Residual Autoencoder with Curriculum Learning for Image Colorization

Md Abdullah Al Mahmud
2013708642
Department of Electrical and Computer Engineering (ECE)
North South University (NSU)
Dhaka, Bangladesh
abdullah.mahmud7@northsouth.edu

B. M Monjur Morshed
2011530642
Department of Electrical and Computer Engineering (ECE)
North South University (NSU)
Dhaka, Bangladesh
monjur.morshed1@northsouth.edu

Taneem Ahmed
2013102042
Department of Electrical and Computer Engineering (ECE)
North South University (NSU)
Dhaka, Bangladesh
taneem.ahmed@northsouth.edu

***Abstract*—This research paper introduces a novel approach for colorizing grayscale images using deep learning. Our method employs a custom-designed convolutional neural network, originally meant for image classification, to disentangle and merge content and style elements from diverse images. By combining grayscale image content with stylistic characteristics derived from similar color images, we add colors in a visually pleasing manner. The colorization process involves a residual autoencoder with curriculum learning. Our approach involves utilizing the Image Colorization Dataset from Kaggle (https://www.kaggle.com/datasets/aayush9753/image-colorization-dataset). We trained the model with color images followed by edge-enhanced colored images, edge-enhanced grayscale images, and grayscale images, respectively. This progressive learning strategy helps the model gradually acquire complex colorization patterns and enhances convergence. Experimental results demonstrate the effectiveness of our model, achieving an MSE Loss of 0.0085 and an MSSSIM Score of 0.9167. To assess image quality, we conduct visual comparisons between our CNN and residual connection approaches, as score-based judgment proves challenging. Overall, our research showcases an innovative and effective method for realistically colorizing grayscale images, leveraging convolutional neural networks, autoencoders, and curriculum learning.**

***Index Terms*—Deep Learning, Colorization, CNN, Residual Autoencoder, Curriculum Learning**

I. Introduction

The process of image colorization has long been a challenging task in computer vision and image processing. Over the years, various techniques and approaches have been proposed to tackle this problem and bring life to grayscale images by adding appropriate colors. In recent times, deep learning models, particularly autoencoders, have shown promising results in image colorization tasks.

In this report, we present our project titled "Residual Autoencoder for Image Colorization with Curriculum Learning." Our objective was to explore the capabilities of autoencoders in the context of image colorization and propose an improved architecture to address the limitations observed in existing models.

To begin with, we delved into the fundamental workings of autoencoders, a type of neural network that aims to reconstruct its input data at the output layer. Autoencoders have been widely used for various tasks, including image reconstruction, denoising, and compression. We thoroughly studied the principles behind autoencoders and gained insights into their potential for image colorization.

We initially implemented a basic autoencoder with a convolutional neural network (CNN) architecture. Although this approach yielded acceptable results, we observed that it failed to adequately colorize different components within an image. This limitation prompted us to investigate alternative techniques to enhance the colorization process.

Inspired by the paper "Deep Koalarization: Image Colorization using CNNs and Inception-Resnet-v2" [1], which proposed the fusion layer by concatenating the output of a pretrained Inception-Resnet-v2 model with the CNN encoder's output, we sought to introduce a novel architecture that could overcome the identified shortcomings. Instead of employing the fusion layer, we replaced the CNN layers with residual blocks within our autoencoder architecture.

The introduction of the residual block aimed to enhance the colorization capabilities by leveraging residual connections and facilitating the propagation of important features through the network. This approach, inspired by the findings of He et al. in their work on residual networks [2], demonstrated the potential for improving image colorization outcomes.

Inspired by the concept of curriculum learning, which proposes a gradual and structured learning process by presenting training samples in an ordered manner, we incorporated this approach into our colorization framework. By carefully designing a curriculum of training samples, we aimed to guide the learning process to focus on simpler cases before tackling

more complex colorization scenarios.

The curriculum learning strategy allowed our model to gradually learn and refine its colorization capabilities, starting from simpler images with clear features and gradually progressing to more challenging cases. This enabled our model to better capture and reproduce color patterns, resulting in improved colorization results across a wide range of images.

Throughout this report, we present a detailed analysis of our proposed "Residual Autoencoder for Image Colorization with Curriculum Learning" model. We evaluate its performance on various datasets and compare it with existing state-of-the-art techniques. Our experiments showcase the effectiveness of the residual block in enhancing the colorization process and its ability to accurately colorize different components within images.

