image and require many images to be processed. Fully automatic methods are end-to-end deep learning-based coloring methods and are widely used today. In particular, there are several methods for creating models based on CNNs that receive only grayscale images as inputs. Therefore, it is called data-driven automatic colorization. However, as mentioned earlier, this method has no information (scribbles or exemplar images) regarding the desired colors; thus, it is likely to choose the wrong color. Cheng et al. [4] first created a fully automatic colorization method that uses a deep neural network. By training large amounts of data, people can plausible colors can be predicted for grayscale images.

These models are fundamentally based on the structure of encoder–decoder models because they need to extract the features of the grayscale image and reconstruct the color using this information; i.e., a similar structure to encoder–decoder models. Iizuka et al. [10] trained local and global inputs together to provide additional class information. Following Iizuka et al. [10], Baldassarre et al. [2] trained semantic information by providing classification classes. Zhao et al. [28] used semantic segmentation information to find the color. Zhang et al. [26] developed a very creative method for learning a particular label for all pixels while using a basic structure model. Several recent methods for coloring grayscale images using generative adversarial networks (GANs) [7] have emerged. Isola et al. [12] used a U-net-based generator with conditional GAN.

In addition, Nazeri et al. [17] used a similar method to that used by Isola et al. [12]. Recently, Vitoria et al. [21] discussed methods for using improved WGANs while providing class information, such as that by Iizuka et al. [10]. Variational autoencoders (VAE) [14] were used to learn color embedding in Deshpande et al. [6]. New models are emerging that use more effective inputs and differ from existing methods. One of these methods is influenced by natural language processing (NLP), whereby the user enters textual information regarding the color of the ground truth image [1]. Changqing et al. [29] created a new dataset [31] that directly contains text information and a model that allows users to enter text directly with sketch information. [30] developed the model further.

The goal of image colorization is to assign a suitable color value to each pixel of a target grayscale image such that it looks natural. Because image colorization technique can provide abundant information of goal-oriented scenario for human vision (Xiao et al. 2013, 2022; An et al. 2020), this technique has been widely used in many systems and applications, such as historical photographs and videos (Chen et al. 2018), artist assistance (Chen et al. 2022; Liu et al. 2022; Hensman and Aizawa 2017; Ren et al. 2018; Shinya et al. 2018), remote sensing (Ji et al. 2021; Song et al. 2018; Liu et al. 2018), night vision imaging systems (Xu et al. 2021; Liang et al. 2021; Suarez et al. 2018; Dong et al. 2018), restoration of aged or degraded images (Nazeri et al. 2018), and interior finish (Zhu et al. 2018). In the past few years, image colorization has attracted increasing attention from researchers and achieved great improvement (Viana et al. 2020; Abo-Hammour et al. 2014a, 2014b; Arqub and Abo-Hammour 2014). However, due to the diversity of object colors in the real world and the fuzziness of human perception for colors (Tang et al. 2021; Faridul et al. 2016; Cheng et al. 2015; Larsson et al. 2016a), it is still a challenging task with no perfect solution. While there exists some methods to

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305017
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

answer the image colorization problem. These methods can be broadly classified into four categories: scribble-based methods

III. PROPOSED METHOD

To explain the detailed method for colorizing grayscale images with the proposed model, given a grayscale image L, the proposed model produces aandb values, which are color channels in the CIE Lab color space. The proposed model receives the input image (grayscale image), passes through a feature extraction part (encoding part), a reconstruction part (decoding part), and finally creates two color channels. We selected FusionNet [18] as the encoder–decoder model, with several modifications. Furthermore, the new loss function was applied to infer the point estimates of the colors from the predicted color distribution

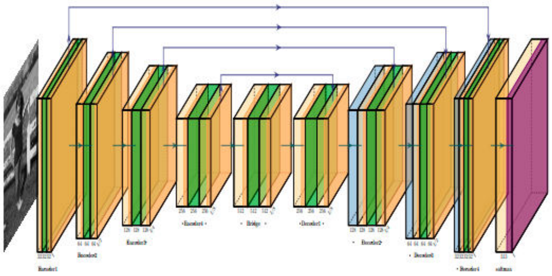


FIGURE 1. Proposed Architecture in this paper

We define a new loss function for the coloring task; i.e., for input grayscale data X ∈ R h×w×1 , the model has to find the a, b channels, such as Yˆ = G(X) ∈ R h×w×2 . The most basic loss function is the mean square error (MSE), which is an image-to-image regression approach. The MSE loss function equation can be written as follows:

