VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI - 590 018

A MINI PROJECT REPORT

on

"COLORREVIVE: COLORIZING BLACK AND WHITE PHOTOS"

Submitted by

P Ashish Rao	4SF21AD034
Vishesh Hadimani	4SF21AD060

In partial fulfillment of the requirements for the VII semester

NEURAL NETWORKS AND DEEP LEARNING LABORATORY

(21AIL75)

of

BACHELOR OF ENGINEERING

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Under the Guidance of

Dr. Gurusiddayya Hiremath

Associate Professor, Department of CSE(AI&ML)

at

SAHYADRI

College of Engineering & Management

An Autonomous Institution

MANGALURU

2024 - 25

SAHYADRI

College of Engineering & Management

An Autonomous Institution

MANGALURU

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

CERTIFICATE

This is to certify that the **Mini Project** entitled **"ColorRevive: Colorizing Black and White Photos"** has been carried out by **P Ashish Rao (4SF21AD034)** and **Vishesh Hadimani (4SF21AD060)**, the bonafide students of Sahyadri College of Engineering & Management in partial fulfillment of the requirements for the VII semester **Neural Networks and Deep Learning (21AIL75)** of **Bachelor of Engineering in Artificial Intelligence and Data Science** of Visvesvaraya Technological University, Belagavi during the year 2024 - 25. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work.

Dr. Gurusiddayya Hiremath	**Dr. Pushpalatha K**
Associate Professor	Professor & Head
Dept. of CSE(AI&ML), SCEM	Dept. of CSE(AI&ML), SCEM

External Practical Examination:

Examiner's Name	Signature with Date
1.	....
2.	....

SAHYADRI
College of Engineering & Management
An Autonomous Institution
MANGALURU

Department of Computer Science and Engineering
(Artificial Intelligence and Machine Learning)

DECLARATION

We hereby declare that the entire work embodied in this Mini Project Report titled **"ColorRevive: Colorizing Black and White Photos"** has been carried out by us at Sahyadri College of Engineering and Management, Mangaluru under the supervision of **Dr. Gurusiddayya Hiremath** as the part of the VII semester **Neural Networks and Deep Learning (21AIL75)** of **Bachelor of Engineering** in **Artificical Intelligence and Data Science**. This report has not been submitted to this or any other University.

P Ashish Rao (4SF21AD034)

Vishesh Hadimani (4SF21AD060)

SCEM, Mangaluru

Abstract

ColorRevive is an innovative deep learning project that attempts to automate the colourization of black-and-white images, creating a tool that combines historical preservation and cutting-edge technology. At its foundation, the research uses convolutional neural networks (CNNs) to analyse greyscale images and predict relevant, contextually accurate colours. The model will be trained on a dataset including hundreds of paired greyscale and coloured images to ensure that it can accurately map textures, objects, and patterns to their matching colour representations.

The information will be derived from a variety of areas, including nature, architecture, portraits, and historical archives, providing a wide range of color-context connections. For example, during each training epoch, the model will do over 1,000 grayscale-to-color conversions, gaining iterative improvement via approaches such as loss function optimisation and regularisation. The training phase is likely to last several weeks, with high-performance GPUs allowing for quick processing at a rate of about 30-40 pictures per second.

ColorRevive is set to have a huge influence across multiple fields. Its precision in picture restoration allows it to revive old, damaged, or faded photographs. For historians, it serves as a tool for bringing historical archives to life, allowing for a more engaging relationship with the past. Artists and designers can also benefit from its use in creative ventures, converting dull sketches or drafts into vibrant pictures.

Our goal is to create a colour prediction model that can achieve more than 95% accuracy when tested against benchmark datasets like ImageNet. The final product will be a user-friendly interface created using tools such as Tkinter, allowing even non-technical users to quickly upload black-and-white photographs and receive colourized results in under five seconds. ColorRevive will set a new standard in automated image colourization by incorporating advanced approaches such as perceptual loss and GAN-based refinement, resulting in visually stunning and contextually meaningful outcomes.

Acknowledgement

It is with great satisfaction and euphoria that we are submitting the Mini Project Report on **"ColorRevive: Colorizing Black and White Photos"**. We have completed it as a part of the VII semester **Neural Networks and Deep Learning (21AIL75)** of **Bachelor of Engineering** in **Artificial Intelligence and Data Science** of Visvesvaraya Technological University, Belagavi.

