TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

PROJECT NO.: THA079MSISE012

ETHNICITY-AWARE AUTO-COLORIZATION OF GRAYSCALE HUMAN PORTRAITS USING CNN

BY

RUPESH POUDEL

A PROJECT

SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATICS AND INTELLIGENT SYSTEMS ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

KATHMANDU, NEPAL

AUGUST, 2024

Ethnicity-Aware Auto-colorization of Grayscale Human Portraits Using CNN

by

Rupesh Poudel

THA079MSISE012

Project Supervisor

Er. Umesh Kanta Ghimire

A project report submitted in partial fulfillment of the requirements for the degree of

Master of Science in Informatics and Intelligent Systems Engineering

Department of Electronics and Computer Engineering

Institute of Engineering, Thapathali Campus

Tribhuvan University

Kathmandu, Nepal

August, 2024

ACKNOWLEDGMENT

This project would not have been possible without the support and guidance of several individuals who significantly contributed to its preparation and completion.

First and foremost, I want to express my deep gratitude to my supervisor, **Er. Umesh Kanta Ghimire**, for his invaluable guidance, insightful feedback, thoughtful suggestions, and constant encouragement. I am also sincerely thankful to my M.Sc. coordinator, **Er. Dinesh Baniya Kshatri**, for his exceptional coordination of the project work, constructive criticism, and patience.

I extend my heartfelt thanks to my friend, **Mr. Bibek K.C.**, for generously allowing me to use his GPU whenever needed throughout the project.

Additionally, I am grateful to my classmates and friends for their advice and moral support. Lastly, I owe special thanks to my parents for their unconditional support and unwavering belief in me.

Rupesh Poudel
THA079MSISE012
August, 2024

ABSTRACT

This project aims to introduces an novel approach to image colorization by integrating ethnicity-aware techniques into a convolutional neural network (CNN). Traditional methods often fail to accurately render diverse human features, leading to generic and inaccurate colorizations. This approach utilizes a dual-CNN architecture, with one network dedicated to detecting ethnicity from facial features and another for colorization. The ethnicity data informs the colorization process, enabling adjustments to color palettes that reflect ethnically accurate skin tones, eye, hair and beard color. And also consistently colorizing visible neck and hand regions to ensure a coherent and realistic appearance. By using CNN, the project uses LAB color space to predict the chrominance component (A and B) from the luminance (L) extracted from the grayscale image and the ethnicity information provided by the Ethnicity detection model. The project includes a balanced dataset comprising major ethnic groups Black, Asian, Indian, White, Middle Eastern and Latino Hispanic each represented uniformly to ensure comprehensive learning across different demographics.

Keywords: *Colorization, Convolution Neural Networks, Ethnicity Detection, LAB Color Space*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AUC	Area Under the Curve
CBAM	Convolutional Block Attention Module
cGAN	Conditional Generative Adversarial Network
CMYK	Cyan, Magenta, Yellow, and Black
CNN	Convolutional Neural Network
CSV	Comma Separated Value
FERET	Facial Recognition Technology
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
HSL	Hue, Saturation, and Lightness
HSV	Hue, Saturation, and Value
LAB	Luminance, a-dimension, and b-dimension
LPIPS	Learned Perceptual Image Patch Similarity
ML	Machine Learning
MORPH	Morphological Pattern Recognition
MSE	Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
PR	Precision Recall
ReLU	Rectified Linear Unit
RGB	Red, Green, and Blue
RMSE	Root Mean Squared Error
ROC	Receiver Operating Characteristic
SOTA	State of the Art
SSIM	Structural Similarity Index Measure
Tanh	Hyperbolic Tangent
VGG	Visual Geometry Group

1 INTRODUCTION

1.1 Background

The Convolutional Neural Network (CNN) has brought new possibilities in the field of auto-colorization of grayscale images. Traditionally, the colorization process was a manual task, both labor-intensive and highly subjective. However, with the advent of machine learning and CNNs, it is now possible to handle and learn complex image patterns, offering a robust solution for automating this task. This project introduces ethnicity-aware mechanisms into the colorization process, ensuring that restored images not only gain color but also reflect ethnically accurate representations. By leveraging CNNs, the project enhances the precision and cultural relevance of colorization, with promising applications in historical photo restoration, film colorization, personalized media content creation, and even educational tools, significantly contributing to the advancement of image processing technologies.

1.2 Motivation

The primary motivation for this project arises from the desire to enhance the viewing experience of old black and white images, transforming them into more vibrant and informative contents. The motivation behind this project stems from the need for culturally sensitive and realistic colorization of grayscale images, which is often neglected in traditional methods. Moreover, the advancements in machine learning, especially with CNNs, it can effectively learn color application from vast datasets, predicting and assigning complex color dynamics that manual processes cannot achieve consistently or at scale. Thus, the project is driven by a dual motivation: on one hand, the cultural importance of making historical visual content more accessible, authentic, and engaging for contemporary audiences; on the other hand, the technological opportunity to push the boundaries of what automated systems can achieve in the field of image processing. By integrating these motivations, the project aims to set a new standard for the colorization of grayscale images, contributing both to the preservation of cultural heritage and the advancement of image processing technologies.

1.3 Problem Statement

The majority of historical and archival visual content exists in grayscale. These types of media provide a valuable importance into the past but the lack of color can make these

images less engaging and less interactive for the modern viewers who are used to with the watching and seeing colorful contents. Moreover, current colorization techniques, especially manual ones, are time-consuming, often inconsistent, and lack the ability to scale to the vast quantities of old images available.

The challenge, therefore, lies in developing a robust and scalable system capable of accurately identifying ethnic characteristics from grayscale images and applying a corresponding color palette that reflects these characteristics. This approach not only enhances the visual appeal of the images but also significantly boosts their authenticity, making them more educational and ethnically relevant.

Therefore, this project aims to develop a robust system for efficiently colorizing grayscale images to color. This aims to enhance the visual quality and appeal of the historical media which makes it more accessible and enjoyable for new audiences.

1.4 Project Objectives

- **Develop a CNN-based Auto-colorization System:** To develop a CNN to predict realistic colors for grayscale images, trained on diverse human portrait datasets with ethnic labels.
- **Enhancing the visual appeal and accessibility of historical media:** Integrate a colorization algorithm using ethnic information to apply culturally accurate colors, ensuring consistent colorization of faces, necks, and hands.

1.5 Scope of Project

The ethnicity-aware auto-colorization system successfully enhanced the quality and cultural accuracy of colorized grayscale images. By leveraging Convolutional Neural Networks (CNNs) for ethnicity detection, the system accurately identified and differentiated ethnic features in human portraits. This information was effectively utilized to apply appropriate and realistic colorization to facial features, necks, and hands, ensuring that each individual's appearance was represented with a high degree of cultural sensitivity. The system demonstrated significant potential in applications such as historical photo restoration, film colorization, and personalized media content creation, where maintaining the authenticity of diverse ethnic appearances is crucial.

Throughout the project, it became evident that the accuracy of ethnicity detection and colorization heavily relied on the quality and diversity of the training dataset. Instances where the dataset lacked sufficient representation of certain ethnic groups highlighted the model's limitations, leading to biased or less effective colorization for those groups. Additionally, while the model was designed to handle images featuring multiple individuals of different ethnicities, challenges arose in such scenarios, necessitating further fine-tuning to ensure consistent results across all ethnicities. The project also confirmed that computational resources and processing time were considerable, particularly when dealing with large datasets, underscoring the need for optimization in future iterations.

1.6 Potential Project Applications

- **Historical Media Restoration:** This project can transform historical media by restoring color to old photographs, making them more engaging for modern audiences.

- **Entertainment and Media:** The system can be utilized in educational materials to provide more realistic and culturally accurate visual representations. For example, in history textbooks or educational documentaries, accurately colorized historical photos can provide students with a more engaging and authentic learning experience.

- **Educational Enhancement:** Educators can use colorized historical visuals as effective teaching tools which makes history and cultural studies more engaging and relatable for students.

- **Personal and Creative Use:** Individuals interested in genealogy can revive their old family photos to enhance their connection with past members.

- **Forensic and Investigative Applications:** Assisting law enforcement and forensic experts in enhancing grayscale surveillance footage and crime scene photos.

- **Cultural Preservation:** Museums and cultural organizations can use the technology to restore and preserve important historical artifacts, providing the public with a more vivid connection to history.

1.7 Originality of Project

This project brings several original contributions to the field of image colorization by addressing the following research gaps:

- Introducing a novel approach that incorporates ethnicity detection into the autocolorization process, ensuring that the colorization of grayscale images accurately reflects the diverse ethnic backgrounds of individuals.
- Expanding the traditional scope of image colorization by including not only facial skin color but also considering facial features like eyes, hairs, beards and consistently colorizing visible neck and hand region with the inclusion of diverse datasets.

1.8 Organisation of Project Report

The report for this project is structured into six chapters, each building upon the previous to create a clear and cohesive narrative. After this introductory chapter, which outlines the problem and motivation for the project, Chapter 2 presents a literature review. This chapter examines key publications in the field, identifying a significant research gap that the project aims to address. The review sets the foundation for the project by contextualizing it within existing research. Chapter 3 details the methodology used for implementing the project. This chapter explains the tools, techniques, and processes chosen, along with the reasoning behind these choices. It covers data collection, model development, and the implementation process, ensuring the reader understands the step-by-step approach taken. Chapter 4 provides an overview of the results obtained from the project. This chapter presents the findings, aligning them with relevant research and highlighting their significance. The results are discussed in the context of the project's objectives and the broader field of study. Chapter 5 offers a comprehensive discussion and analysis of the results. This chapter explores the implications of the findings, comparing them with similar studies, and discussing the strengths and limitations of the methodology. It reflects on how well the project addressed the research problem and any unexpected outcomes. Chapter 6 discusses Future Enhancements, outlining potential areas for further development and suggesting new research avenues. This chapter considers how the project could be expanded or refined and its practical applications in

different contexts. Finally, the report concludes with Chapter 7, summarizing the key findings and providing a conclusion. This chapter revisits the research problem, reflects on the project's contributions, and highlights its significance and potential impact on the field.

2 LITERATURE REVIEW

2.1 Ethnicity Classification

"Classification of Ethnicity Using Efficient CNN-Models on MORPH and FERET Datasets Based on Face-Biometrics" (Abdulwahid Al Abdulwahid, 2023): The paper proposes the use of efficient convolutional neural network (CNN) models for classifying ethnicity based on facial biometrics. The authors developed two CNN models, Model A and Model B, and evaluated their performance on the MORPH and FERET datasets. Model A focused on gender classification and achieved an accuracy of 85% in classifying individuals into four ethnic groups. Model B was designed for ethnicity classification and reached an accuracy of 86% on the same task. The models were trained and tested using a holdout approach, and the authors highlighted that the results were obtained by focusing only on the central region of the face, which saved time and resources. The key strengths inclides the use of efficient CNN models for ethnicity classification, which is an important but understudied area in facial biometrics. And the evaluation of the models on two publicly available datasets, MORPH and FERET, which allowed for a comprehensive assessment of the approach. The focus on the central region of the face, which demonstrated the potential for accurate classification without the need for processing the entire face. The main weakness of the paper is the lack of a detailed analysis of the model architectures and the feature extraction techniques employed. Additionally, the paper does not provide a comparative analysis with state-of-the-art methods beyond a few references, which would have strengthened the positioning of the proposed approach. [1]

2.2 Technologies for Image Colorization

"Human Face ImageColorization with Dual-Scale Attention U-Net", (Ben Want et al., 2022): The paper proposes a dual-scale attention U-Net architecture for colorizing grayscale human face images. The key contributions includes Dual-scale convolution module where the authors use two different sized convolution kernels (3x3 and 7x7) in the U-Net backbone to extract features at multiple scales, improving the ability to capture details. CBAM attention module where the authors integrate the Convolutional Block Attention Module (CBAM) in the skip connections to help the network focus on salient regions and suppress unnecessary areas, reducing boundary leakage and detail

loss. And MS-SSIM-L1 loss function inspired by image super-resolution, the authors use a combination of MS-SSIM and L1 loss to better capture texture and edge information during training. The quantitative results show the proposed method outperforms previous deep learning-based colorization approaches like Pix2Pix and Zhang et al. in terms of PSNR and SSIM. Qualitatively, the method produces more realistic and detailed colorized human face images, especially for challenging old historical photos. The key strengths of the paper are the effective dual-scale and attention mechanisms, as well as the novel loss function tailored for face image colorization. However, the method still struggles with consistently coloring the background regions, and the training time is relatively long. Overall, the paper presents a well-designed deep learning approach for high-quality grayscale human face image colorization, demonstrating the benefits of multi-scale feature extraction and spatial-channel attention mechanisms.[2]

"Automatic Gray-Image Coloring Method Based on Convolutional Network". (Jiayi Fan et al., 2022): The paper proposes an automatic gray image coloring method based on convolutional neural networks (CNNs). The authors divide the coloring process into two parts - coloring the foreground based on a reference image, and coloring the background based on prior knowledge. For the foreground coloring, the method extracts semantic information from the grayscale image using a CNN model, and then transfers the color from the designated area of the reference image to the corresponding area of the grayscale image. This is done by iteratively optimizing a random noise map to match the content features of the grayscale image and the style features of the reference image. For the background coloring, the authors leverage prior knowledge to fill in the background regions. Finally, the foreground and background colorings are combined using a Poisson fusion technique. The experimental results show that the this method achieves good coloring effects, with advantages in network size and coloring quality compared to other deep learning based methods like VGG and differential networks. The qualitative comparisons demonstrate that the CNN-based coloring produces more vivid and realistic results, with better handling of details and lighting/shading effects. The main strengths of this work are the hybrid foreground-background coloring approach leveraging semantic information and reference images, as well as the effective CNN-based implementation. The weakness is that the method still relies on manual selection of reference images, and the fusion between foreground and background could be further improved. Overall, the

paper presents a promising CNN-based automatic gray image coloring technique with solid experimental validation. [3]