This Project contributes to the field of image colorization by proposing an improved autoencoder architecture that leverages residual connections to achieve superior results. By addressing the limitations observed in existing models, our approach opens up new possibilities for enhancing the realism and quality of colorized images.

II. Related Work

Image colorization, as a challenging task in computer vision, has garnered significant attention in recent years. Autoencoders have emerged as a popular approach for image colorization due to their ability to learn meaningful representations of input data. Several studies have explored the use of autoencoders in image colorization, demonstrating their effectiveness in generating accurate and visually appealing colorizations.

Autoencoder-based approaches typically involve training an encoder-decoder architecture, where the encoder encodes the input grayscale image into a lower-dimensional latent space, and the decoder reconstructs the colored output image from the latent representation. This framework enables the model to learn a compressed representation of the input image while simultaneously learning to reconstruct the color information [4].

Convolutional autoencoders, which incorporate convolutional layers to capture local spatial dependencies, have been widely used for image colorization. These architectures excel at capturing fine-grained features and spatial relationships within the image [5]. Recurrent autoencoders, leveraging recurrent connections, have also been explored to model sequential dependencies in the image data, enabling more accurate and coherent colorization results [6].

The integration of residual connections in autoencoder architectures has been found to enhance the model's ability to capture fine details and improve colorization results. Residual connections allow for the direct flow of information from earlier layers to later layers, enabling the model to learn residual functions and alleviate the vanishing gradient problem [7].

Additionally, the concept of curriculum learning has been applied to the training process of autoencoder-based models for image colorization. Curriculum learning involves gradually increasing the difficulty of the training examples presented to the model. For image colorization, this can be achieved by starting with color images and progressively transitioning to grayscale images or edge-enhanced images. This curriculum allows the model to learn color distributions, edge information, and fine details in a more structured and effective manner [8].

Previous studies have demonstrated the effectiveness of autoencoder-based approaches for image colorization. Zhang et al. proposed the Colorful Image Colorization model, which employed a deep autoencoder architecture to predict color abstractions from grayscale images [4]. Iizuka et al. introduced the Let there be Color! model, which utilized a conditional autoencoder to generate colorizations conditioned on user hints [5]. Guadarrama et al. proposed PixColor, a recurrent autoencoder-based model that utilized a sequence-to-sequence framework for image colorization [6].

III. Model Architecture

A. Residual Block:

In our proposed *Residual Autoencoder for Image Colorization with Curriculum Learning* architecture, we introduced three different types of residual blocks as the building blocks of our model: *IdentityBlock*, *ConvolutionalBlock*, and *DeconvolutionalBlock*. These blocks are responsible for capturing and propagating important features through the network, facilitating the enhancement of colorization results.

1. IdentityBlock:

The IdentityBlock serves as the simplest residual block in our architecture. It consists of two convolutional layers, each followed by a ReLU activation function. The first convolutional layer operates on the input tensor, while the second convolutional layer processes the output of the first layer. The final output of the block is obtained by adding the original input tensor (identity) to the output of the second convolutional layer. This connection allows for the preservation and direct propagation of important features from the input to the output, enabling the model to capture fine details during colorization.

2. ConvolutionalBlock:

The ConvolutionalBlock is designed to handle down-sampling operations in the network. It consists of three convolutional layers, each followed by a ReLU activation function. The first convolutional layer applies a 1x1 kernel to reduce the number of input channels and adjust the spatial dimensions, if necessary. The subsequent convolutional layer, with a 3x3 kernel, further processes the data, followed by another 1x1 convolutional layer. Batch normalization is applied to stabilize the learning process and improve the efficiency of the network. Additionally, a down-sampling shortcut connection is introduced to match the dimensions of the input and output tensors. By incorporating this block, the model can capture and utilize both local and global context information during the colorization process.