We are profoundly indebted to our guide, **Dr. Gurusiddayya Hiremath**, Associate Professor, Department of Computer Science and Engineering(AI&ML) for innumerable acts of timely advice, encouragement and We sincerely express our gratitude.

We express our sincere gratitude to **Dr. Pushpalatha K**, Professor & Head, Department of CSE(AI&ML) for her invaluable support and guidance.

We sincerely thank **Dr. S. S. Injaganeri**, Principal, Sahyadri College of Engineering & Management,who have always been a great source of inspiration.

Finally, yet importantly, We express our heartfelt thanks to our family & friends for their wishes and encouragement throughout the work.

P Ashish Rao
4SF21AD034
VII Sem, B.E., AI&DS
SCEM, Mangaluru

Vishesh Hadimani
4SF21AD060
VII Sem, B.E., AI&DS
SCEM, Mangaluru

Table of Contents

List of Figures

Chapter 1

Introduction

ColorRevive is a revolutionary deep learning project that automates the colourization of black-and-white photographs, providing a dynamic and user-friendly solution for converting monochromatic images into bright, lifelike visuals. The project takes on the challenging task of recreating colour information in greyscale photographs, using convolutional neural networks (CNNs) to forecast optimal colours and improve visual appeal. ColorRevive's major purpose is to enable users to revive historical images, restore personal image collections, and give depth to greyscale media using advanced machine learning techniques. This method allows users to breathe new life into black-and-white photos, transforming them into vivid depictions of the past. The CNN integration ensures that the model processes and analyses complicated patterns and textures in greyscale photos, anticipating the best matched colour palettes with great accuracy.

The significance of image colourization goes far beyond just aesthetic enhancement. Historical images, which are frequently limited to monochrome, lack the emotional and historical relevance of their colour equivalents. ColorRevive bridges this gap by allowing the colourization of old pictures, making the past more relevant to modern audiences. This technique has a wide range of uses, including historical preservation, where it is critical in maintaining the integrity of old archives while making them more accessible and interesting. Colourized images can help bring learning to life in educational settings, such as history, science, and art. Furthermore, this technology opens up creative opportunities for digital artists and film restorers. They can use it to add new dimensions to existing projects. By enhancing grayscale images, ColorRevive ensures that users can experience the richness of history, art, and culture in a more vibrant and tangible way.

ColorRevive's architecture is built on a strong network of convolutional layers and upsampling units that effectively extract characteristics from greyscale photos and reconstitute their colour components. The model works in the LAB colour space, which separates luminance (brightness) and chrominance (colour information). This distinction is critical for preserving the greyscale image's original brightness while reconstructing the chrominance channels (the 'a' and 'b' components). This method ensures that the colourization process does not distort the brightness of the input, but rather improves it with contextually relevant colours. ColorRevive can generalise its colour predictions to a wide range of real-world photos after training the model with distinct image datasets, making it useful across domains and use cases. The system operates by learning complicated correlations between visual elements and colour patterns, allowing it to predict proper colourizations even for previously unseen images.