"Fine-grained semantic ethnic costume high-resolution image-colorization with conditional GAN.", (Di Wu et al., 2021): The paper provides a method for high-resolution image colorization of ethnic costumes, employing a fine-grained semantic approach with a conditional generative adversarial network (cGAN). Talking about the architecture authors use Pix2PixHD as the backbone network and modify it to incorporate fine-grained semantic information of different regions of the ethnic costumes as input conditions to the generator. The discriminator also takes the fine-grained semantic masks as input along with the generated and real images. The authors compare their method against several state-of-the-art colorization algorithms using the PSNR and SSIM metrics. Their method outperforms the compared approaches, achieving a PSNR of 26.45 and SSIM of 0.914. Visually, the colorization results from the proposed method show superior performance in preserving the local details and color distribution of the complex ethnic costumes compared to other methods. Talking about the strength, it uses fine-grained semantic information to guide the colorization process for ethnic costumes with rich and diverse color features. And also includes some weakness that, the method relies on manually annotated fine-grained semantic masks, which can be labor-intensive to obtain. Exploring automatic semantic segmentation could help address this. The experiments are limited to a custom dataset of four Chinese ethnic minority costumes. Evaluating the generalization to a wider range of ethnic costumes would be beneficial. Overall, the paper presents a novel, effective approach for high-quality colorization of ethnic costume images by incorporating fine-grained semantic guidance, demonstrating the importance of semantic information for this challenging task.[4]

"Colorization of B/W images using deep neural networks" (David Futschik, 2018): The authors explore the task of fully automatic colorization of grayscale cartoon images using deep convolutional neural networks (CNNs). They propose and compare two different CNN architectures: a plain CNN model and a residual CNN model inspired by ResNet. The authors train various variants of these models with different loss functions (KL divergence, L2) and regularization techniques (dropout, rebalancing). Quantitatively, the plain CNN with L2 loss performs best on metrics like RMSE and PSNR, but produces

desaturated colors. The KL divergence loss with rebalancing helps predict more vibrant colors. Qualitatively, the plain CNN handles larger objects and backgrounds better, while the residual CNN performs well on smaller objects. However, both architectures struggle with consistent colorization across video sequences. The authors propose post-processing steps like segmentation with flood-fill and ensemble averaging to improve the qualitative results. Overall, the method shows promise in plausibly colorizing individual cartoon images, but temporal consistency across video frames remains a key challenge. The unique properties of the cartoon dataset, like lack of textures and large uniform regions, make it a difficult test case compared to natural images.[5]

"Colorful Image Colorization" (Richard Zhang et al., 2016): This paper proposes a method for automatically colorizing grayscale photographs using a convolutional neural network (CNN). For the architechure it use a feed-forward CNN with a VGG-style architecture, without any pooling layers. They use dilated convolutions to maintain spatial resolution. Rather than using a standard regression loss, the authors treat colorization as a classification problem, predicting a distribution over quantized ab color values. They also use class rebalancing to capture the diversity of colors. At test time, the authors use an "annealed-mean" operation to combine the predicted color distribution into a final colorized output. Author evaluate their method using several metrics. On a perceptual "colorization Turing test", their method fools human participants 32% of the time, significantly better than previous work. They also show strong performance on using the colorized images for object classification, outperforming previous self-supervised feature learning approaches. The paper showcases many compelling examples of the method producing realistic and vibrant colorizations, even on challenging scenes. However, it also highlights failure cases where the method struggles with consistent long-range predictions or confuses certain color semantics.
The strength is the ability to produce high-quality, diverse colorizations with limited user interaction. A weakness is that the method still struggles in some situations, and further work may be needed to make it fully robust. But the paper represents an important step forward in this problem.[6]

"Let there be Color!" (Satoshi Iizuka et al., 2016): The authors present a novel deep learning approach for automatic colorization of grayscale images. Their method

combines both global and local image features using a Convolutional Neural Network architecture. The global features capture high-level semantic information about the image, while the local features capture low-level details. These features are fused using a novel ”fusion layer” and then processed by a colorization network to predict the final chrominance of the image. Quantitatively, the authors show that their approach significantly outperforms a strong baseline CNN model as well as the state-of-the-art method, with user studies indicating that 92.6% of the colorized images are considered ”natural” compared to only 70% for the baseline. Qualitatively, the authors demonstrate impressive colorization results on a diverse set of images, including historical black-and-white photographs from over a century ago. The key strengths of this work are the innovative fusion of global and local features, the ability to leverage large-scale classification datasets to learn better global priors, and the end-to-end trainable architecture that can handle images of arbitrary resolution. The main weakness is the inherent ambiguity in colorization, where there may not be a single correct solution, leading to some failure cases where the model does not capture the full semantic context. Overall, this is a highly impactful work that advances the state-of-the-art in automated image colorization using deep learning.[7]

2.3 Identified Research Gap

The identified research gap in the field of human portrait colorization lies in accurately rendering skin tones and features like eye color across different ethnic groups, and consistently coloring the visible areas like the neck and hands. Current technologies often employ a uniform approach to colorization, which fails to account for the varied skin tones and facial features across different ethnic groups, leading to potential misrepresentations and a loss of historical authenticity. This gap highlights the critical need for an advanced colorization system that integrates ethnicity detection, allowing for color adjustments that respect and accurately represent the ethnic background and historical contexts of the subjects in the image.

3 METHODOLOGY

3.1 Theoretical Formulations

3.1.1 Convolution Neural Network

A Convolutional Neural Network (CNN) is a type of artificial neural network designed primarily for processing and analyzing images for feature extraction. CNNs have become the cornerstone of computer vision tasks and have shown remarkable success in tasks like image classification, object detection, facial recognition, and more.
A typical CNN is composed of three main layers: the convolutional layer, the pooling layer, and the fully connected layer.

Convolutional Layers: Convolution is the core operation in CNNs. Convolutional layers use filters (also known as kernels) to scan the input image. These filters are small, learnable matrices that slide over the input image to extract various features. The convolution operation applies element-wise multiplication and summation, effectively capturing patterns in different regions of the image. Multiple filters are used in each convolutional layer to capture different features, such as edges, textures, and shapes.
The gray scale image is represented as 2 dimensional matrix of pixels in which each element contains the pixels intensity. Similarly, for the color image there are three such arrays or channels namely Red, Green and Blue. The output of the convolution operation is given by the formula:

$$\text{Output size} = \frac{N - f + 2p}{S} + 1 \times \frac{N - f + 2p}{S} + 1 \tag{3.1}$$

When padding is not applied and the stride is set to one, the output feature map is smaller in size compared to the input. If the input is padded with the zero then the output of the feature map is same as the input size. For example, the stride is two that means the output of the is just half as the input size and the padding is adjusted in such a way the output of the feature map is same as the input by using the keyword 'same'.

Pooling Layers: Pooling layers down sample the spatial dimensions of the feature maps while retaining their essential information. Max-pooling is a common pooling technique that retains the maximum value within a local region of the feature map and discards

the rest. This reduces computational complexity and retains important features. And another pooling technique is average pooling which takes the average value from a patch of feature map.

Fully Connected Layers (Dense Layers): After several convolutional and pooling layers, the network often ends with one or more fully connected layers. Fully connected layers flatten the feature maps into a 1D vector and apply linear transformations and activation functions. The output of the final fully connected layer is used for classification, regression, or other specific tasks.

3.1.2 Upsampling:

Upsampling, also known as upsizing or interpolation, is a digital image processing technique used to increase the size or resolution of an image. This process involves creating new data points to fill in the gaps between existing data points, effectively making the image larger while preserving its visual quality to some extent. While upsampling can increase the size of an image, it cannot magically create new information or detail that was not present in the original image. The quality of the upscaled image depends on the interpolation method used and the degree of upscaling, and there may be some loss of sharpness and clarity, especially when significantly increasing the image size.

3.1.3 Activation Functions:

Activation functions are a critical component in neural networks, including CNN, as they introduce non-linearity into the model which enables it to learn and represents complex pattern in the data. At each learn it applies non-linear transformations to captures complex patterns.

ReLU (Rectified Linear Unit) is a popular activation function used in neural networks, especially in convolutional neural networks (CNNs). It outputs the input directly if it is positive; otherwise, it outputs zero. This non-linearity enables the network to learn complex patterns and functions that a linear model cannot capture. ReLU is computationally efficient and avoids the vanishing gradient problem, which is common with activation functions like sigmoid and tanh. This makes it suitable for training deep networks as it allows gradients to propagate effectively.

Tanh (hyperbolic tangent) activation function is another popular activation function which output value ranges in -1 and 1. Tanh is zero-centered, meaning its output is centered around zero, which can lead to faster convergence during training as the mean activation is closer to zero. The tanh activation function is used in the final convolutional layer before the output layer. Since the task involves predicting the color channels (A and B channels of the LAB color space) normalized between -1 and 1, tanh is appropriate because it naturally outputs values in this range.

3.1.4 Regularization

Regularization is a technique used to improve the generalization of machine learning models by preventing overfitting. Overfitting occurs when a model learns the noise in the training data rather than the actual underlying patterns, resulting in poor performance on unseen data. Regularization techniques add constraints or penalties to the learning process to ensure that the model does not become too complex. Common techniques includes:

Dropout: During training, randomly sets a fraction of input units to zero at each update. This helps prevent units from becoming overly dependent on each other and promotes the network's ability to learn more resilient features. The dropout rate (e.g., 0.5) is a hyperparameter that specifies the probability of dropping a unit.

Early Stopping: Early stopping monitors the model's performance on a validation set and stops training when performance starts to deteriorate. This prevents overfitting by stopping the training process before the model starts to memorize the training data.The objective is to train the model until it achieves the minimum possible training error before the validation error begins to plateau or increase.

Data Augmentation: Data augmentation is a regularization technique that modifies model training data. It expands the size of the training set by artificially increasing the size of the training set by creating modified versions of the training data. This can include rotations, translations, scaling, and flipping of images, which helps the model generalize better by learning from a more diverse dataset.

3.1.5 Optimizer

An optimizer adjusts the weights and biases of the network to minimize the loss function, directly influencing how quickly and accurately the model learns. In our auto colorization model, you have selected the RMSProp (Root Mean Square Propagation) optimizer. RMSProp is particularly well-suited for this task because it adapts the learning rate for each parameter individually based on the average of recent magnitudes of the gradients. This adaptive nature helps stabilize the learning process and accelerates convergence, making RMSProp effective for handling non-stationary objectives commonly encountered in complex tasks like image colorization. RMSProp is efficient in such scenarios because it computes adaptive learning rates, thus preventing issues related to overly aggressive updates that could destabilize training. Additionally, RMSProp is computationally efficient and requires minimal memory, making it suitable for deep learning models handling large datasets, as is the case with your colorization model.

3.1.6 Color Space

Color spaces are a way to organize colors in a coordinate system, typically defined by three or more dimensions, which represent various components of color. Different color spaces are used depending on the application, such as RGB for display screens, CMYK for printing, and HSV or HSL for color picking in graphic design. Each color space serves a specific purpose and offers unique advantages for certain types of image processing tasks.

LAB Color Space: The LAB color space is particularly significant for image processing tasks like colorization because it separates luminance from color information, making it highly useful for applications where color manipulation is required independent of lighting.

Components of LAB Color Space:

L Component (Luminance): Represents lightness, ranging from 0 (black) to 100 (white). It corresponds to the brightness of the color, independent of any actual color information.

A Component: Represents the position between magenta and green, with negative values indicating green and positive values indicating magenta.

B Component: Represents the position between yellow and blue, with negative values indicating blue and positive values indicating yellow.

Releavance to this proposed project Since the L component is directly derived from the grayscale image, it remains unaltered during the colorization process. This ensures that the lightness or darkness of the original image is maintained, which is crucial for preserving the original image's details and textures. The task of the CNN in the colorization process primarily involves predicting the A and B components. By training the model to only focus on these chrominance components, the colorization process becomes more efficient and can be more finely tuned, as it does not need to worry about affecting the lightness of the image.

3.2 Mathematical Modelling

3.2.1 RGB to LAB Conversion

The conversion from RGB (Red, Green, Blue) to LAB (Luminance, a-dimension, b-dimension) involves several transformations. LAB color space is designed to approximate human vision and is more perceptually uniform than RGB. This means that an equal numerical change in these values results in a roughly equivalent change in how it's visually perceived.[8]

Step 1: Linearize RGB Values

RGB values are typically stored in a compressed format to optimize space and typically range from 0 to 255. These values need to be linearized and scaled to a range from 0 to 1. This is necessary to correct for gamma compression applied during the storage of these values:

$$R' = \frac{R}{255}, \qquad G' = \frac{G}{255}, \qquad B' = \frac{B}{255} \tag{3.2}$$

Step 2: Convert RGB to XYZ

The linearized RGB values are transformed to the XYZ color space using a specific transformation matrix. The matrix coefficients are derived based on the standard observer

and the illuminant conditions (normally D65 illuminant is used).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124564 & 0.3575761 & 0.1804375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9503041 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{3.3}$$

Step 3: Scale XYZ to Reference White

To make the XYZ values relative to the white point (typically D65), they are scaled accordingly. This adjustment is necessary for correct color interpretation under different light sources:

$$X = \frac{X}{X_{\text{reference}}}, \qquad Y = \frac{Y}{Y_{\text{reference}}}, \qquad Z = \frac{Z}{Z_{\text{reference}}} \tag{3.4}$$

Step 4: Convert XYZ to LAB

Finally, the scaled XYZ values are converted to LAB. The transformation involves a non-linear adjustment which better correlates with human vision:

$$L = 116 \cdot f\left(\frac{Y}{Y_0}\right) - 16 \tag{3.5}$$

$$a = 500 \cdot \left(f\left(\frac{X}{X_0}\right) - f\left(\frac{Y}{Y_0}\right)\right) \tag{3.6}$$

$$b = 200 \cdot \left(f\left(\frac{Y}{Y_0}\right) - f\left(\frac{Z}{Z_0}\right)\right) \tag{3.7}$$

Where $f(t)$ is defined as a piecewise function to transform XYZ values:

$$f(t) = \begin{cases} t^{1/3} & \text{if } t > \left(\frac{6}{29}\right)^3 \\ 7.787 \cdot t + \frac{4}{29o} & \text{otherwise} \end{cases} \tag{3.8}$$

This conversion of RGB to LAB color space creates the enless possibilities in image processing and computer vision. [8]

3.2.2 Data Augmentation

Data augmentation enhances the diversity of training data by applying random transformations to original images. Here, we discuss the mathematical modeling of common data augmentation techniques.