$$L_{MSE} = \frac{1}{2}\sum_{h,w} \|Y - \hat{Y}\|_2^2.$$

The optimal solution of the MSE is the average set for the image. When applied, the resulting values (i.e., the values a and b) are almost intermediate and thus do not represent the vivid saturation found in nature. Therefore, we obtain a grayish and awkward color. Hence, we use a method to find a specific label for each output pixel and compare it with the original label for that pixel. This method was first proposed for colorful image colorization by Zhang et al. [26]. In terms of multinomial classification, a multinomial cross-entropy loss function can be used. The model must learn the mapping Zˆ = F(X), where Zˆ ∈ R h×w×n and n = 313. To compare the predicted Zˆ against the ground truth Z, the ground truth color Y is converted to Z.

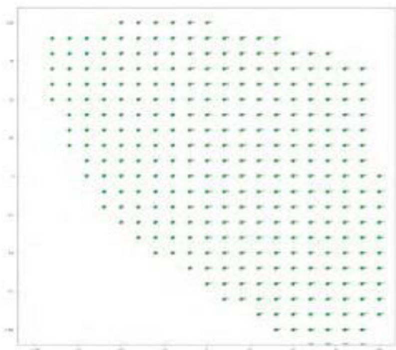


FIGURE 2. Quantized bins of 313 representative points in CIE Lab Color

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

To obtain this point, we quantize 313 representative points of channels a and b, as shown in Figure 2. For a particular pixel's a and b values, the nearest representative point becomes the label of that pixel. Using the k-nearest neighbor method, each pixel in the ground truth image Y can be encoded as a vector that contains the label of the classification problem. Instead of a one-hot vector, we found five quantized bins in the order close to each pixel; i.e., the 5-nearest neighbors are found and, using a Gaussian kernel with $\sigma = 5$, are weighted proportionally to their distance from the ground truth. When this method is applied to each pixel, a vector of $h \times w$ dimensions is generated. This allows the cross-entropy loss function to be applied between vectors from the output of the model and vectors from each pixel of the ground truth. By adding all the loss values of each pixel, we specify the complete loss function. Multinomial cross-entropy loss function, defined in Eq.(3.1)

$$L_{cl} = -\sum_{h,w}\sum_{n} Z_{h,w,n} \log(\hat{Z}_{h,w,n}).$$

As mentioned earlier, the proposed model was modified from FusionNet [18]. Unlike the existing FusionNet

model, the input/output size was set to 256 × 256. If the input/output size is too large, such as FusionNet, the training is too long to handle a large number of images. To resize images, we used bicubic linear interpolation to use the lowest-order interpolation method to connect pixel values at adjacent grid boundaries smoothly. The number of downscaling and upscaling of the proposed model was reduced by one compared to FusionNet. Thus, the model reduces the feature map size from 256 × 256 to 32 × 32; this is because the input size of our model is smaller than that of FusionNet and thus much information can be lost if the feature map size is further reduced. Furthermore, we used a 2-stride convolution layer to reduce the size of the feature map while max-pooling is used in FusionNet. ResBlock (Residual Block) placed between the encoding and decoding blocks is the same as that of FusionNet. It was first introduced by He et al. [8]. ResBlocks can take various forms. A detailed description of the block used in the proposed architecture is shown in Figure 3. It is designed to create a type of shortcut (skip connection) so that gradation can flow well, even if the layer is deep. There is a long skip connection that links information from previous feature maps to future feature maps. It provides the results of adding the first input feature map and the last feature map three times after convolution and batch normalization [11] layers. The filter size of all convolutional layers was 3 × 3.

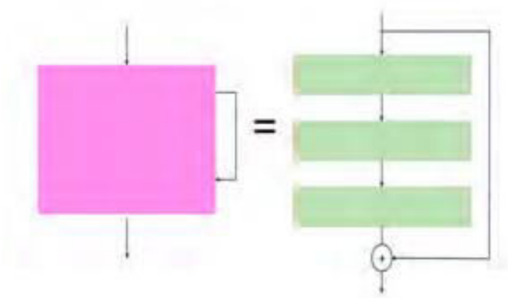


FIGURE 5.3. Detail description Residual Block from FusionNet [18]

The decoder is a reconstruction part and has a structure similar to that of the encoder. The decoder was composed of four decoding blocks. First, the size of the feature map is expanded through a deconvolutional block. After the long skip-connection, which concatenates with the feature map of the same size in the encoding block to create an expanded feature map, the resulting feature map passes sequentially through the convolutional + batch-normalization layer and ResBlock. Resblocks are used in the same structure as those used in the encoding blocks. The feature map enters convolutional + batch normalization once more. This process was repeated four times until the size of the feature map was equal to the size of the input image. However, for the last convolutional layer of these iterations, the output depth was set to 313. Similar to the encoder, all convolution layers use the Relu activation function but, in the last layer, the Softmax activation function is used.