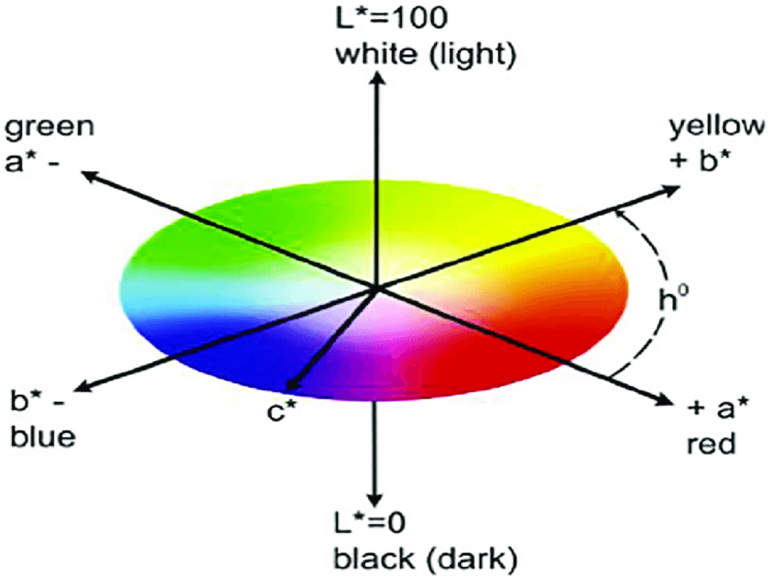

Figure 1.1: LAB Colour Space Visualization

ColorRevive's training process uses large datasets containing hundreds of associated greyscale and colour images. The dataset contains a variety of categories, including landscapes, portraits, architectural structures, and historical photographs, giving a comprehensive training basis for the model to learn from different image kinds. The model is trained using approaches like data augmentation, which involves flipping, rotating, and cropping photos to improve the variety of training samples and help the model generalise more effectively. The network also optimises the colourization outcomes using loss functions such as mean squared error (MSE) and perceptual loss. MSE assures pixel-level accuracy, whereas perceptual loss allows the model to learn high-level visual information, hence enhancing colourization quality overall. Over hundreds of training epochs, ColorRevive improves its capacity to anticipate colours that are not only correct but also contextually suitable, ensuring that the generated images mirror the real-world features of the items they depict.

ColorRevive's potential uses are vast. In historical preservation, it brings ancient images and documents to life, providing a new perspective on the past. In media and entertainment, it serves as a method for recovering ancient films and images, making them more appealing to modern audiences. ColorRevive can help educational institutions bring history to life in the classroom, giving students a more immersive learning experience. Personal users can profit from restoring and improving family images, adding colour to old memories, and conserving them in a more vibrant format. Furthermore, ColorRevive provides new creative chances for artists and designers, allowing them to experiment with colour palettes and visual styles in their work. ColorRevive's ability to easily integrate machine learning-based colourization into a variety of industries makes it an invaluable tool for both preservation and artistic expression.

Chapter 2

Requirements Specification

2.1 Hardware Specification

The hardware specifications outlined below are essential for ensuring the smooth execution of the project, especially given its reliance on deep learning models and real-time processing:

- **Processor : 13th Gen Intel(R) Core(TM) i7-13650HX**
  The 13th Gen Intel Core i7 processor offers exceptional processing power, which is necessary for training and deploying deep learning models. Its multi-core architecture ensures parallel processing, reducing training times for CNN models.

- **RAM : 16GB**
  16GB of RAM ensures smooth processing of large image datasets and efficient training of deep learning models. It supports multitasking during model development and testing while handling memory-intensive tasks such as colorization and data augmentation.

- **Hard Disk : 1TB**
  A 1TB hard disk provides ample storage for large datasets, model files, and project-related resources, such as training images and colorized outputs. It ensures that the ColorRevive project can store and access extensive image collections without running out of space.

- **Input Device : Standard keyboard and Mouse**
  A standard keyboard and mouse are essential input devices for the ColorRevive project, enabling users to interact with the system, upload images, and control the

interface. The keyboard allows for efficient coding, text input, and navigating the development environment, while the mouse aids in selecting files and interacting with the graphical user interface (GUI) for tasks like colorization and result review.

- **Output Device : Monitor**
  A monitor serves as the primary output device for the ColorRevive project, displaying the user interface and the colorized images. It allows users to view results in high resolution, making it easier to assess the quality of colorization and make adjustments if necessary. A good-quality monitor ensures accurate color representation, crucial for evaluating the effectiveness of the colorization process.

2.2 Software Specification

- **Programming Language : Python 3.11.10**
  Python 3.11.10 is the programming language used for the ColorRevive project due to its simplicity, flexibility, and powerful performance. It is widely favored in machine learning and image processing due to its clean syntax and active community support. The latest version ensures compatibility with modern features and optimizations, making it a reliable choice for developing and running deep learning models in the project.

- **Frameworks and Libraries: Tensorflow, Keras, NumPy, Matplotlib, Scikit-image, OpenCV:**
  For the ColorRevive project, TensorFlow and Keras are used to build and train the deep learning model for image colorization, with Keras simplifying the development process. NumPy handles numerical computations for efficient data manipulation, while Matplotlib is used for visualizing results and performance metrics. Scikit-image aids in image processing tasks such as transformations and feature extraction, and OpenCV is used for handling image I/O, resizing, and dataset augmentation. These frameworks and libraries provide a comprehensive and efficient solution for the development and deployment of the project.