3.2.2.1 Photometric Augmentation

Brightness Adjustment

Brightness adjustment is a photometric augmentation technique used to modify the brightness of an image. This is achieved by adding or subtracting a constant value to all pixel intensities in the image. The purpose of this adjustment is to make the model robust to varying lighting conditions.

The mathematical formula for brightness adjustment is given by:

$$I' = I + \beta \tag{3.9}$$

where:

- I is the original image.
- I' is the brightness-adjusted image.
- β is the brightness factor, which can be a positive or negative value.

The value of β determines the degree of brightness adjustment:

- A positive β increases the brightness, making the image appear lighter.
- A negative β decreases the brightness, making the image appear darker.

The brightness factor β is typically sampled from a specified range, for example, $[-0.5, 0.5]$. In this range:

- $\beta = 0.5$ would result in the maximum increase in brightness.
- $\beta = -0.5$ would result in the maximum decrease in brightness.

It is important to ensure that the pixel values remain within the valid range, typically $[0, 255]$ for 8-bit images, to avoid overflow or underflow. This clipping ensures that all pixel values are valid after the brightness adjustment.

For color images, the brightness adjustment is usually applied equally across all color channels (R, G, B) to maintain the color balance.

This augmentation technique enhances the model's ability to generalize to images captured under different lighting conditions, thus improving the model's robustness and performance in diverse real-world scenarios.

3.2.2.2 Geometric Augmentation

1. Rotation

Rotation is a transformation that rotates an image by a specified angle θ. The rotation matrix for a 2D image around the origin is:

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{3.10}$$

Given an image I and a rotation angle θ, the rotated image I_{rot} can be computed using bilinear interpolation:

$$I_{\text{rot}}(x', y') = \sum_{x,y} I(x, y) \cdot \text{kernel}(x' - x, y' - y) \tag{3.11}$$

where:

- (x, y) represents the coordinates of the original image.
- (x', y') are the coordinates in the rotated image.

- kernel represents the bilinear interpolation kernel.

2. Translation

Translation shifts an image by a specified distance along the x and y axes. Given a translation vector (t_x, t_y), the translated image I_{trans} is computed as:

$$I_{\text{trans}}(x, y) = I(x - t_x, y - t_y) \tag{3.12}$$

where:

- (t_x, t_y) is the translation vector.
- $I(x, y)$ are the coordinates of the original image.

3. Scaling

Scaling resizes an image by multiplying its coordinates by a scaling factor s. The scaled image I_{scale} is given by:

$$I_{\text{scale}}(x', y') = I\left(\frac{x'}{s}, \frac{y'}{s}\right) \tag{3.13}$$

where:

- s is the scaling factor.
- (x', y') are the coordinates in the scaled image.

4. Shearing

Shearing distorts an image by shifting one axis while keeping the other fixed. For a shearing factor α, the sheared image I_{shear} is computed as:

$$I_{\text{shear}}(x', y') = I(x' - \alpha \cdot y', y') \tag{3.14}$$

where:

- α is the shearing factor.

- (x', y') are the coordinates in the sheared image.

5. Flipping

Flipping mirrors an image either horizontally or vertically. The flipped image I_{flip} is given by:

$$I_{\text{flip}}(x', y') = I(W - x' - 1, y') \tag{3.15}$$

where:

- W is the width of the image.
- (x', y') are the coordinates in the flipped image.

3.2.3 Convolutional Neural Networks (CNNs)

Convolutional Layers

Convolutional layers perform the key operation in CNNs where a kernel (or filter) convolves across the input image. The mathematical operation for convolution is:

$$s(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) \cdot K(i - m, j - n) \tag{3.16}$$

where:

- I represents the input image.
- K is the convolutional kernel.
- $s(i, j)$ is the output feature map at position (i, j).

Activation Functions

Activation functions introduce non-linearities into neural networks, enabling them to learn and model complex patterns in the data. Two commonly used activation functions are the Rectified Linear Unit (ReLU) and the Hyperbolic Tangent (Tanh) functions.

1. ReLU (Rectified Linear Unit)

The ReLU activation function is defined as:

$$f(x) = \max(0, x) \tag{3.17}$$

where:

- x represents the input to the activation function.
- $f(x)$ is the output of the function.

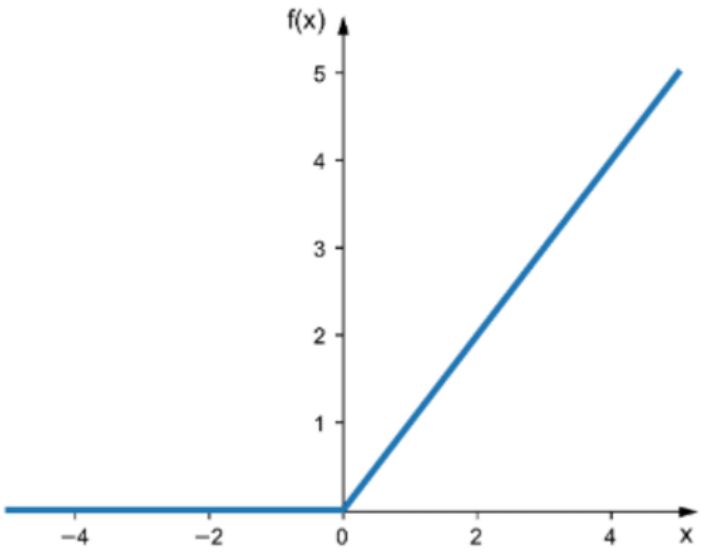


Figure 3.1: ReLU Graph [9]

The ReLU function returns the input value as it is if it's positive; otherwise, it returns zero. This simple non-linear transformation helps to address the vanishing gradient problem by allowing gradients to pass through unaltered when the input is positive. As a result, ReLU is effective in speeding up the convergence of the training process.

2. Tanh (Hyperbolic Tangent)

The Tanh activation function is defined as:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.18}$$

where:

- x represents the input to the activation function.
- $\tanh(x)$ is the output of the function.

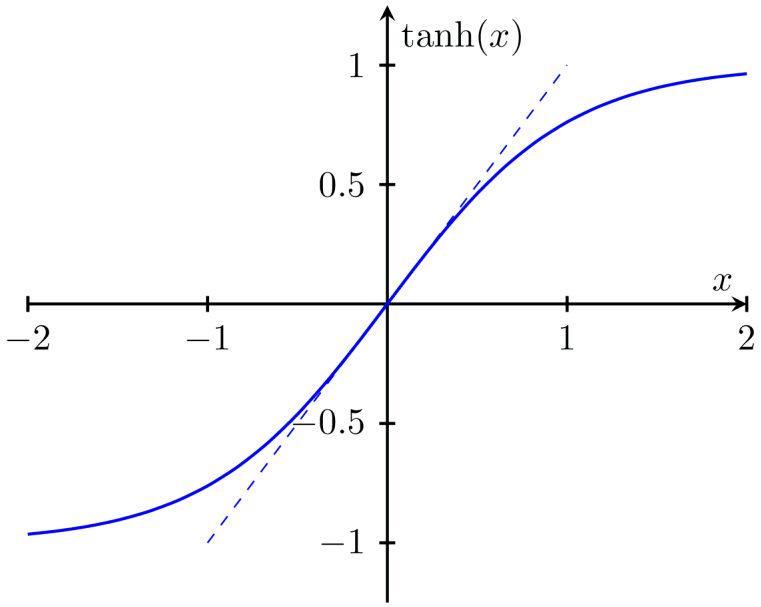


Figure 3.2: Tanh Graph [10]

The Tanh function outputs values in the range $[-1,1]$, which makes it zero-centered. This aids the optimization process by keeping the average of the activations near zero, which enhances training stability. The Tanh function is particularly useful in cases where the input data is normalized and has both positive and negative values.

Pooling Layers Pooling layers reduce the spatial dimensions of the feature map. Max pooling, for example, selects the maximum value within a specified window:

$$p(i,j) = \max_{l \in [i,i+k], m \in [j,j+k]} I(l,m) \tag{3.19}$$

where:

- $p(i,j)$ is the output of the pooling operation.
- $I(l,m)$ is the input at position (l,m).

- k is the size of the pooling window.

Fully Connected Layers Fully connected layers integrate features for final output predictions:

$$y = Wx + b \tag{3.20}$$

where:

- x is the flattened feature vector from previous layers.
- W is the weight matrix.
- b is the bias vector.
- y is the output vector.

Loss Functions Loss functions measure the error between the network's predictions and actual targets. Mean Squared Error (MSE) is commonly used in regression tasks:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.21}$$

where:

- y_i is the true value for the i-th sample.
- $\hat{y}_i$ is the predicted value for the i-th sample.
- N is the number of samples.

This concludes the mathematical modeling of CNN.[11]

3.3 System Block Diagram

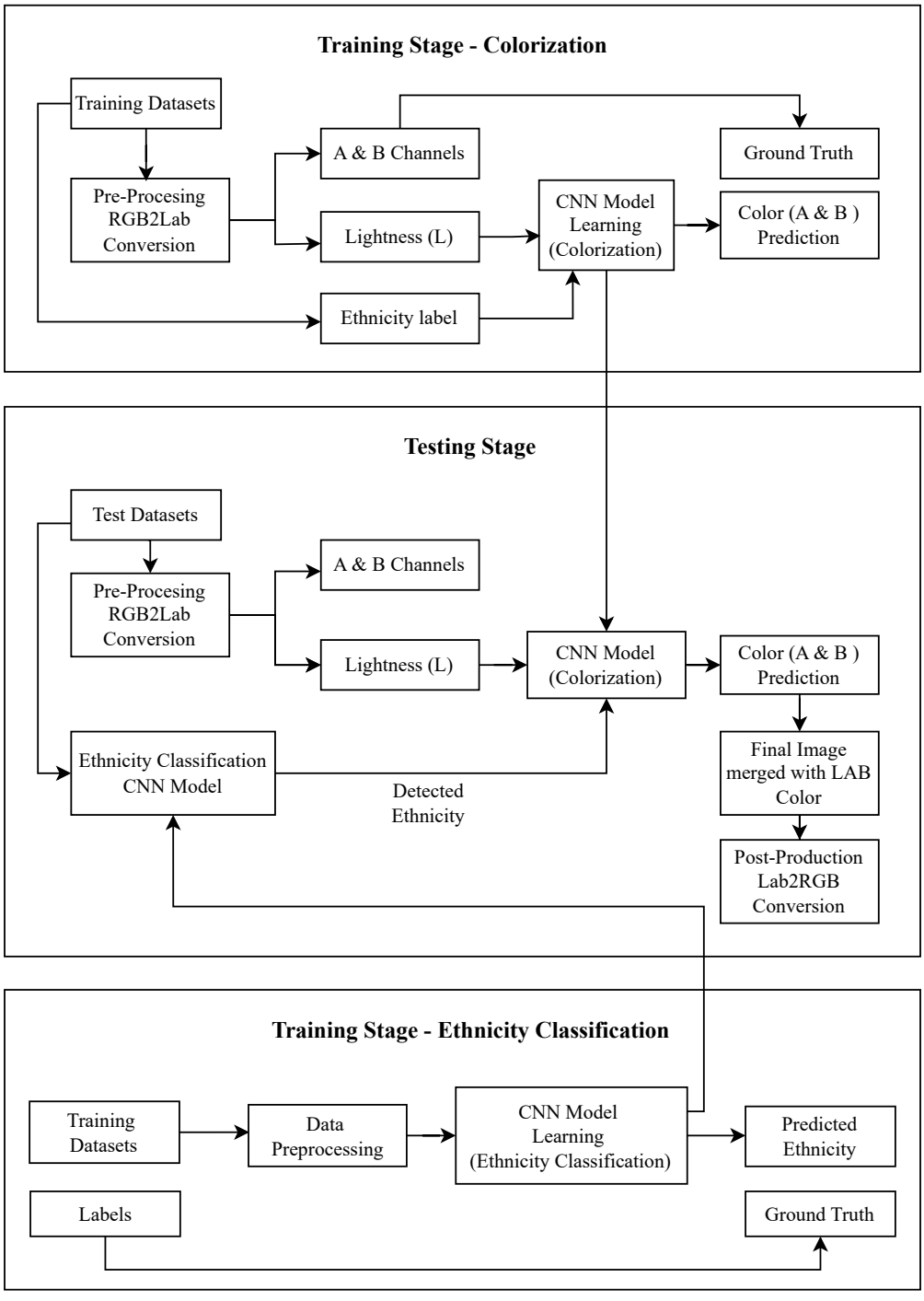


Figure 3.3: System Block Diagram

Figure 3.3 shows the system block diagram of the overall system.