3. DeconvolutionalBlock:

The DeconvolutionalBlock serves as the counterpart to the ConvolutionalBlock and handles up-sampling operations. It consists of three transpose convolutional layers, each followed

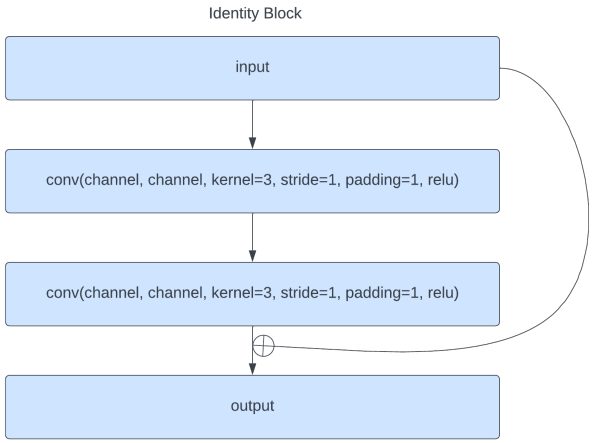

Fig. 1. Identity Block

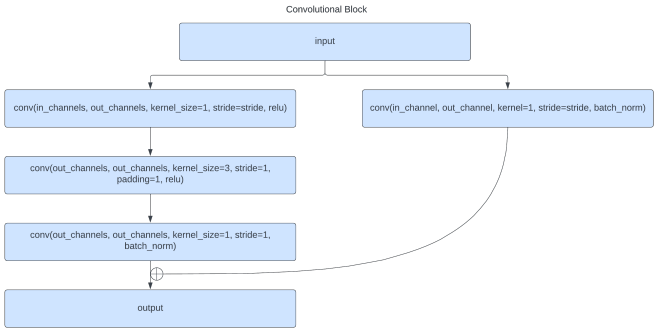

Fig. 2. Convolutional Block

by a ReLU activation function. The first transpose convolutional layer, with a 4x4 kernel, performs up-sampling to increase the spatial dimensions of the input tensor. The subsequent transpose convolutional layer, with a 3x3 kernel, further processes the data, followed by a final 1x1 convolutional layer. Similar to the ConvolutionalBlock, batch normalization is applied to improve the stability and efficiency of the network. Additionally, a shortcut connection with a matching transpose convolutional operation is included to facilitate the flow of information from the input to the output. By incorporating this block, the model can effectively capture and utilize high-level semantic information during the colorization process.

These residual blocks play a crucial role in our proposed architecture, enabling the network to capture and propagate important features throughout the colorization process. By leveraging the strengths of each block, our model can achieve improved performance and generate more accurate and visually appealing colorized images.

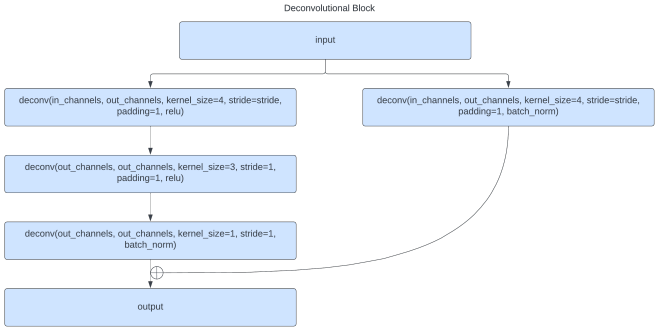

Fig. 3. Deconvolutional Block

B. Curriculum Learning

In our project, we incorporated the concept of curriculum learning during the training of our model for image colorization. Curriculum learning is a training strategy inspired by educational curricula, where the learning process starts with simpler and well-structured tasks and gradually progresses to more complex and challenging ones. This approach aims to facilitate the learning process by gradually exposing the model to increasingly difficult examples.

In the context of image colorization, we designed a curriculum learning strategy that involved training the model on different types of images in a specific order. Specifically, we followed the curriculum progression from color images to edge-enhanced colored images, then to edge-enhanced grayscale images, and finally to grayscale images. The rationale behind this curriculum is to gradually introduce the model to more challenging tasks, allowing it to learn progressively and improve its performance over time.

By starting with color images, the model initially learns to capture and reproduce the color information present in the training data. This phase helps the model develop a basic understanding of color distribution and relationships in images.

Next, we move on to training the model on edge-enhanced colored images. These images provide an additional level of complexity by incorporating edge information. By exposing the model to edge-enhanced images, it learns to capture not only color information but also the structural details and boundaries between different objects in the image.

In the subsequent phase, the model is trained on edge-enhanced grayscale images. These images remove the color information entirely and emphasize the edge details. By training on edge-enhanced grayscale images, the model focuses on learning to extract and reproduce accurate edge information, enhancing its ability to capture fine details and contours.