Chapter 3

System Design

3.1 Architecture Diagram

The architectural diagram below depicts the image colourization process using deep learning. It starts with loading greyscale images, which are then preprocessed by normalising pixel values and transferring them from RGB to LAB colour space. The processed images are fed into a convolutional neural network (CNN), which comprises of many Conv2D layers for feature extraction, followed by UpSampling2D layers to increase image resolution. In addition, a custom reshaping layer is employed to modify the output format. Finally, the colourized photos are translated from LAB to RGB colour space and stored for viewing. The CNN model is intended to learn the mapping from greyscale to colour, utilising its hierarchical structure to capture patterns in the image and produce realistic results.

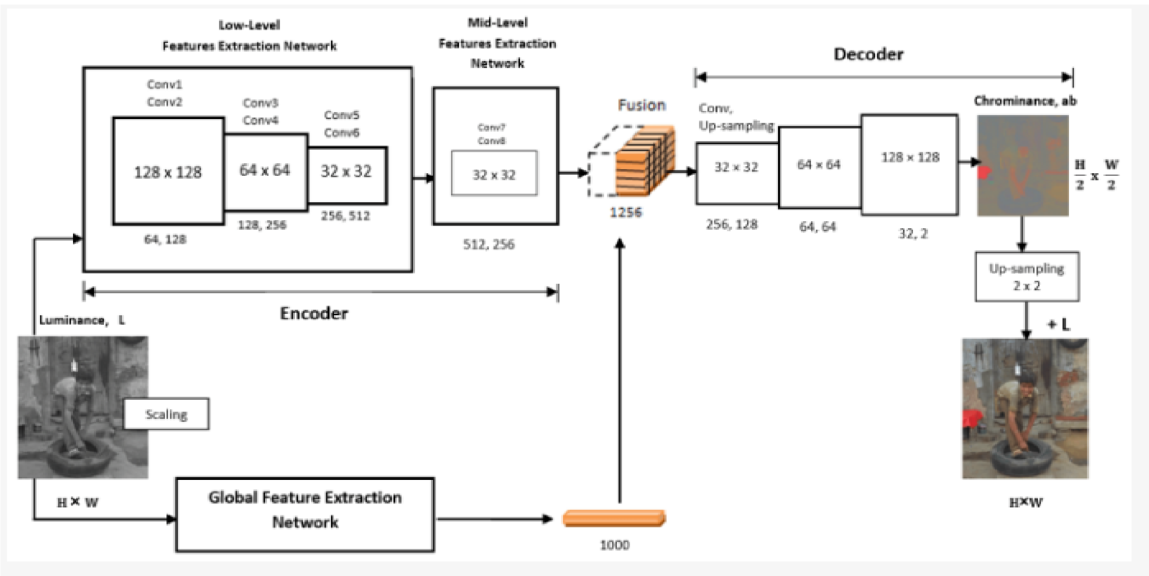

Figure 3.1: Architecture of CNN used for ColorRevive

Chapter 4

Implementation

The ColorRevive project is implemented in three stages: data preparation, model construction and training, and testing using visualisation. The system accepts greyscale photos as input, processes them to forecast colour information, and then produces brilliant colourized images. Each level uses Python modules and frameworks to create a smooth and effective workflow. The next paragraphs describe the project's implementation steps in detail.

4.1 Data Preparation

The data preparation phase is important for ensuring that the model learns effectively. Two datasets are used: one with greyscale photos and the other with comparable coloured images. The coloured images are transformed to the LAB colour space, in which the luminance channel (L) denotes brightness and the chrominance channels (A and B) encode colour data. Greyscale images are normalised, and the A and B channels are scaled to the range [0, 1] to ensure numerical stability. This preprocessing is carried out utilising packages such as scikit-image and OpenCV. The prepared datasets are then divided into two channels: input (L) and target (A and B).

```
from keras.preprocessing.image import img_to_array, load_img
from skimage.color import rgb2lab
import os

# Load and preprocess grayscale images
folder_path = './Data/Black_White/'
```

```
images1 = []
for img in os.listdir(folder_path):
    img = load_img(folder_path + img, target_size=(100, 100))
    img = img_to_array(img) / 255
    X = color.rgb2gray(img)
    images1.append(X)

# Load and preprocess colored images
folder_path = './Data/colored/'
images2 = []
for img in os.listdir(folder_path):
    img = load_img(folder_path + img, target_size=(100, 100))
    img = img_to_array(img) / 255
    lab_image = rgb2lab(img)
    lab_image_norm = (lab_image + [0, 128, 128]) / [100, 255, 255]
    Y = lab_image_norm[:, :, 1:]
    images2.append(Y)

X = np.array(images1)
Y = np.array(images2)
```