As shown in the figure 3.3 this project outlines the two stages: training and testing.

Training Stage:

Preprocessing: Initially, images in the training dataset are preprocessed by converting them from RGB to Lab color space. This step separates the luminance component (L) from the color components (a and b).

Colorization CNN Model Training: The CNN model has been designed to take the L component as input along with the ethnicity label of the image and predict the a and b components, effectively learning to colorize the grayscale image. The model's architecture facilitates this by including layers that focus on interpreting the luminance information and predicting the corresponding colorization iteratively by comparing with the ground truth values. During training, the Mean Squared Error (MSE) is calculated between the predicted a and b components and the ground truth (actual a and b components from the original image). This error metric guides the optimization process, helping to adjust the model weights to minimize prediction errors.

Ethnicity Classification CNN Model Training: In the training stage of an ethnicity classification model, a diverse dataset of labeled images representing various ethnic groups is first collected and preprocessed to standardize image sizes, formats, and pixel values. Convolutional Neural Network (CNN) optimized for image classification tasks, is then trained using this dataset. Validation is conducted periodically with a separate set of data to monitor and optimize the training process, ensuring the model accurately predicts ethnicity without overfitting to the training set.

Testing Stage:

Ethnicity Classification: During the testing phase the ethnicity of the person on the input image is first detected using the trained Ethnicity Classification Model to predict the ethnicity of the person.

Image Testing: For grayscale images, the testing phase involves using the trained model to predict the a and b components based on the L component extracted from the input image and the predicted ethnicity, ensuring more accurate and culturally appropriate colorization. The predicted color components are then combined with the L component

to reconstruct the colorized image in Lab color space, which is subsequently converted back to RGB for display.

3.4 Instrumentation Requirements

Hardware Requirements:

- **NVIDIA GeForce RTX 4070 Ti:** This GPU was used primarily for training the CNN model on the final large dataset. Its intensive computational power is able to handle the required computational demand. The device type is NVIDIA GeForce RTX 4070 Ti. And this GPU is currently owned by a friend who has agreed to provide access for the duration of the project when requried.
- **MacBook Pro 2020 (Intel Version):** The MacBook Pro is used for initial testing, research, development and lighter computational tasks. It provides a portable and convenient platform for research and coding. Type of device is MacBook Pro 2020 Intel Version. And this device is already owned by me and which will be available throughtout all the stages of the project.

Software Requirements: I am using Juypter Notebook as the primary interface for coding, testing and visualizing machine learning processes. And Tensorflow was utilized for building and training the neural network model and it was paired with CUDA which is a parallel computing platform by NVIDIA which allows acceleration of computational tasks on NVIDIA GPUs.

3.5 Dataset Explanation

The dataset chosen for this project is highly relevant as it includes images of human portraits from various ethnic backgrounds, which is crucial for the ethnicity detection and colorization task. By having a diverse set of images, the model can learn the unique characteristics and color profiles associated with different ethnicities. This diversity ensures that the colorization results are accurate and ethnically appropriate, addressing the project's objective to provide realistic and inclusive colorization for multi-ethnic images. The fig 3.4 shows the glimpse of dataset diversity.

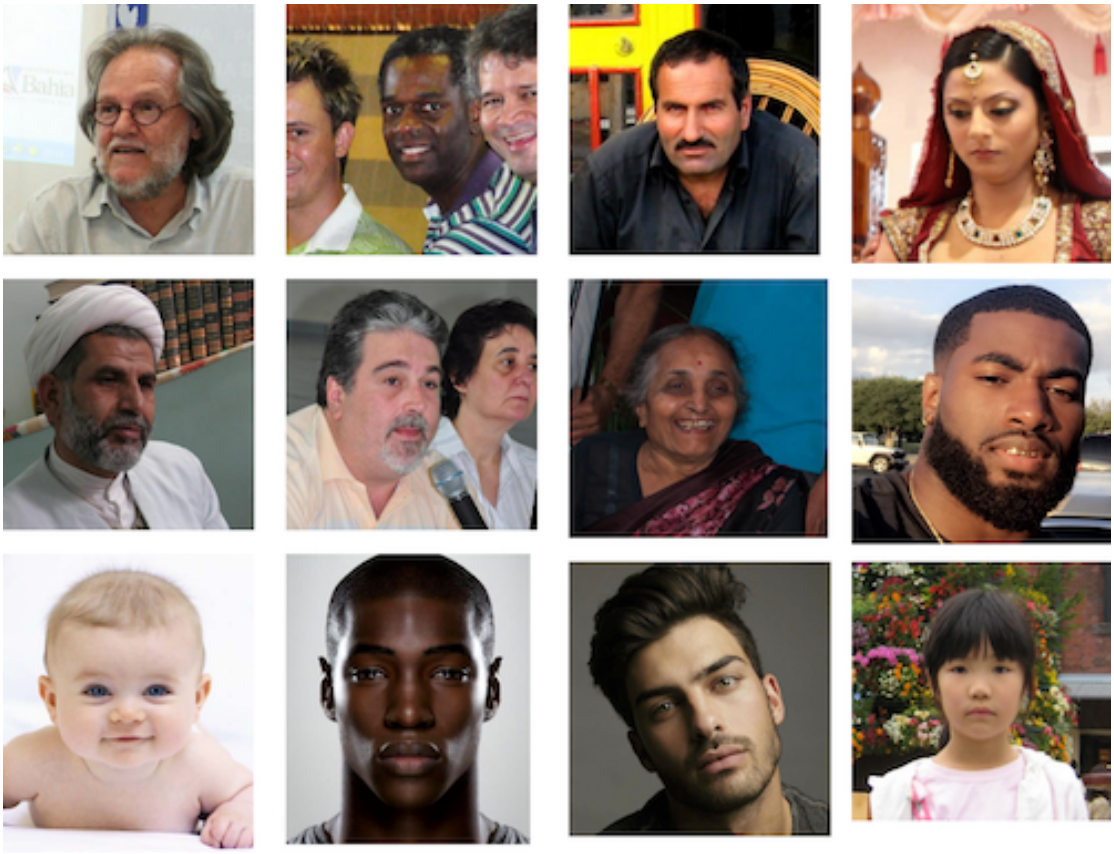


Figure 3.4: Glimpse of Dataset

Dataset Preparation The preparation of the dataset involved collecting images from the below-mentioned sources in table 3.1 placing all in the same bucket. All images were collected and categorized into sub-folders based on their ethnic labels, replacing the initial CSV labeling system. This categorization helps streamline the training process by allowing direct usage of image data organized by ethnicity. Outliers and low-quality images were manually removed to enhance the overall quality and representatives of the dataset. This process ensures that the dataset is well-prepared, diverse, and robust, providing a solid foundation for training the ethnicity detection and colorization model effectively.

Removal of dataset outliers In the process of refining our dataset for the ethnicity-aware auto-colorization project, we performed outlier removal to ensure high-quality input data. This involved excluding images that did not meet our criteria, such as black and white images, which are incompatible with our colorization goals. We also removed images that were blurred or had very low brightness, as they could negatively impact the

model's performance. Additionally, we excluded images without visible humans and those with overly cut-out faces, as they do not provide sufficient contextual information for accurate ethnicity detection and colorization. This careful curation of the dataset helps in training a more accurate and reliable model. 3.5 shows the glimpses of outliers in the dataset.

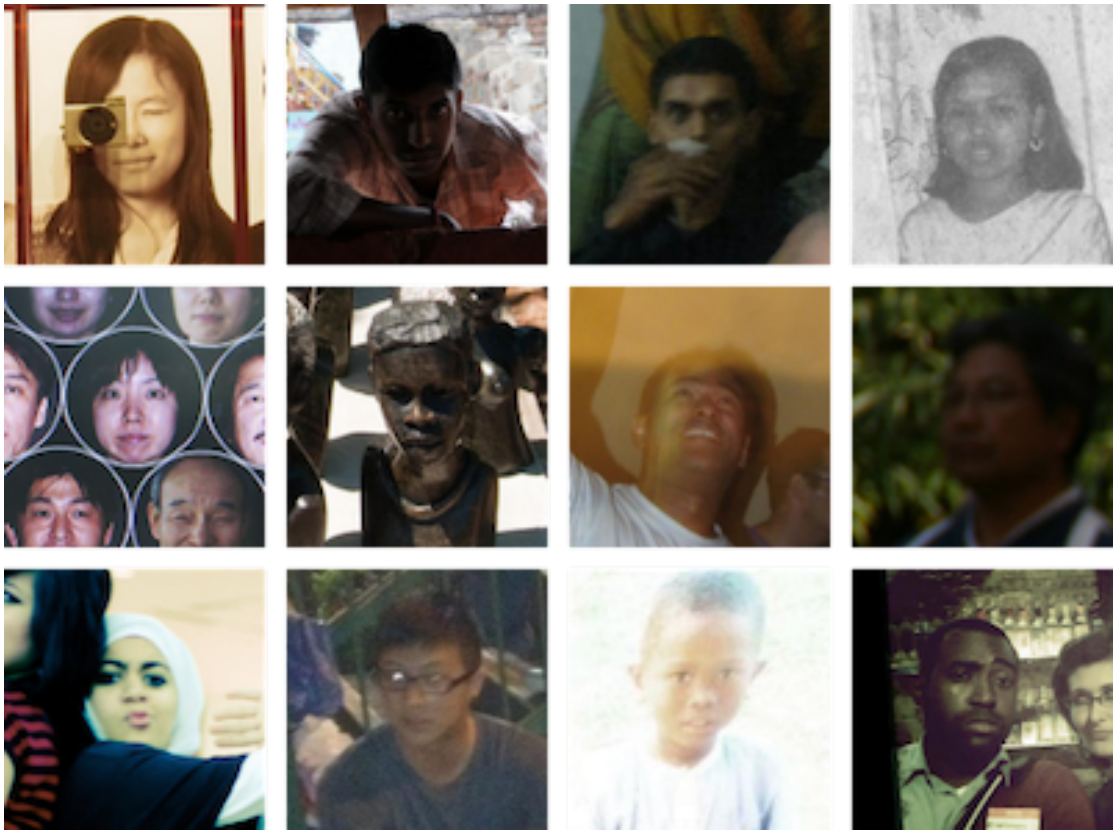


Figure 3.5: Outliers in Dataset

Contents of the dataset

Images

The dataset contains the coloured images of human portraits having diverse ethnic groups including African, Asian, Caucasian, Indians etc with mix of genders and various age groups to ensure the model's applicability across different demographics.

Labels

CSV file will includes detailed labels for each image, specifying the ethnic group. And this label will be used to train the model to recognize and differentiate between various ethnicities.

But the final structure of the dataset is segregated like mentioned in the fig 3.6

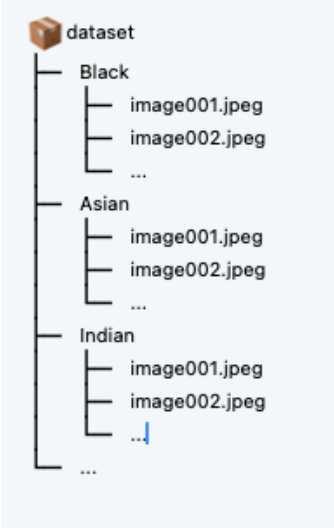


Figure 3.6: Dataset Folder Structure

Sources of dataset

Table 3.1: Detailed Description Dataset Sources

Dataset	**Description**
FairFace Dataset	Contains a collection of human portraits of 6 different race groups: White, Black, Indian, Asian, Middle Eastern, and Latino. From this source I have extracted 50,000 images focusing on balanced ethnic representation and removing outliers from these data. [12]
Kaggle Human Face Dataset	Contains a collection of images with a mix of all common races, age groups, genders, and different lighting conditions, along with some GAN-generated images.
Manual Curation	Created a few thousand images manually from open-source image libraries like Unsplash, featuring human portraits with multiple ethnic group people as the above source mainly includes single person per image. And also included few thousand Nepali people dataset which are sourced from entrance applicants and is provided by my supervisor Er. Umesh Kanta Ghimire