The bridge follows similar formats to the encoder and decoder. However, the bridge simply links information of the

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

feature map from the encoder to the decoder and consists of only one block. FusionNet, based on the proposed models, also has an intermediate bridge, reducing the size of the feature map. However, the proposed model receives a feature map from the encoder and sends it to the decoder section at a constant size (the same resolution). After completing the encoder task, the feature map that enters the bridge goes through the same process as the encoding block to create a feature map to send the decoder.

IV. LITERATURE REVIEW

In this section, a brief overview of the existing colorization methods and their advantages or limitations are provided In general, image colorization techniques can be divided into four categories: scribble-based method, color transferbased (example-based) method, learning-based method, and hybrid method. 1) Scribble-based method For scribble-based image colorization, users should give some with specified colors, such as color lines or points. Then, the colors will be automatically propagated on the whole image according to a given similarity detection and diffusion method (He et al. 2018). In 2004, Levin et al. (2004) first proposed an image colorization method based on a simple premise that the neighboring pixels with similar intensities should have the similar colors; this work provided a fundamental assumption for scribble-based image colorization method. For

overcoming the limitation of color bleeding around object boundaries in Levin's method (Levin et al. 2004), Huang et al. (2005) proposed an image colorization method based on an adaptive edge detection scheme. Then, Luan et al. (2007) presented a colorization method by considering the similar intensity of neighboring pixels and the similar texture of remote pixels to propagate the specified colors by users. The schematic diagram of scribble-based image colorization method is shown in Fig. 1 (Shinya et al. 2018). The shortcomings of scribble-based methods are that the performance is rely on manual work and experience in providing good scribbles, thus it is difficult to colorize a complex image and easy to appear artifacts. Generally, the scribble-based methods are suitable for cartoon image colorization and not appropriate for the colorization of natural scene, because the content of cartoon image is simple compared with the natural scene, such as people photos and scenic photos. 2) Color transfer-based method The pioneering works of color transfer-based method for image colorization were realized by Reinhard et al. (2001) and Welsh et al. (2002) who provided color transfer techniques between two images. Generally, most color transferbased methods can be regarded as a global algorithm, thus spatial information may be ignored in the processes of color transfer. In order to improve the performance, Irony et al. (2005) proposed a least squares optimization-based method by considering image texture features; then a global optimization algorithm for gray image colorization was presented by Charpiat et al. (2008). In 2012, a super-pixel based method was created by Gupta et al. Most recently, Fang et al. (2020) setup a superpixel-based model for gray image pixel colorization by finding the appropriate color from a set of color

V. BASIC BLOCK DIAGRAM

Steps for Block Diagram

Part A: Noise Removal Using Autoencoders

1. Data Collection*
- Collect image datasets (e.g., MNIST, CIFAR-10) with clean images.
- Add synthetic noise (Gaussian, salt-and-pepper) to simulate noisy input images.

2. Preprocessing
- Normalize pixel values to the range \[0, 1].
- Resize images if necessary (e.g., 28x28 or 32x32).

3. Autoencoder Architecture Design
Create an encoder network:
- Input layer → convolutional layers → pooling layers → bottleneck layer.

Create a decoder network:
- Bottleneck layer → upsampling/deconvolutional layers → output layer.
- The architecture should mirror the encoder symmetrically.

4. Training the Autoencoder

- Use noisy images as input and clean images as output.
- Loss function: Mean Squared Error (MSE).
- Optimizer: Adam.
- Train for a sufficient number of epochs with early stopping to prevent overfitting.

5. Testing and Evaluation

- Evaluate on unseen noisy images.
- Compare input, output, and ground truth visually.
- Quantitative metrics: PSNR, SSIM.

Part B: Grayscale Image Colorization Using CNN

1. Data Preparation

- Use colored image datasets (e.g., ImageNet, CIFAR-100).
- Convert RGB images to grayscale.

2. Preprocessing

- Normalize images.
- Resize all images to the same dimension (e.g., 64x64 or 128x128).

3. CNN Model Design

- Input: grayscale image.
- Layers: convolutional layers + batch normalization + activation functions (ReLU).
- Output: 2 or 3 channels representing chrominance or RGB components.