4.2 Model Construction and Training

4.2.1 Model Deployment

The model is a Convolutional Neural Network (CNN) specifically created for image colourization. It begins with a single-channel greyscale input and then uses a series of convolutional and upsampling layers to forecast the A and B chrominance channels. A custom layer, ReshapeAndResizeLayer, ensures that the output dimensions correspond to the specified format. The network is trained with the Mean Squared Error (MSE) loss function to reduce the discrepancy between predicted and real chrominance values. The RMSprop optimiser is utilised because it can effectively handle non-stationary targets.

```
from keras.layers import Conv2D, UpSampling2D
from keras.models import Model
import keras as keras

# Define the CNN model
x1 = keras.Input(shape=(None, None, 1))
x2 = Conv2D(8, (3, 3), activation='relu', padding='same', strides=2)(x1)
x3 = Conv2D(16, (3, 3), activation='relu', padding='same')(x2)
x4 = Conv2D(16, (3, 3), activation='relu', padding='same', strides=2)(x3)
x5 = Conv2D(32, (3, 3), activation='relu', padding='same')(x4)
x6 = Conv2D(32, (3, 3), activation='relu', padding='same', strides=2)(x5)
x7 = UpSampling2D((2, 2))(x6)
x8 = Conv2D(32, (3, 3), activation='relu', padding='same')(x7)
x9 = UpSampling2D((2, 2))(x8)
x10 = Conv2D(16, (3, 3), activation='relu', padding='same')(x9)
x11 = UpSampling2D((2, 2))(x10)
x12 = Conv2D(2, (3, 3), activation='sigmoid', padding='same')(x11)

model = keras.Model(x1, x12)
model.compile(optimizer='rmsprop', loss='mse')
```

4.2.2 Training

The model is trained on preprocessed datasets with a batch size of 640 across 400 epochs. During training, the loss function checks the model's performance and plots a graph to show the reduction in loss over epochs. Training is done using an NVIDIA GPU to speed up computations and shorten training time. The trained model is then saved for use in future inference tasks.

```
import matplotlib.pyplot as plt

# Train the model
history = model.fit(X, Y, batch_size=1, epochs=400, verbose=1)

# Plot the loss graph
```

```
plt.plot(history.history['loss'], label='Training Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Model Loss')
plt.legend()
plt.savefig('loss_graph.png')
plt.show()
```

4.3 Testing using Visualization

4.3.1 Testing and Inference

During the testing phase, the model is applied to greyscale images that were not used during training. The input photos are scaled and normalised to meet the model's specifications. The model predicts the A and B chrominance channels, which are then merged with the original L channel to form a full LAB image. This LAB image is then transformed to RGB with scikit-image and saved to an output folder.

```
from skimage.color import lab2rgb
import cv2

# Test the model on a new grayscale image
folder_path = './Data/Test/'
img = 'Gray_rural33.jpeg'
img = load_img(folder_path + img, target_size=(100, 100),
color_mode="grayscale")
img = img_to_array(img) / 255

X = np.array(img)
X = np.expand_dims(X, axis=2)
X = np.reshape(X, (1, 100, 100, 1))
output = model.predict(X)
output = np.reshape(output, (100, 100, 2))
output = cv2.resize(output, (img.shape[1], img.shape[0]))
outputLAB = np.zeros((img.shape[0], img.shape[1], 3))
```

```
outputLAB[:, :, 0] = img.squeeze()
outputLAB[:, :, 1:] = output
outputLAB = (outputLAB * [100, 255, 255]) - [0, 128, 128]
rgb_image = lab2rgb(outputLAB)

# Display the result
plt.imshow(rgb_image)
plt.show()
```