3.6 Description of Algorithms

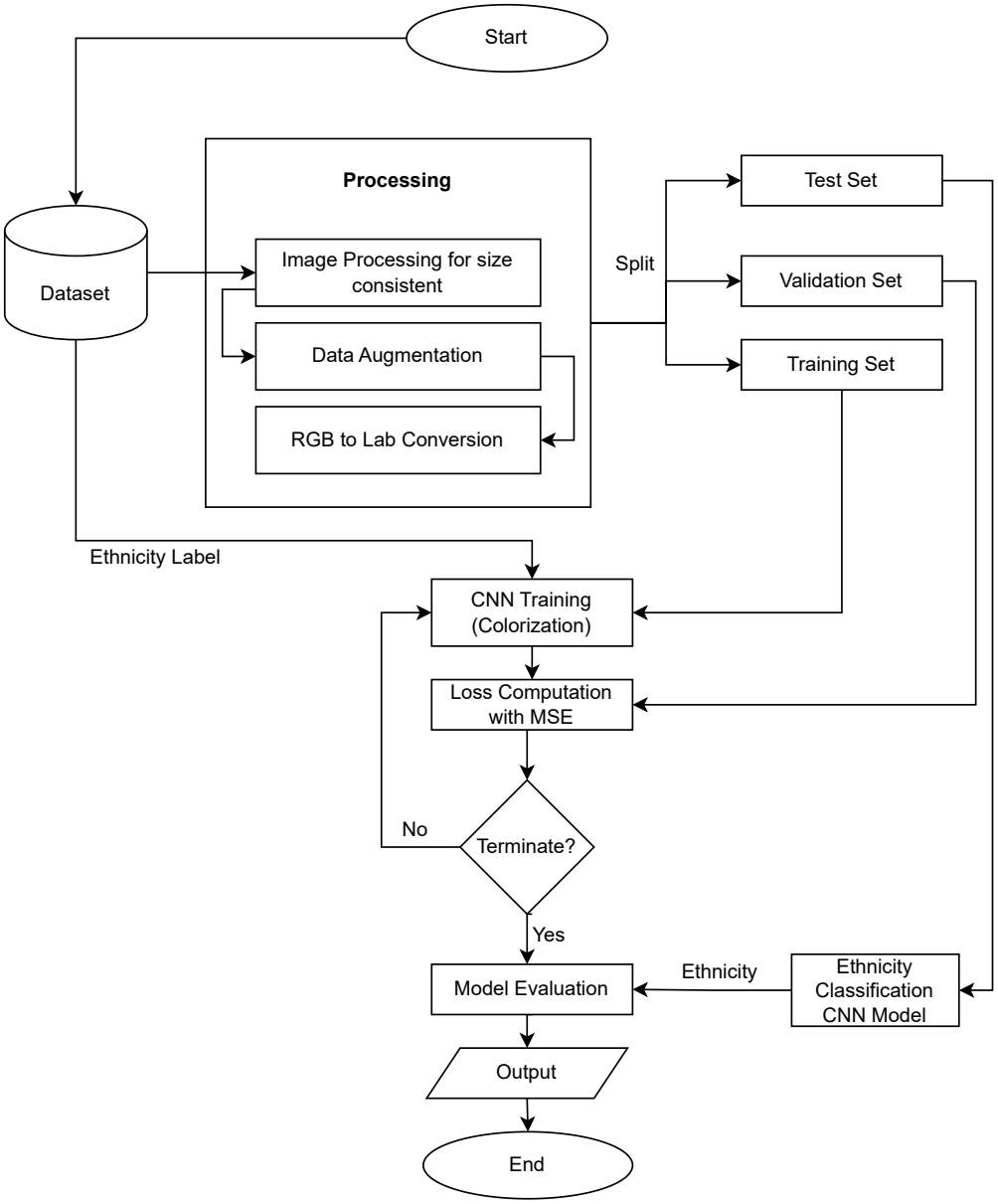


Figure 3.7: Flow Chart of Model

The fig 3.7 shows the flowchart of the overall project starting with the dataset pre processing with image resizing, data augmentation and RGB to Lab Conversion. And splitting datasets to test , validation and training sets. And training model and evaluating the model output.

Preprocessing Step: Converting RGB to LAB Color Space

Algorithm 1 Convert RGB to LAB Color Space

procedure RGB_TO_LAB(R, G, B)
    1. Linearize RGB Values:
    Normalize the RGB values to the range $[0, 1]$
    Apply gamma correction to each channel
    2. Convert RGB to XYZ:
    Transform the linearized RGB values to the XYZ color space using a transformation matrix
    3. Scale XYZ to Reference White:
    Adjust the XYZ values relative to the reference white point
    4. Convert XYZ to LAB:
    Apply the non-linear transformation to obtain LAB values
    Calculate L, a, and b from the scaled XYZ values
    return LAB values (L, a, b)
end procedure

Preprocessing Step: Image Augmentation

Algorithm 2 Apply Image Augmentation

procedure AUGMENT_IMAGE(image)
    1. Apply Spatial Transformations:
        Shear: Apply a shear transformation with a random angle α:
            $I_{\text{shear}}(x', y') = I(x' - \alpha \cdot y', y')$
        Zoom: Scale the image by a random factor z:
            $T_{\text{zoom}} = \begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix}$
            $image \leftarrow T_{\text{zoom}} \times image$
        Rotation: Rotate the image by a random angle θ:
            $T_{\text{rotate}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$
            $image \leftarrow T_{\text{rotate}} \times image$
        Horizontal Flip: Flip the image horizontally with a probability p:
            If $random() < p$, $image \leftarrow \text{flip}(image)$
    2. Apply Photometric Transformations:
        Brightness Shift: Adjust the brightness by a factor β:
            $image \leftarrow image + \beta$
    return The augmented image
end procedure

3.7 Elaboration of Working Principle

3.7.1 Preprocessing of Raw Input Data

1. Data Segregation by Class Labels: In the initial preprocessing step of the project, we organize the dataset by segregating images into class-specific folders based on their labels. The dataset consists of images from six different ethnic groups, and the associated CSV file contains the image filenames and their corresponding ethnicity labels. First, we read the CSV file to retrieve the labels and image filenames. Then, for each unique ethnicity label, we create a corresponding folder within the main dataset directory. Subsequently, we iterate through each row of the CSV file, moving each image to its respective folder based on its ethnicity label. This organization helps streamline the training process for the CNN model by ensuring that all images of the same class are grouped together.

2. Loading the dataset and resizing: The images are loaded from the specified directory using TensorFlow's *image-dataset-from-directory* function. This function helps in organizing the images into batches and labels them automatically. All images are resized to a consistent size of 256x256 pixels to ensure uniformity and to match the input size expected by the CNN model.

3. Conversion to LAB Color Space: Each image is converted from the RGB color space to the LAB color space. This is crucial because the LAB space separates the lightness component (L) from the color components (A and B), which is ideal for the task of colorization where only the color channels need to be predicted.

4. Normalization: The L, A, and B components are normalized to have values between 0 and 1. This helps in stabilizing the learning process as neural networks perform better with smaller, scaled inputs.

5. Data Augmentation: Geometric data augmentation techniques such as shear, zoom, rotation, and horizontal flip also the photo-metric augmentation like brightness shift are applied to the images to artificially increase the dataset size and improve the model's robustness. This helps the model generalize better to unseen data by exposing it to a wider variety of image transformations.

6. Splitting of Dataset: The dataset is split into training, validation, and testing sets.

Typically, 70% of the data is used for training, 20% for validation, and 10% for testing. This ensures that the model is trained on a substantial portion of the data while being validated and tested on separate, unseen portions to evaluate its performance.

By following these preprocessing steps, the raw input data is transformed into a format that is optimal for training a machine learning model for the task of image colorization, leveraging the benefits of the LAB color space and data augmentation techniques to improve model performance and robustness.

3.7.2 Manipulation Through Model Stages

3.7.2.1 Ethnicity Detection Model

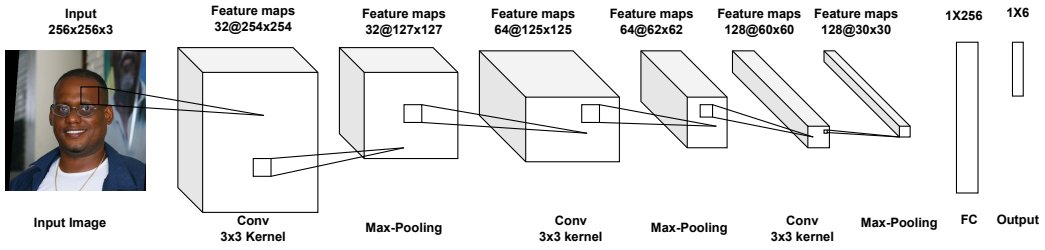


Figure 3.8: Ethnicity Detection Model Architecture

Modal Stages

1. Input layer:

The input layer is the starting point of the neural network, where raw image data is introduced to the model. For this model, the input layer accepts images with dimensions of 256x256 pixels and three color channels (RGB). Prior to feeding the images into the model, they are normalized by scaling the pixel values from their original range of [0, 255] to a range of [0, 1]. This normalization ensures that the training process is more stable and that the model can learn effectively from the data.

2. Intermediate Layer:

The intermediate layers are crucial for extracting and refining features from the input images. These layers include convolutional, batch normalization, max pooling, and dropout layers, each serving a specific purpose:

- **Convolutional Layers:** These layers apply filters to the input images, creating

feature maps that highlight essential features such as edges, textures, and patterns. Each convolutional layer helps the model to detect increasingly complex features as the data progresses through the network.

- **Batch Normalization Layers:** Batch normalization layers standardize the output of the convolutional layers, which speeds up the training process and adds a degree of regularization. This helps in maintaining the stability of the network and improves overall performance.
- **Max Pooling Layers:** Max pooling layers reduce the spatial dimensions of the feature maps, effectively down-sampling the input representation. This reduction in size decreases the computational load for subsequent layers and helps in extracting dominant features, making the model more efficient.
- **Dropout Layers:** Dropout layers introduce regularization by randomly setting a fraction of input units to zero during the training process. This helps in preventing overfitting, ensuring that the model generalizes well to new, unseen data.
- **Flatten Layer:** Before the data can be fed into the dense (fully connected) layers, the 2D matrix of feature maps is flattened into a 1D vector. This transformation allows the dense layers to process the data effectively.

Output Layer

The output layer is a dense layer equipped with a softmax activation function, which generates a probability distribution over the six classes. Each node in this layer represents one of the classes, and the softmax function ensures that the output values sum up to 1. This provides a clear and interpretable classification result, indicating the model's confidence in each class prediction.

3.7.2.2 Colorization Model

The preprocessed data (now in normalized LAB format) enters the CNN, which is structured to predict the A and B components from the L component.[13]

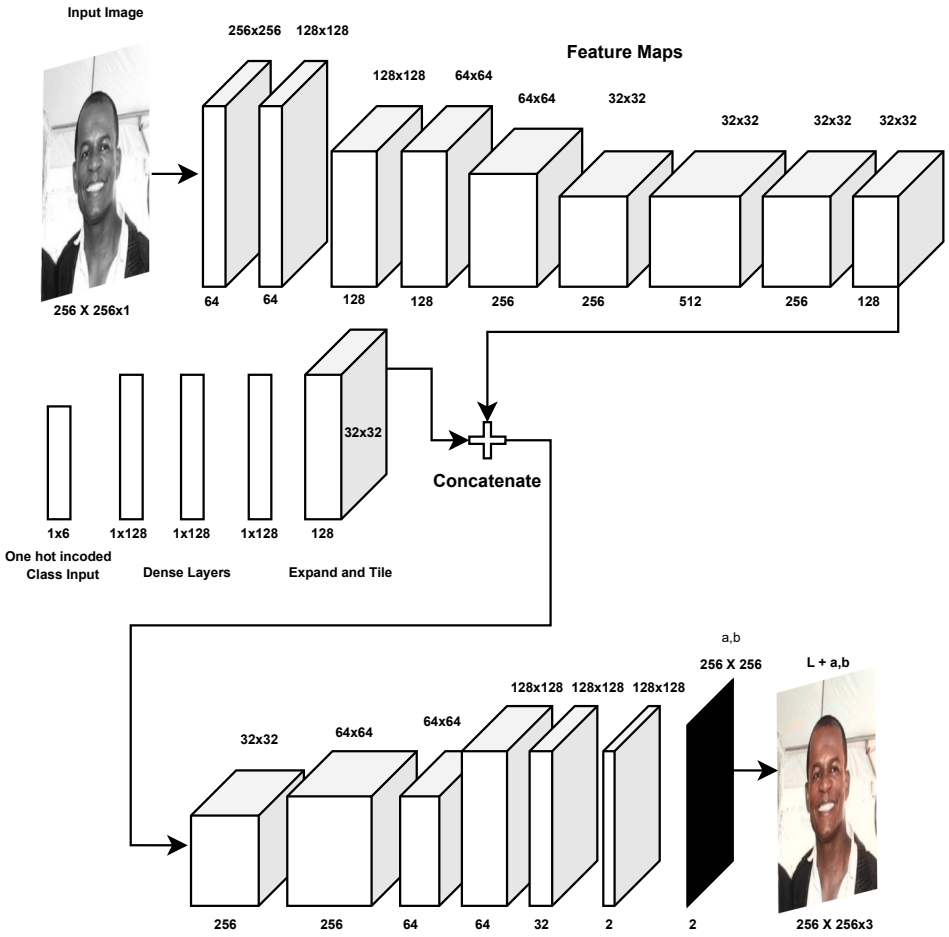


Figure 3.9: Colorization Model Architecture

Modal Stages

1. Input layer:

The input to the model consists of grayscale images with dimensions (256, 256, 1) and one-hot encoded vectors representing the classes associated with the images, useful for providing additional semantic information during the colorization process.

2. Intermediate

- **Convolution Layers:** The first layer applies 64 filters of size 3×3 with ReLU activation, producing feature maps that highlight low-level features such as edges and textures. This is followed by another convolutional layer that not only applies

64 filters of size 3×3 with ReLU activation but also employs striding, reducing the spatial dimensions to 128×128×64. This process of convolution and striding continues, with each layer extracting increasingly complex patterns and reducing the spatial dimensions further. The third and fourth layers use 128 filters, the fifth and sixth layers use 256 filters, and the seventh layer employs 512 filters, all with ReLU activation. This progressive reduction in spatial dimensions through striding helps in downsampling the image while preserving significant features.

- **Class Feature Processing:** Class-specific features are extracted through dense layers. The class labels are processed through three fully connected layers, each with 128 units and ReLU activation. This generates a class feature vector, which is expanded and tiled to match the spatial dimensions of the image feature maps, resulting in a feature map of 32×32×128. This ensures that class-specific information is uniformly incorporated across the entire feature map.

- **Concatenation:** The image feature maps and class feature maps are then concatenated along the channel dimension, forming a combined feature map of size 32×32×256. This concatenation integrates the spatial features from the image with the semantic information from the class labels, enriching the data representation for subsequent processing.