Finally, we train the model on grayscale images, which pose the most challenging task for the model. Without any color or edge cues, the model relies solely on learned representations to generate plausible colorizations. This phase helps the model refine its understanding of image structure and texture, en-

abling it to produce more accurate colorizations even in the absence of explicit edge information.

The curriculum learning approach employed in our project allows the model to gradually learn and refine its colorization capabilities in a structured manner. By exposing the model to different levels of complexity, we aim to enhance its ability to generalize and produce high-quality colorized outputs.

C. Residual Autoencoder

The encoder component learns to extract meaningful features from the input image by applying a series of convolutional and activation layers. These layers progressively reduce the spatial dimensions and increase the number of channels, capturing high-level representations of the image.

The decoder component, on the other hand, takes the latent representation and generates a colorized image. It mirrors the structure of the encoder in a reverse manner, employing deconvolutional layers to upsample the features and reconstruct the image with appropriate color information.

The autoencoder as a whole is trained to minimize the reconstruction error between the input image and the output image produced by the decoder. By doing so, the autoencoder learns to capture the essential features necessary for accurate image reconstruction, effectively enabling the colorization of grayscale images.

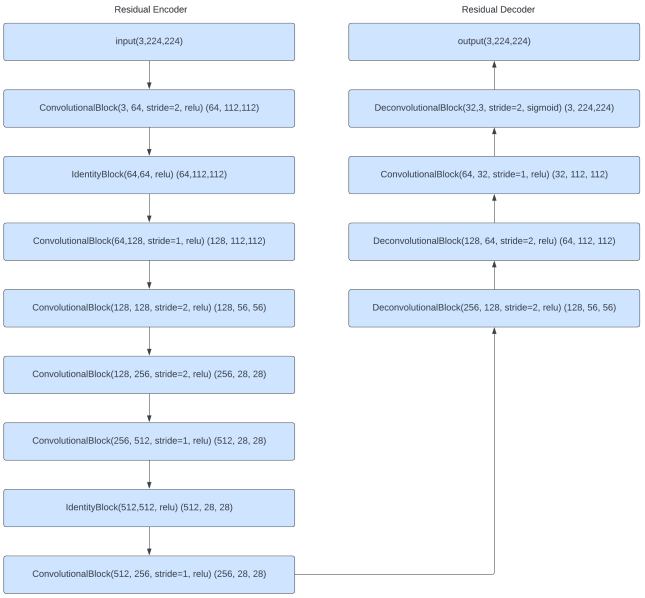

Fig. 4. Residual Autoencoder

IV. Experimental Setup

We used the (1-MSSSIM) loss function as the primary objective during the training phase. The (1-MSSSIM) loss measures the dissimilarity between the predicted colorized image and the ground truth color image, with a lower loss value indicating better similarity [9].

To optimize the model, we utilized the Adam optimizer, which is a popular choice for training neural networks. We set the learning rate to 0.001, which determines the step size for updating the model's parameters during optimization.

The training process was carried out for a total of 100 epochs, where each epoch represents a complete iteration over the entire training dataset. This number of epochs was chosen to allow the model sufficient time to learn the colorization task and converge to a stable solution.

For the purpose of validation, we split our dataset into training and validation subsets. The dataset consisted of a total of 5000 training images, out of which 4500 images were used for training the model, and the remaining 500 images were used for validation. This division allowed us to assess the model's performance on unseen data and monitor its generalization ability.

In order to train our model, we took help of Google Colab. The free acount of Google Colab provides Nvidia Tesla T4 15GB GPU. Our model took 3hrs to complete 100 epochs on this GPU.

During the validation phase, we evaluated the model's performance using different metrics. For testing the model's colorization accuracy, we employed the Mean Squared Error (MSE) loss metric, which quantifies the pixel-wise difference between the predicted colorized image and the ground truth color image. Additionally, we utilized the MSSSIM (Multi-Scale Structural Similarity) score as a performance metric, which measures the structural similarity between the predicted and ground truth images at multiple scales.

For the final testing of the trained model, we employed the separate testing dataset consisting of 739 images. During this stage, we assessed the model's performance using the MSE loss and MSSSIM score to evaluate its colorization accuracy and overall quality of the colorized outputs.