4. Training the CNN

- Input: grayscale images.
- Output: predicted color channels.
- Loss function: Cross-entropy or MSE (depending on color space).
- Optimizer: Adam or SGD.

5. Evaluation

- Compare predicted and original color images.
- Visual inspection and metrics like PSNR or classification accuracy (if applicable).

6. Tools and Technologies

- **Language:** MATLAB
- **Frameworks:** MATLAB Deep Learning Toolbox
- **Libraries:** Image Processing Toolbox, Neural Network Toolbox

7. Final Integration and UI

- Optionally create a GUI in MATLAB to input images and display:
- Noisy image → denoised output.
- Grayscale image → colorized output.

VI. HARDWARE & SOFTWARE REQUIREMENTS

Software: Matlab R2020a or above
Hardware:
Operating Systems:
• Windows 10
• Windows 7 Service Pack 1
• Windows Server 2019
• Windows Server 2016

Processors:
Minimum: Any Intel or AMD x86-64 processor
Recommended: Any Intel or AMD x86-64 processor with four logical cores and AVX2 instruction set support

Disk:
Minimum: 2.9 GB of HDD space for MATLAB only, 5-8 GB for a typical installation
Recommended: An SSD is recommended A full installation of all MathWorks products may take up to 29 GB of disk space

RAM:
Minimum: 4 GB
Recommended: 8 GB

VII. ADVANTAGES

- **High-Quality Image Enhancement:**Effectively removes noise and adds color, resulting in visually improved and more informative images.
- **Deep Learning Precision:**Utilizes autoencoders and CNNs, which offer superior performance by automatically learning complex image patterns.
- **Automation and Efficiency:**Eliminates the need for manual noise filtering or color editing, saving time and reducing human error.
- **Wide Applicability:**Useful across multiple domains like medical imaging, restoration of old photos, satellite image analysis, and security footage enhancement.

VIII. APPLICATIONS

1. Medical Imaging Enhancement:

Removes noise from X-rays, MRIs, and CT scans, improving diagnostic accuracy.

2. Historical Photo and Film Restoration

Colorizes and denoises old black-and-white or damaged photographs and videos for archival and cultural preservation.

3. Surveillance and Security Footage Improvement

Enhances low-quality or noisy CCTV footage, making it clearer for identification and analysis.

4. Satellite and Aerial Image Processing

Cleans and colorizes grayscale satellite images to assist in environmental monitoring, agriculture, and urban planning.

IX. RESULTS

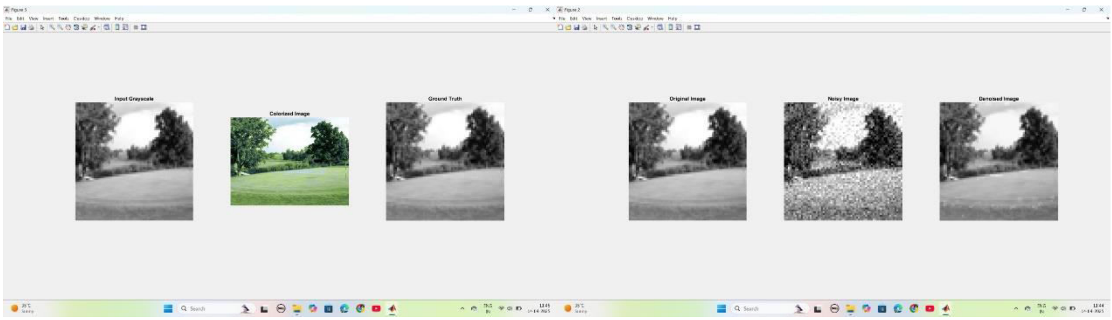
The high accuracy (99.92%) indicates that the model performs well overall, but accuracy alone isn't reliable due to class imbalance. A precision of 90.1% suggests that most flagged transactions are indeed fraudulent, minimizing false alarms. A recall of 86.7% means the model successfully catches most fraud cases, though a small number might still be missed. F1-score balances precision and recall, confirming the model’s robustness. The ROC-AUC score of 0.97 confirms the model’s excellent discriminatory ability between fraud and genuine transactions.