- **Upsampling Layers:** The combined feature map undergoes upsampling to progressively restore the image to its original size. The first upsampling layer increases the dimensions to 64×64×256, followed by a convolutional layer with 64 filters of size 3×3 and ReLU activation, refining the features. This process continues with another upsampling layer, further increasing the dimensions to 128×128×64, followed by a convolutional layer with 32 filters of size 3×3 and ReLU activation.

3. Output layer:

The final layer is a convolutional layer with a tanh activation function, which predicts the two color channels (a and b) of the LAB color space. The tanh activation function ensures that the output values are in the range [-1, 1], which aligns with the normalized

range of the LAB color channels. A final upsampling layer restores the dimensions to 256×256×2, matching the original image size but with predicted color channels.

Sample Manipulation:

If the L component after normalization for a specific pixel is 0.7, the model learns through its layers to predict A and B values, perhaps predicting 0.35 for A and 0.55 for B after processing through all layers.

3.7.3 Post-Processing of Model Output

Once the CNN predicts the A and B components, these need to be post-processed to convert them back into a viewable image format.

1. Combining Channels: The model outputs the A and B channels for each pixel in the image. These predicted channels are combined with the original L channel to form a complete LAB image.

2. Denormalization: The A and B channels output by the model are typically in the range of [-1, 1] due to the tanh activation function used in the final layer. To convert these channels back to their original range, they are multiplied by 128. This denormalization is crucial because the LAB color space expects the A and B channels to be in the range of [-128, 127].

3. LAB to RGB Conversion: The combined LAB image is then converted back to the RGB color space. This conversion is essential because the RGB color space is the standard for image display on most devices.

3.7.4 Sample Calculations for Explanation

1. Preprocessing Steps

Normalization

Normalization scales the pixel values from the range [0, 255] to [0, 1], which helps stabilize and speed up the training process. The formula for normalization is:

$$\text{Normalized\_pixel} = \frac{\text{Original\_pixel}}{255} \tag{3.22}$$

For example, for an image pixel with a value of 120:

$$\text{Normalized\_pixel} = \frac{120}{255} \approx 0.47$$

One-Hot Encoding

One-hot encoding converts categorical labels into binary vectors. For n classes, each label is converted into an n-dimensional vector. For instance, with 6 classes:

- Class 0: [1, 0, 0, 0, 0, 0]
- Class 1: [0, 1, 0, 0, 0, 0]
- Class 2: [0, 0, 1, 0, 0, 0]
- Class 3: [0, 0, 0, 1, 0, 0]
- Class 4: [0, 0, 0, 0, 1, 0]
- Class 5: [0, 0, 0, 0, 0, 1]

Data Augmentation

Data augmentation applies transformations such as rotation, zoom, and flipping to create diverse training samples.

For example, if an image is rotated by 20 degrees:

$$x' = x\cos(20^\circ) - y\sin(20^\circ)$$
$$y' = x\sin(20^\circ) + y\cos(20^\circ)$$

2. ML Manipulation Steps

Convolution

Convolutional layers apply filters to extract features. The output of a convolution operation is given by:

$$\text{Output}(i,j) = \sum_m \sum_n \text{Input}(i+m, j+n) \cdot \text{Kernel}(m,n) \tag{3.23}$$

For a 3×3 filter applied to a region of the input image:

$$\text{Output}(i,j) = \sum_{m=-1}^{1} \sum_{n=-1}^{1} \text{Input}(i+m, j+n) \cdot \text{Kernel}(m,n) \tag{3.24}$$

Striding

Striding reduces the spatial dimensions. If the stride is 2:

$$\text{Original dimension} = 256 \times 256$$
$$\text{After striding} = \left\lfloor \frac{256}{2} \right\rfloor \times \left\lfloor \frac{256}{2} \right\rfloor = 128 \times 128$$

Upsampling

Upsampling increases spatial dimensions. If upsampling by a factor of 2:

$$\text{Original dimension} = 128 \times 128$$
$$\text{After upsampling} = 128 \times 2 \times 128 \times 2 = 256 \times 256$$

3. Post-Processing Steps

Denormalization

Denormalization converts the A and B channels back to the original range [-128, 127].

$$\text{Denormalized\_value} = \text{Normalized\_value} \times 128 \tag{3.25}$$

For a normalized value of 0.5:

$$\text{Denormalized\_value} = 0.5 \times 128 = 64$$

LAB to RGB Conversion

Converting LAB to RGB involves several transformations:

$$Y = \frac{L+16}{116} \tag{3.26}$$

$$X = a \times 0.002 + Y \tag{3.27}$$

$$Z = Y - b \times 0.005 \tag{3.28}$$

Then, converting XYZ to RGB:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{3.29}$$

3.8 Verification and Validation Procedures

For effective verification and validation of the our auto-colorization system, it's crucial to implement a combination of quantitative and qualitative metrics. This approach ensures that the system's output not only meets technical accuracy standards but also aligns with human perceptions of color quality.

3.8.1 Confusion Matrix

The confusion matrix is a performance measurement tool for classification models. It shows the actual versus predicted classifications done by the model. The matrix consists of the following components:

- **True Positives (TP)**: Correctly predicted positive cases.
- **True Negatives (TN)**: Correctly predicted negative cases.
- **False Positives (FP)**: Instances where positive cases are incorrectly predicted (Type I error).
- **False Negatives (FN)**: Instances where negative cases are incorrectly predicted (Type II error).

The confusion matrix structure is shown below:

Table 3.2: Confusion Matrix Structure

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

From the confusion matrix, we can derive several important metrics:

- **Accuracy:** Measures the overall effectiveness of the classifier.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.30}$$

- **Precision:** Represents the proportion of true positive predictions out of all predicted positive cases.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3.31}$$

- **Recall (Sensitivity or True Positive Rate):** Measures the proportion of actual positives correctly identified by the model.

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3.32}$$

- **Specificity (True Negative Rate):** Measures the proportion of actual negatives correctly identified.

$$\text{Specificity} = \frac{TN}{TN+FP} \tag{3.33}$$

- **F1-Score:** The harmonic mean of precision and recall.

$$\text{F1-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.34}$$

3.8.2 Receiver Operating Characteristic (ROC) Curve

The ROC curve is a graphical tool that demonstrates the diagnostic performance of a binary classifier system as its decision threshold is adjusted. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings.[14]

- **True Positive Rate (TPR):** Also known as sensitivity or recall.

$$\text{TPR} = \frac{TP}{TP + FN} \tag{3.35}$$

- **False Positive Rate (FPR):** The probability of false alarm.

$$\text{FPR} = \frac{FP}{FP + TN} \tag{3.36}$$

The Area Under the ROC Curve (AUC) provides a single scalar value summarizing the performance of the model. An AUC of 1 indicates perfect classification, while an AUC of 0.5 suggests random guessing.

3.8.3 Precision-Recall (PR) Curve

The PR curve is a plot of precision against recall for different threshold values. It is particularly useful in cases of imbalanced datasets.

- **Precision:** Measures the accuracy of positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.37}$$

- **Recall:** Measures the coverage of actual positives.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.38}$$

The area under the precision-recall curve (PR AUC) provides a summary measure of the model's performance, similar to the ROC AUC.

Interpretation

These metrics provide comprehensive insights into a model's classification performance. They help in understanding not only the accuracy of the model but also the types of errors it is prone to, and how different thresholds affect performance metrics.

3.8.4 Mean Squared Error (MSE)

MSE measures the average of the squares of the errors, the average squared difference between the estimated values (colorized output) and what is regarded as the true values (original color images). Lower MSE values indicate higher accuracy in colorization.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{3.39}$$

Where,

- n - Number of data points or pixels in the image.
- Y_i - Actual value of the i-th data point in the original color image.
- $\hat{Y}_i$ - Predicted value of the i-th data point by the colorization model.

3.8.5 Peak Signal-to-Noise Ratio (PSNR)

PSNR is used to measure the quality of reconstruction of lossy compression codecs, defined as the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. PSNR is useful in the image processing as it gives more intrepretable indication of the quality of the image that means higher PSNR value typically indicate higher quality of the image that is perceived by the real human eyes. [15]

$$\text{PSNR} = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{3.40}$$

Where,

- MAX_I - Maximum possible pixel value of the image (usually 255 for 8-bit images).
- MSE - Mean Squared Error as defined above.

3.8.6 Structural Similarity Index (SSIM)

The Structural Similarity Index (SSIM) is a perceptual metric used to evaluate the similarity between two images by comparing their structural information, luminance, and contrast. It is designed to align more closely with human visual perception than traditional metrics like MSE or PSNR.[16] The SSIM index between two images x and y is calculated as:

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3.41}$$

where μ_x and μ_y are the mean intensities, σ_x and σ_y are the standard deviations, σ_{xy} is the covariance of x and y, and C_1 and C_2 are constants to stabilize the division. SSIM values range from -1 to 1, where 1 signifies complete similarity.

3.8.7 Learned Perceptual Image Patch Similarity (LPIPS)

The Learned Perceptual Image Patch Similarity (LPIPS) is a perceptual metric designed to evaluate the visual similarity between two images. Unlike traditional metrics like MSE or PSNR, which rely on pixel-wise comparisons, LPIPS leverages deep neural networks to assess image similarity based on features that are more aligned with human visual perception.

LPIPS computes the perceptual distance between two images by passing them through a pretrained convolutional neural network (CNN), such as VGG or AlexNet, and comparing the resulting feature representations. The network used is typically trained on large-scale image classification tasks, which equips it with the ability to capture both low-level details (e.g., edges and textures) and high-level features (e.g., shapes and objects) that are important for human perception.

The LPIPS score is calculated as a weighted average of the differences between the images' feature representations across several layers of the network. The formula for LPIPS can be generalized as follows:

$$\text{LPIPS}(x,y) = \sum_{l} \frac{1}{H_l W_l} \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} w_l |\phi_l^h(x) - \phi_l^h(y)|_2^2 \tag{3.42}$$

Where:

- $\phi_l(x)$ and $\phi_l(y)$ represent the feature maps of images x and y at layer l.
- H_l and W_l are the height and width of the feature maps at layer l.
- w_l is the weight assigned to the feature map differences at layer l.

The output of LPIPS is a single scalar value, where lower values indicate higher perceptual similarity between the two images, and higher values suggest more significant perceptual differences.

Application in Auto-Colorization

In the context of auto-colorization, LPIPS is employed to measure the perceptual quality of the colorized images produced by the model. It compares the generated images with the ground truth color images, providing a metric that reflects how close the model's output is to what would be perceived as a realistic and accurate colorization by a human viewer. By incorporating LPIPS in the validation process, we ensure that our model not only meets technical accuracy standards but also delivers outputs that are visually pleasing and align with human expectations of color quality. [17]

3.8.8 Qualitative Evaluation

Visual comparisons will be made between the colorized images and the ground truth color images to assess the model's success in generating realistic and visually appealing colorizations. Here the colorization system's performance in real-world terms, measuring how well the colorized images meet human standards of color accuracy and aesthetic quality. This evaluation process is crucial as the quantitative metrics like MSE and PSNR might not always fully capture the visual aspects of the sensible color.

4 RESULTS

The results of our experiments on auto colorization of grayscale images using a Convolutional Neural Network (CNN) is presented in this section.

4.1 Ethnicity Detection Model

For our ethnicity detection model, we conducted three experiments to optimize the model's performance by varying the dataset size, learning rate, and other hyperparameters. In the first experiment, we utilized a unbalanced dataset with a specific number of images per class, adjusting the learning rate to find an optimal balance between convergence speed and stability. In the second experiment, we balanced the dataset size, introducing more diversity and complexity to the data. Alongside, we fine-tuned other hyperparameters, such as batch size and the number of epochs, to enhance the model's accuracy and generalization capabilities. Finally, in the third experiment, we further refined the learning rate and applied early stopping to prevent overfitting. These experiments are crucial in understanding the model's sensitivity to different configurations and identifying the most effective setup for robust ethnicity detection.

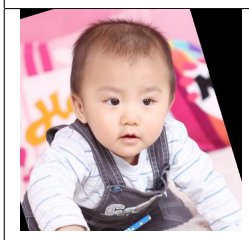
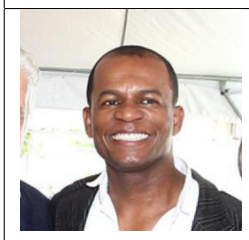
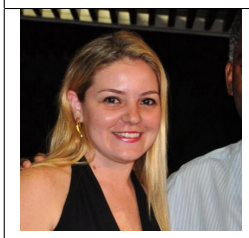
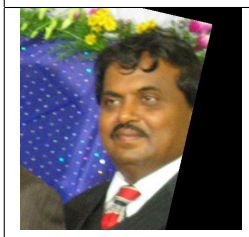
- **Experiment 1:** Total 9000 images with unbalanced images per class, Batch size of 16, learning rate of 0.0001 and 60 epochs.
- **Experiment 2:** Total 9000 images with 1500 images per class, Batch size of 32, leaning rate of 0.0001 and 120 epochs.
- **Experiment 3:** Total 9000 images with 1500 images per class, batch size of 32, learning rate of 0.00001, and 134 epochs with early stopping.

4.1.1 Outputs for Various Scenarios

4.1.1.1 Best Case Scenario

The best-case scenario is obtained in the Asian, Black, White and Indian ethnic Group. The following images in table 4.1 represent the best-case outputs of different ethnic group where the model achieved high prediction scores. The high prediction scores suggest that the model has learned to identify distinctive features and patterns associated with these ethnic groups, leading to reliable and consistent classification outcomes.