V. Result

The performance of our image colorization models was evaluated using two key metrics: Test MSE Loss and Test MSSSIM Score. The models underwent various training approaches and utilized different architectural configurations.

Model	Test MSSSIM Score	Test MSE Loss
CNN + Curriculum	0.9258	0.0097
Residual	0.9136	0.0089
Residual + Curriculum (Old)	0.9160	0.0089
Residual + Curriculum (Final)	0.9167	**0.0085**

Our final model achieved a Test MSE Loss of 0.0085 and a Test MSSSIM Score of 0.9167. This model employed residual blocks and a residual autoencoder. The training process followed a curriculum learning approach, where the first 15 epochs were trained with colored pictures as input. Subsequently, 15 epochs were trained with edge-enhanced colored pictures, followed by 35 epochs with edge-enhanced grayscale images. The remaining 35 epochs involved training with normal grayscale images.

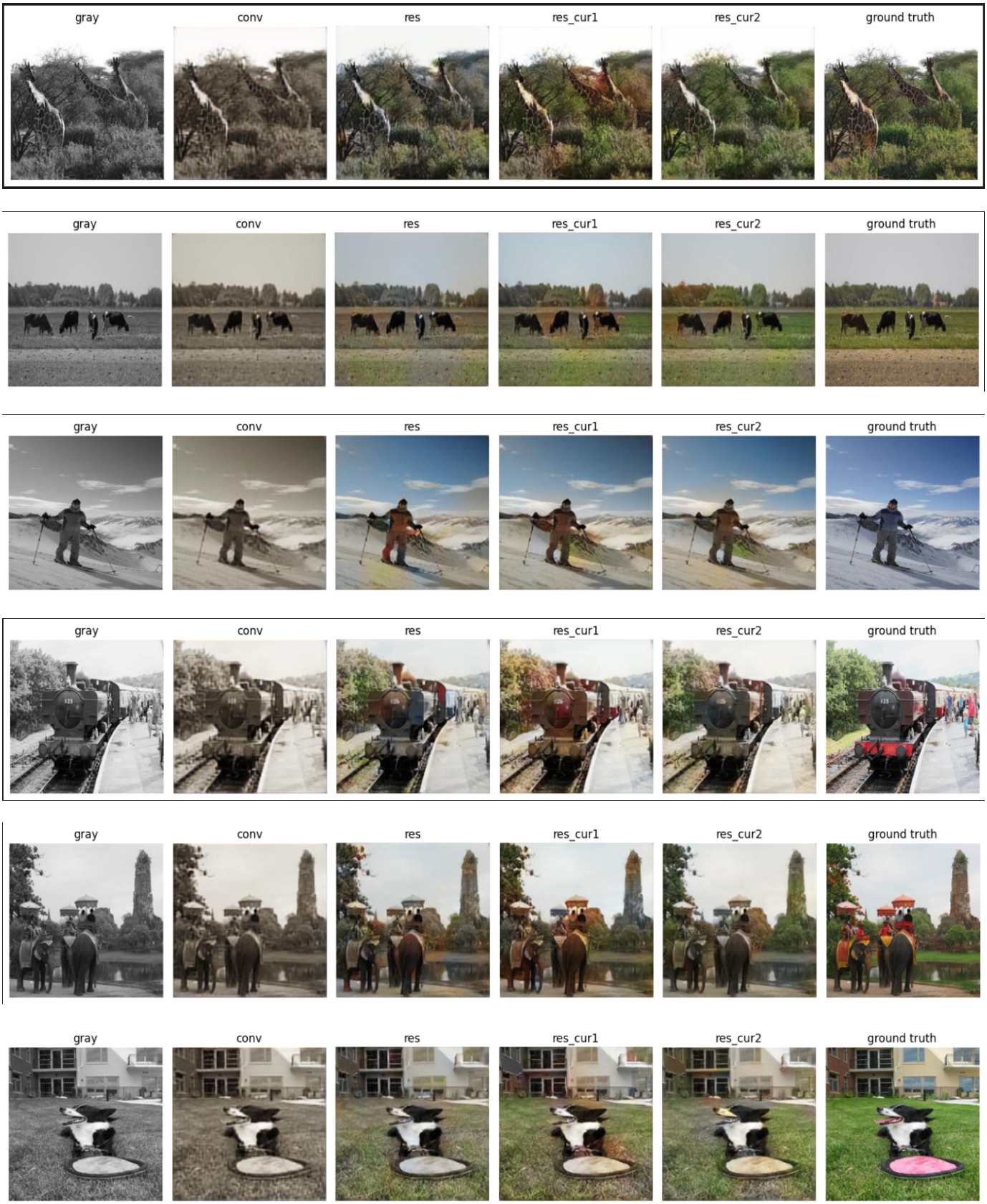

Fig. 5. starting from the left, we observe the following images: the input image, the generated output by the CNN autoencoder model, the generated output from the Residual model, the generated output from the Residual Curriculum model, and immediately after, the generated output from our final Residual Curriculum model. Finally, we have the Ground Truth image

In comparison, the other three models were outperformed by our final model. The first model obtained a Test MSE Loss of 0.0097 and a Test MSSSIM Score of 0.9258. It utilized a CNN autoencoder and followed a similar curriculum learning process as our final model.

The second model achieved a Test MSE Loss of 0.0089 and a Test MSSSIM Score of 0.9136. It utilized a residual autoencoder and underwent a different curriculum learning process. The model was trained for 100 epochs using normal grayscale images as input.

The third model employed both residual blocks and a residual autoencoder. It followed a curriculum learning approach where the first 25 epochs were trained with colored pictures as input, followed by 25 epochs with edge-enhanced colored pictures, 25 epochs with edge-enhanced grayscale images, and the remaining 25 epochs with normal grayscale images. This model achieved a Test MSE Loss of 0.0089 and a Test MSSSIM Score of 0.9160.

Despite the CNN model outperforming our final model in terms of MSSSIM score, a thorough examination of the output images reveals a substantial bias present in the CNN's results. Regrettably, the CNN model exhibits a limited approach by solely applying a sepia color overlay to the input images, thereby failing to capture the true essence and intricate details of the images. In figure 5, we can see that our final model surpasses the CNN by generating images that closely resemble the original inputs, showcasing a remarkable ability to produce highly accurate representations and faithfully preserve essential visual elements. The superiority of our final model becomes evident when considering its exceptional fidelity in generating images that closely resemble the original inputs, elevating its overall performance beyond the quantitative measure of MSSSIM score.