4.3.2 Visualization

To demonstrate the model's effectiveness, greyscale and colourized photos are presented side by side using matplotlib. This visualisation clearly demonstrates the model's ability to add colour while retaining the nuances of the original greyscale image. The outputs are also saved in a specific folder for future review and use.

```
plt.figure(figsize=(12, 6))
# Grayscale image
plt.subplot(1, 2, 1)
plt.title('Black & White')
plt.imshow(img.squeeze(), cmap='gray')
plt.axis('off')

# Colorized image
plt.subplot(1, 2, 2)
plt.title('Colorized')
plt.imshow(rgb_image)
plt.axis('off')

plt.show()
```

Chapter 5

Results and Disscussion

The ColorRevive project results show that the model can effectively colourize greyscale images. The qualitative results demonstrate realistic and brilliant colour restorations for a wide range of test photos, including natural landscapes and human portraits. For example, in rural situations, flora appears green, skies are blue, and skin tones appear realistic, demonstrating the model's contextual grasp of colours.

Figure 5.1: Demonstration of ColorRevive using an image of a Tree

The loss graph provides a quantitative measure of the training process, showing a steady decline in loss values over 400 epochs reaching as low as 0.0011. This indicates that the model effectively learned to minimize errors, with the loss stabilizing at a low value by the end of training. The smooth trend reflects appropriate training configurations, such as the learning rate and batch size, contributing to the model's stability and convergence.

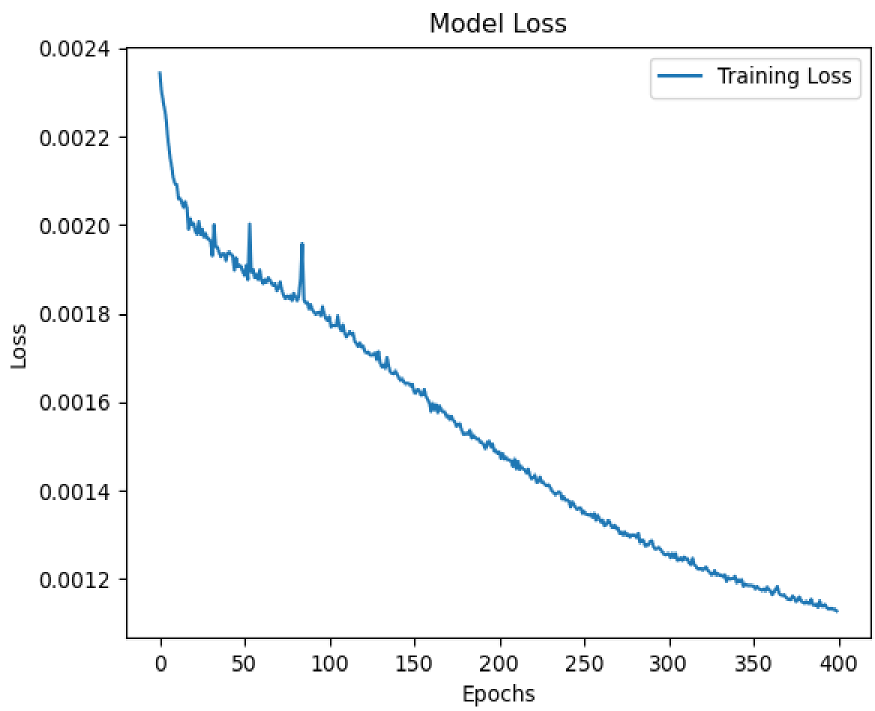

Figure 5.2: Loss Graph obtained during the training phase

Despite the success, some limitations were observed, such as challenges in handling complex textures or ambiguous color contexts. These outputs, though visually appealing, could benefit from further fine-tuning or larger and more diverse datasets. The use of an RTX 4060 GPU expedited training, highlighting the role of hardware in deep learning tasks. Overall, the project successfully demonstrates automated colorization and lays the groundwork for future improvements.

Figure 5.3: Demonstration of ColorRevive using an image of a Tree

Chapter 6

Conclusion and Future work

Finally, the ColorRevive project demonstrates the actual application of deep learning algorithms to the demanding job of automatic image colourization. Using a convolutional neural network (CNN), the model effectively restored vivid and contextually correct colours to greyscale images. The training process resulted in a consistent reduction in loss values, demonstrating the design and parameters' efficiency in learning complex mappings between greyscale and colour domains. The results were especially impressive for natural landscapes and human faces, illustrating the model's capacity to generalise across image types. However, complicated textures and unclear colour contexts posed issues, resulting in inaccurate outcomes on occasion. These restrictions indicate potential enhancements for future editions.