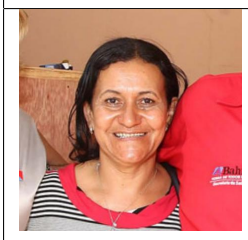
Table 4.1: Best Case Scenario: Ethnicity Detection

Image	True Value	Predicted Value	Prediction Score
	Asian	Asian	1.0000
	Black	Black	1.0000
	White	White	0.9999
	Indian	Indian	0.9943

4.1.1.2 Worst Case Scenario

In the worst-case scenario, our ethnicity detection model struggles with accurately predicting the Latino and Middle Eastern ethnic groups, resulting in lower prediction scores compared to the other categories. The following images in table 4.2 represent the worst-case outputs with lower somewhat lower prediction scores.

Table 4.2: Worst Case Scenario: Ethnicity Detection

Image	True Value	Predicted Value	Prediction Score
	Latino	Indian	0.9991
	Latino	Black	0.9999
	Middle East	Asian	1.0000
	Indian	Black	1.0000

4.1.2 Quantitative Metrics

We visualized the training and validation loss, confusion matrix, and ROC metrics over the epochs to understand the model's learning process and performance.

4.1.2.1 Loss Function

The following figure shows the training and validation Mean Squared Error (MSE) over the epochs for each experiment. Experiment 3, which utilized 9000 images with a learning rate of 0.00001, shows the best performance with a smooth decreasing curve in validation loss.

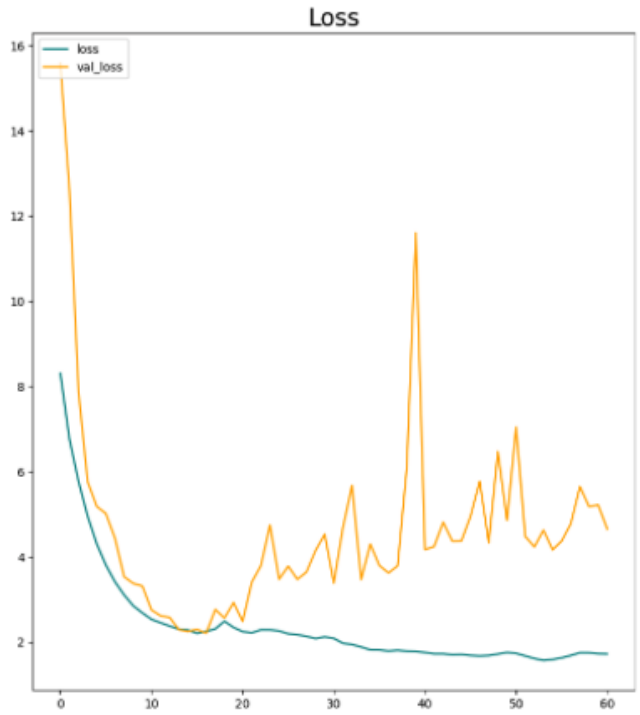


Figure 4.1: MSE Graph of Experiment 1 of Ethnicity Model

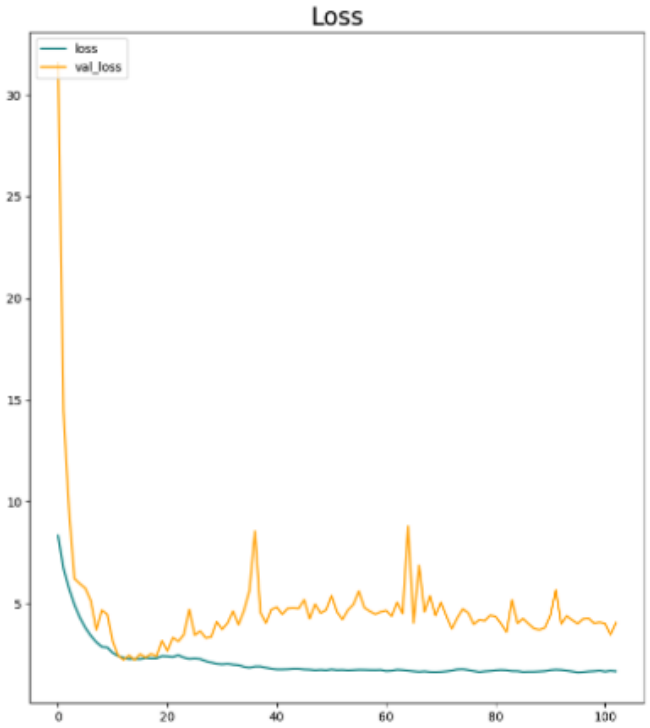


Figure 4.2: MSE Graph of Experiment 2 of Ethnicity Model

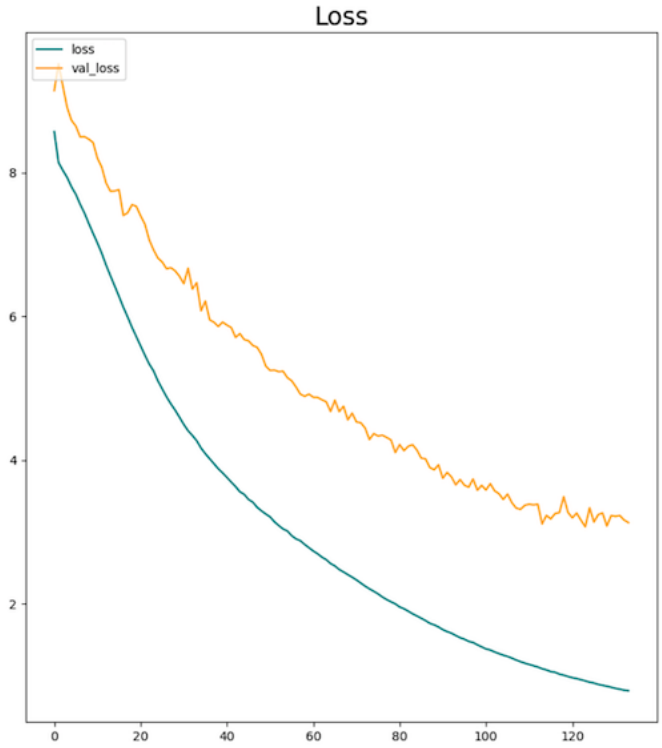


Figure 4.3: MSE Graph of Experiment 3 of Ethnicity Model

4.1.2.2 Confusion Matrix

The following figure shows the confusion matrix for each experiment. Both experiments reveal that the model struggles with accurately classifying the Latino and Middle Eastern ethnic groups, showing high misclassification rates. However, it performs well with Asian, Black, White, and Indian groups, achieving high accuracy in these categories. In Experiment 2, overall accuracy improved, indicating better generalization and fewer instances of confusion between ethnic groups, especially for well-represented classes. However, confusion remains a notable issue for Latino and Middle Eastern classifications, where the model still struggles to distinguish between these and other groups accurately. Experiment 3, which used a learning rate of 0.00001, further improved validation performance with a smoother decreasing curve in validation loss. However, it still shows significant confusion when classifying the Latino group. While the model demonstrated improved stability and generalization, the persistent challenge with Latino and Middle Eastern classifications underscores the need for further optimization.

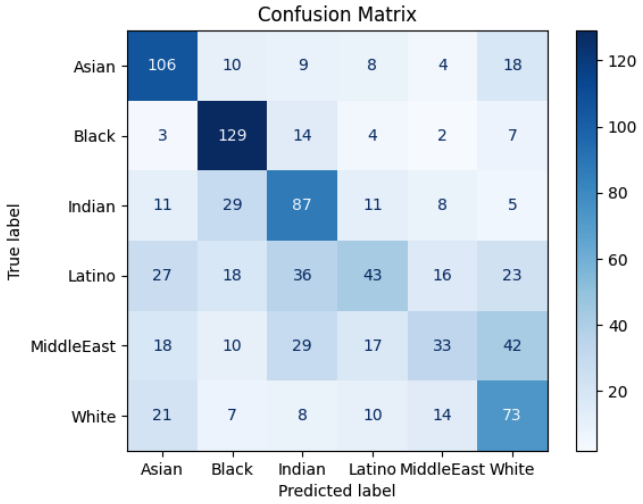


Figure 4.4: Confusion matrix of Experiment 1 of Ethnicity Model

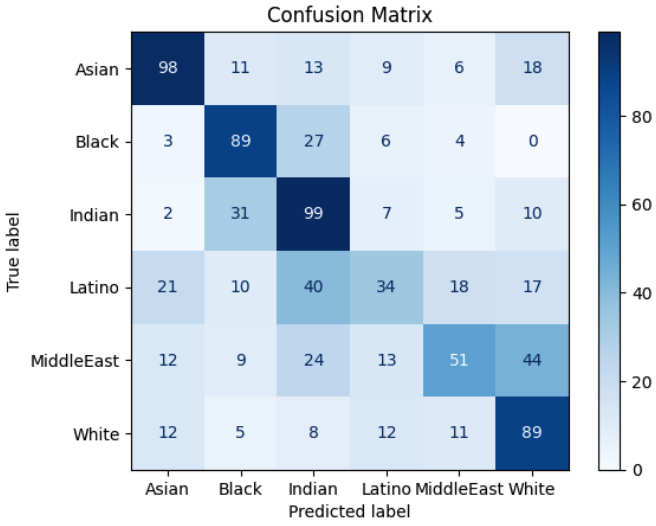


Figure 4.5: Confusion matrix of Experiment 2 of Ethnicity Model

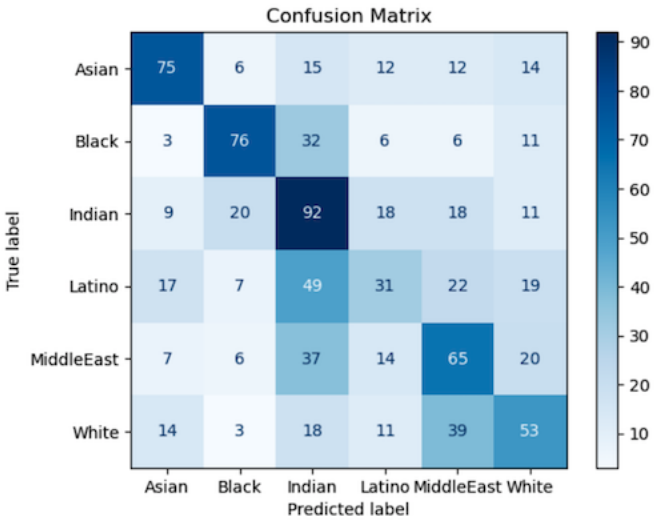


Figure 4.6: Confusion matrix of Experiment 3 of Ethnicity Model

4.1.2.3 Receiver Operating Characteristics(ROC)

The ROC (Receiver Operating Characteristic) curve is a graphical representation used to evaluate the performance of a classification model at various threshold settings. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR), providing insights into the model's ability to distinguish between different classes.

In Experiment 1 as shown in fig 4.7 , the ROC curves for Asian, Black, White, and Indian ethnic groups demonstrate a good level of discrimination, with areas under the curve (AUC) close to 1. This indicates strong predictive performance for these groups. However, the ROC curves for the Latino and Middle Eastern groups are less favorable, with lower AUC values, suggesting that the model struggles to accurately differentiate these groups from others, resulting in higher false positive rates.

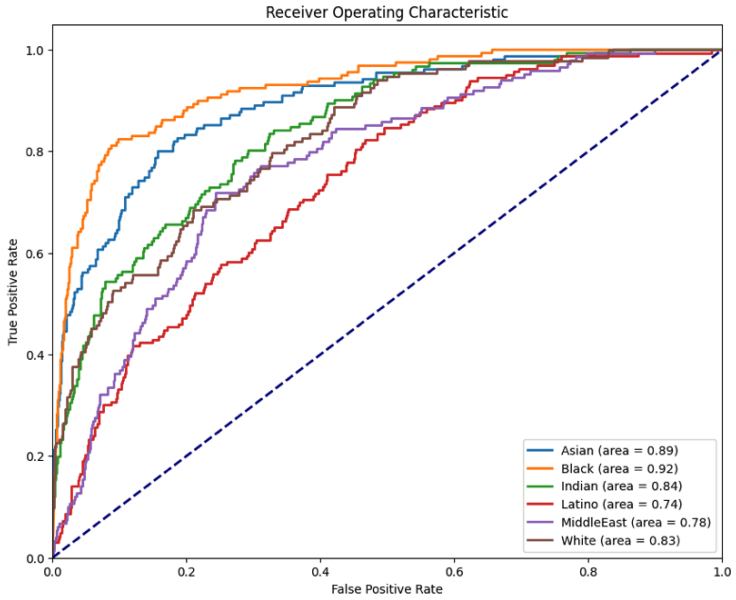


Figure 4.7: ROC Curve of Experiment 1 of Ethnicity Model

In Experiment 2 as shown in fig 4.8, the ROC curves show an overall improvement across all ethnic groups. The AUC values for Asian, Black, White, and Indian groups remain high, further solidifying the model's strong performance in these areas. Notably, the ROC curves for Middle Eastern groups also show improvement, with increased AUC values compared to Experiment 1. But Latino AUC still low, the AUC values still highlight the need for further enhancement in these specific groups to reduce misclassification.