CONCLUSION

In this study, we compared the performance of four image enhancement models utilizing different architectural configurations and training approaches. Our findings highlight the importance of both architectural design and training approach in image enhancement models. The incorporation of residual blocks and a residual autoencoder in our final model contributed to its superior performance. Additionally, the curriculum learning strategy, gradually exposing the model to different image types during training, played a crucial role in enhancing the model's ability to handle various image characteristics.

Further research can explore alternative architectural configurations and training methodologies to potentially improve image enhancement performance. Additionally, investigating the generalizability of these models across different datasets and image types would provide valuable insights for real-world applications.

Overall, our study showcases the potential of residual blocks and a residual autoencoder in image enhancement tasks and emphasizes the significance of curriculum learning for achieving superior performance in such models.

REFERENCES

[1] Baldassarre, F. (2017, December 9). Deep Koalarization: Image Colorization using CNNs and Inception-ResNet-v2. arXiv.org. https://arxiv.org/abs/1712.03400

[2] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[3] Z. Wang, E.P. Simoncelli, and A.C. Bovik, "Multi-Scale Structural Similarity for Image Quality Assessment," in *Proceedings of the 37th Asilomar Conference on Signals, Systems, and Computers*, Vol. 2, pp. 1398-1402, 2003.

[4] Zhang, Richard, et al. "Colorful image colorization." *European Conference on Computer Vision*. Springer, Cham, 2016.

[5] Iizuka, Satoshi, Edgar Simo-Serra, and Hiroshi 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 (2016): 110.

[6] Guadarrama, Sergio, et al. "PixColor: Pixel recursive colorization." *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017.

[7] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[8] Bengio, Yoshua, et al. "Curriculum learning." *ICML* 2009.

[9] Z. Wang, E.P. Simoncelli, and A.C. Bovik, "Multi-Scale Structural Similarity for Image Quality Assessment," in *Proceedings of the 37th Asilomar Conference on Signals, Systems, and Computers*, Vol. 2, pp. 1398-1402, 2003.