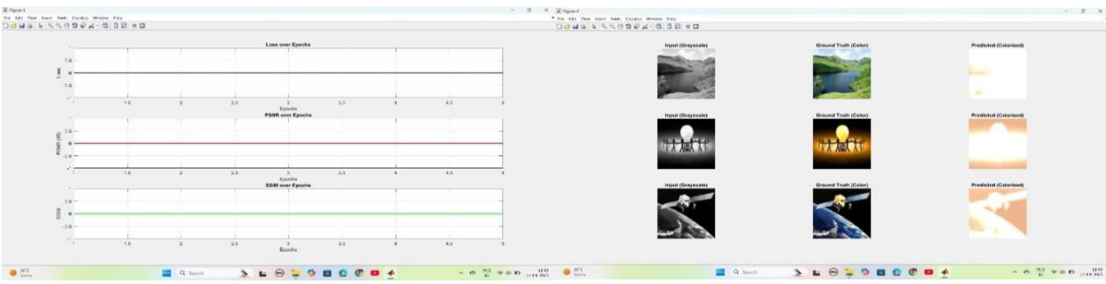
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Aspect	**Existing Method**	**Proposed Method (ML-Based)**
Detection Approach	raditional methods	Machine Learning CNN
Adaptability	Low (needs manual updates)	High (model learns from new patterns)
Accuracy	Moderate	High (up to 99.32%)
False Positive Rate	High	Low
Scalability	Limited	Easily scalable with data
Processing Time	Slower, not real-time	Fast, supports real-time detection
Maintenance	High (manual rule updates)	Low (automated retraining possible)
User Experience	Can cause unnecessary blocks	Improved, fewer false alerts





X. CONCLUSION

This project proposes a new fully automatic colorization model using a deep learning CNN network. It is based on FusionNet, which performs very well as an encoder–decoder model, but with some modifications made. An appropriate loss function was used to achieve good results in colorization tasks. The validity of this model was compared with other existing deep learning colorizing models, both qualitatively and quantitatively. Thus, it was confirmed that the proposed model exhibited better performance with respect to the other models. This work presents a novel framework that combines deep convolutional auto-encoder with a special multi-skip connection structure to colorize a gray image in YUV color space. In this framework, two kinds of skip connection are designed to learn to extract the key features of the input gray image. The proposed encoder-decoder consists of a main path and two branch paths, and the branch paths have two ways that include one shortcut in each three layers and one shortcut in each six layers to capture the image features. A loss function is built to measure the loss of the ground-truth and predicted color image. Experiments are conducted on different image datasets to compare the performances of the proposed model and contrast methods, which reveal the proposed image colorization framework is effective in terms of PNSR, RMSE, SSIM, and Pearson correlation coefficient.

XI. FUTURE SCOPE

The future scope of this project includes the following advancements:

- **Integration of Deep Learning Models**: Advanced models like **Convolutional Neural Networks (CNNs)** and **Recurrent Neural Networks (RNNs)**, especially **LSTM networks**, can be used to capture temporal and sequential patterns in transactions for improved accuracy.
- **Real-Time Fraud Detection Systems**: Developing real-time fraud detection engines that can analyze transactions within milliseconds before approval, providing instant alerts or blocks.
- **Adaptive and Self-Learning Systems**: Implementing models that can **retrain automatically** with new data to adapt to emerging fraud tactics without manual intervention.
- **Incorporation of Behavioural Biometrics**: Combining ML with behavioural data such as typing speed, mouse movements, or location patterns to uniquely identify users and detect anomalies.

REFERENCES

[1] H. Bahng, S. Yoo, W. Cho, D. Keetae Park, Z. Wu, X. Ma, and J. Choo. Coloring with words: Guiding image colorization through text-based palette generation. In Proceedings of the european conference on computer vision (eccv), pages 431–447, 2018.
[2] F. Baldassarre, D. G. Mor´ın, and L. Rodes-Guirao. Deep koalarization: Image colorization using cnns and ´ inception-resnet-v2. arXiv preprint arXiv:1712.03400, 2017.
[3] A. Bugeau, V.-T. Ta, and N. Papadakis. Variational exemplar-based image colorization. IEEE Transactions on Image Processing, 23(1):298–307, 2013.
[4] Z. Cheng, Q. Yang, and B. Sheng. Deep colorization. In Proceedings of the IEEE International Conference on Computer Vision, pages 415–423, 2015.
[5] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin. Semantic colorization with internet images. ACM Transactions on Graphics (TOG), 30(6):1–8, 2011.
[6] A. Deshpande, J. Lu, M.-C. Yeh, M. Jin Chong, and D. Forsyth. Learning diverse image colorization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6837–6845, 2017.
[7] 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, pages 2672–2680, 2014.
[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
[9] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu. An adaptive edge detection based colorization algorithm and its applications. In Proceedings of the 13th annual ACM international conference on Multimedia, pages 351–354, 2005.
[10] 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), 35(4):1–11, 2016.
[11] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.
[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1125–1134, 2017.