Building on this basis, future work could address these issues by using larger and more diverse datasets to capture a broader range of textures and settings. Advanced architectures, such as Generative Adversarial Networks (GANs), may enhance the realism of outputs by producing more visually realistic colourizations. Adding user-guided colourization options would increase the model's versatility by allowing users to influence specific colours for greater personalisation and accuracy. Furthermore, tailoring the system to specific uses, such as recovering historical black-and-white images or improving medical imaging, might greatly increase its impact. Optimising training with distributed computing or leveraging high-performance GPUs may help cut training time and enhance scalability, allowing for the development of more sophisticated models.

References

[1] Joshi, M.R., Nkenyereye, L., Joshi, G.P., Islam, S.R., Abdullah-Al-Wadud, M. and Shrestha, S., 2020. Auto-colorization of historical images using deep convolutional neural networks. Mathematics, 8(12), p.2258.

[2] An, J., Kpeyiton, K.G. and Shi, Q., 2020. Grayscale images colorization with convolutional neural networks. Soft Computing, 24(7), pp.4751-4758.

[3] Limmer, M. and Lensch, H.P., 2016, December. Infrared colorization using deep convolutional neural networks. In 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 61-68). IEEE.

[4] Anwar, S., Tahir, M., Li, C., Mian, A., Khan, F.S. and Muzaffar, A.W., 2025. Image colorization: A survey and dataset. Information Fusion, 114, p.102720.

[5] Huang, S., Jin, X., Jiang, Q. and Liu, L., 2022. Deep learning for image colorization: Current and future prospects. Engineering Applications of Artificial Intelligence, 114, p.105006.

[6] Morimoto, Y., Taguchi, Y. and Naemura, T., 2009. Automatic colorization of grayscale images using multiple images on the web. In SIGGRAPH 2009: Talks (pp. 1-1).

[7] Tejashwini, K., Soumya, M., Kalyani, P., Deepika, G. and Sujatha, C.N., 2024. Colorization of Black and White Image and Video Using CNN. Advances in Computational Intelligence and Its Applications, p.296.

[8] Ambalathankandy, P., Ou, Y., Kaneko, S. and Ikebe, M., 2024. A Psychological Study: Importance of Contrast and Luminance in Color to Grayscale Mapping. arXiv preprint arXiv:2402.04583.

[9] Kumar, D., Asha, S. and Anwar, S.R., 2024, March. Implementing Image Colorization with Generative Adversarial Networks. In 2024 2nd International

Conference on Artificial Intelligence and Machine Learning Applications Theme: Healthcare and Internet of Things (AIMLA) (pp. 1-5). IEEE.

[10] Abbadi, N.K.E. and Razaq, E.S., 2020. Automatic gray images colorization based on lab color space. Indonesian Journal of Electrical Engineering and Computer Science, 18(3), pp.1501-1509.

[11] An, J., Kpeyiton, K.G. and Shi, Q., 2020. Grayscale images colorization with convolutional neural networks. Soft Computing, 24(7), pp.4751-4758.

[12] Poterek, Q., Herrault, P.A., Skupinski, G. and Sheeren, D., 2020. Deep learning for automatic colorization of legacy grayscale aerial photographs. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13, pp.2899-2915.

[13] Abualola, A., Gunawan, T.S., Kartiwi, M., Ambikairajah, E. and Habaebi, M.H., 2021. Development of Colorization of Grayscale Images Using CNN-SVM. In Advances in Robotics, Automation and Data Analytics: Selected Papers from iCITES 2020 (pp. 50-58). Springer International Publishing.

[14] Abbadi, N.K.E. and Razaq, E.S., 2020. Automatic gray images colorization based on lab color space. Indonesian Journal of Electrical Engineering and Computer Science, 18(3), pp.1501-1509.

[15] Qazzaz, A.A.M., 2024. An Overview of the Most Important Methods for Coloring Grayscale Images. Al-Furat Journal of Innovations in Electronics and Computer Engineering.