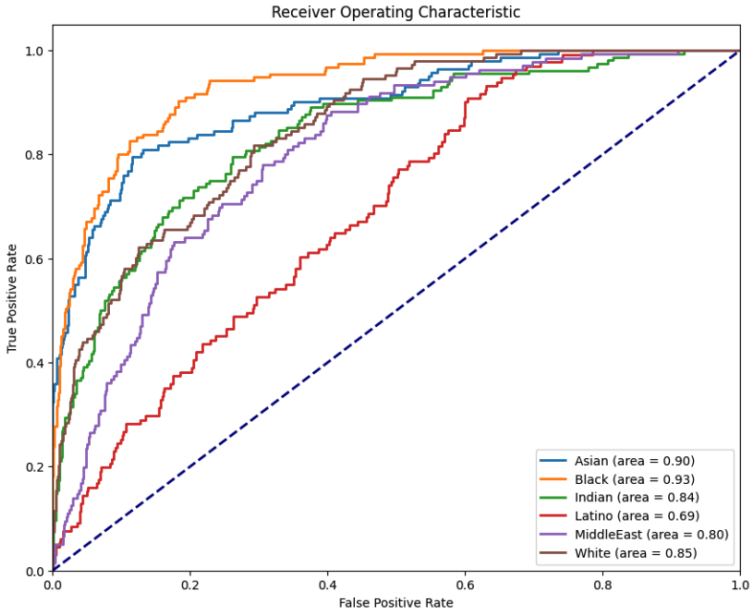


Figure 4.8: ROC Curve of Experiment 2 of Ethnicity Model

In Experiment 3 as shown in fig 4.9, the ROC curves doesn't show significant improvement but still has increased the area of overall classes to few points.

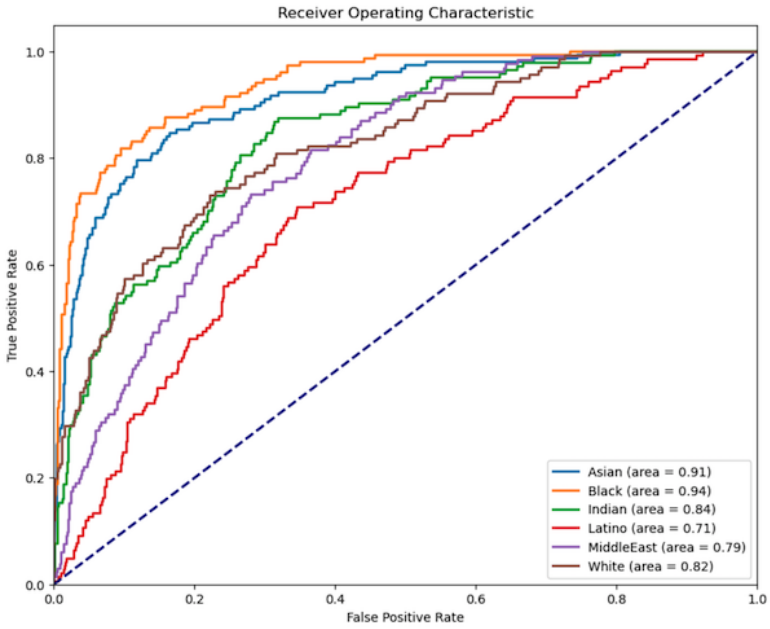


Figure 4.9: ROC Curve of Experiment 3 of Ethnicity Model

4.1.2.4 Precision Recall Curve (PR)

The PR (Precision-Recall) curve is another important tool for evaluating the performance of a classification model, especially when dealing with imbalanced datasets. It plots Precision (the proportion of true positive predictions among all positive predictions) against Recall (the proportion of true positives identified out of all actual positives), providing insights into the trade-offs between these metrics.

In Experiment 1, the PR curves for the Asian, Black, White, and Indian groups are relatively strong, indicating high precision and recall for these classifications. This suggests that the model is good at correctly identifying these ethnic groups while minimizing false positives. However, the PR curves for Latino and Middle Eastern groups are less favorable, with lower precision and recall, indicating that the model struggles with accurately identifying these groups and often misclassifies them.

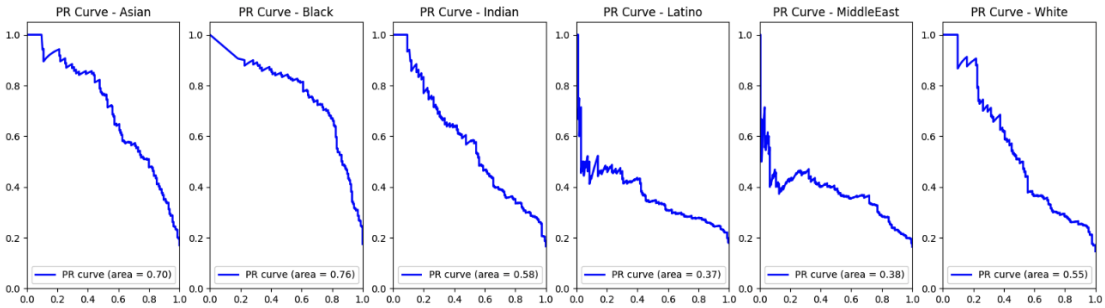


Figure 4.10: PR Curve of Experiment 1 of Ethnicity Model

In Experiment 2, the Asian, Black, White, and Indian groups continue to show high precision and recall, demonstrating the model's robust performance in these categories. The PR curves for the Latino and Middle Eastern groups still do not achieve the same level of performance as the other groups, highlighting ongoing challenges in achieving high precision and recall.

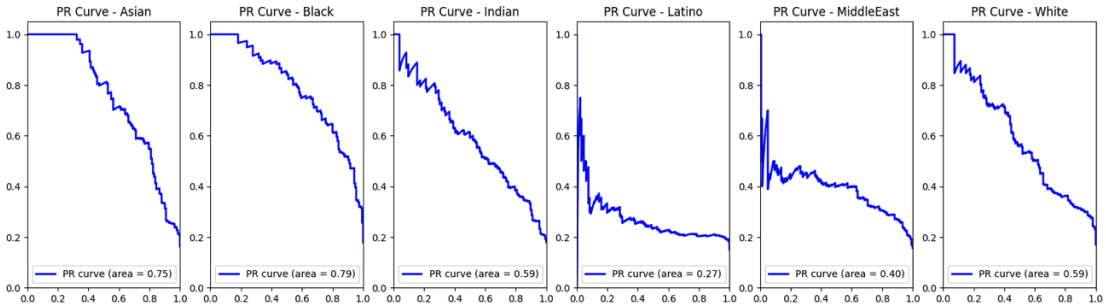


Figure 4.11: PR Curve of Experiment 2 of Ethnicity Model

In Experiment 3, the Asian, Black, White, and Indian groups continue to show high precision and recall, demonstrating the model's robust performance in these categories. Here PR for the MiddleEast shows improvement in precision and recall and still struggling with latino.

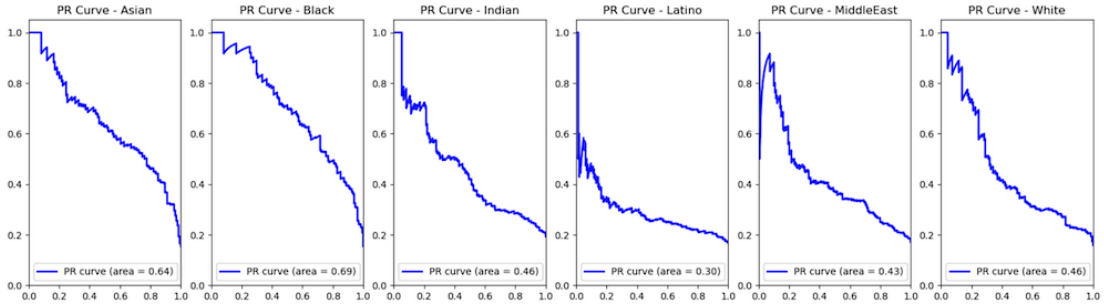


Figure 4.12: PR Curve of Experiment 3 of Ethnicity Model

4.2 Colorization Model

For the colorization model three initial experiments were conducted with varying batch sizes, dataset sizes, and epochs to analyze the impact on model performance. The performance of the model was evaluated using metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). And following experiments were conducted with 6 classes of ethnicity.

- **Experiment 1:** Total 1200 images with 200 images per class, Batch size of 16, and 100 epochs.
- **Experiment 2:** Total 1200 images with 200 images per class, Batch size of 32, and 300 epochs.

- **Experiment 3:** Total 2400 images with 400 images per class, Batch size of 32, 150 epochs and removing outlier datasets

After these initial experiments, we conducted a further experiment to assess the model's performance with a significantly larger dataset:

- **Latest Experiment :** A total of 9000 images were used, with 1500 images per class. The batch size was set to 16, and the model was trained for 100 epochs. And later we fine tune the model using the model and recompiling with lower leaning rate of 1e-5 and using early stopping callback to stop overfitting and saving training time. Also we use callback to reduce the learning rate if validation loss plateaus. And this run for 25 epochs. This experiment aimed to evaluate the scalability of the model and the effect of a larger, more diverse dataset on the colorization quality.

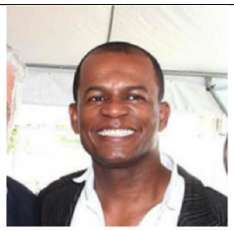
Each of these experiments provided valuable insights into the model's behavior under different training conditions and helped in fine-tuning the model for better performance in colorizing grayscale images based on ethnicity.

4.2.1 Outputs for Various Scenarios

4.2.1.1 Best Case Scenario

For the best-case scenario, we focused on images from all ethnic groups that had minimal background noise, front orientation of the face, a single person in the image, and fewer obstacles like glasses or beard. The following images in table 4.3 represent the best-case outputs of black ethnic group where the model achieved high PSNR, SSIM and low LPIPS scores.

Table 4.3: Best Case Scenario: Black Ethnic group

Grayscale Image	Ground Truth	Colorized Image	PSNR	SSIM	LPIPS
			31.56	0.97	0.0416
			25.84	0.96	0.1126

Below in table 4.4, we present examples from multiple ethnic group under best case scenario. Here we represent Middle East, Indian, Asian and Latino ethnic group respectively. And these visual result and the PSNR, SSIM and LPIPS score shows the promising result for these scenerio where there is minimal background noise, front orientation of the face, a single person in the image, and fewer obstacles like glasses or beard.

Table 4.4: Best Case Scenario: Middle East, Indian, Asian and Latino Ethnic group respectively

Grayscale Image	Ground Truth	Colorized Image	PSNR	SSIM	LPIPS
			25.6	0.94	0.0804
			23.7	0.91	0.1746
			22.0	0.89	0.1749
			25.3	0.96	0.1142

Below in table 4.5, we present examples from various ethnic groups under the best-case scenario where multiple faces are present in the image. This scenario showcases the model's ability to accurately colorize images featuring multiple faces with clear face and minimal obatacles. The visual results, along with the PSNR, SSIM and LPIPS scores, demonstrate the model's effectiveness in handling complex scenarios with multiple subjects. In these cases, the model successfully maintains consistent colorization across different faces while preserving the unique skin tones and features of each ethnic group.

Table 4.5: Best Case Scenario: Multiple faces in single image

Grayscale Image	Ground Truth	Colorized Image	PSNR	SSIM	LPIPS
			25.5	0.95	0.0784
			22.5	0.90	0.1156
			24.0	0.94	0.1249

The model's performance in these cases can be attributed to the clear delineation of features and less complex background, making it easier for the model to predict accurate colors.

4.2.1.2 Worst Case Scenario

The worst performance was observed in images with low contrast, complex backgrounds, and overlapping features. And also the orientation of the face, multiple people in a image, hand and neck region and those ethic with lighter skin tones. These conditions were challenging for the model to accurately colorize.

Table 4.6 illustrates the worst-case scenarios where the model faces significant challenges. The model struggles particularly with accurately colorizing the lighter skin tones of individuals from the White ethnic group. This issue arises due to the subtle variations in lighter skin tones, which are harder for the model to distinguish and replicate accurately.

Furthermore, the model has difficulty colorizing the visible neck and hand regions across

various ethnic groups. These areas often exhibit different shading and texture compared to the face, leading to inconsistent colorization. This highlights the model's limitations in handling complex lighting and shading variations in these regions.

Table 4.6: Worst Case Scenario - White Ethnic Group

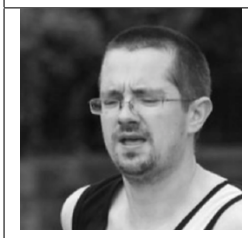
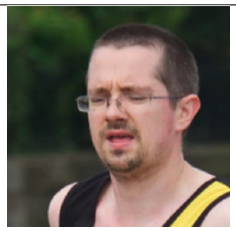
Grayscale Image	Ground Truth	Colorized Image	PSNR	SSIM	LPIPS
			24.39	0.93	0.1719

Table 4.7 highlights the worst-case scenarios where the model encounters significant difficulties. The model struggles particularly with accurately colorizing hair and visible body parts such as the neck and hands in some of the images.

Table 4.7: Worst Case Scenario: Hair and visible body part color

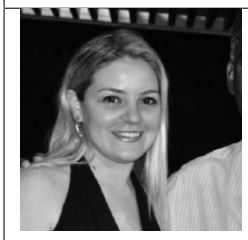
Grayscale Image	Ground Truth	Colorized Image	PSNR	SSIM	LPIPS
			25.83	0.95	0.0859

Table 4.8 illustrates the worst-case scenarios where the model faces significant difficulties in colorizing images accurately. In particular, the model struggles with correctly colorizing beard colors in individuals from the Middle Eastern ethnic group. This challenge arises due to the varied and often complex textures and shades of beard hair, making it difficult for the model to achieve accurate and consistent colorization.

Table 4.8: Worst Case Scenario: Beard Color

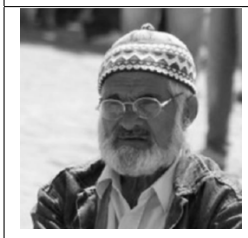
Grayscale Image	Ground Truth	Colorized Image	PSNR	SSIM	LPIPS
			22.62	0.94	0.1417

Table 4.9 highlights the worst-case scenarios where the model encounters significant challenges. The model particularly struggles with colorizing multiple faces in a single image, especially when the facial orientation is not directly facing the camera or when the image contains the face a bit far without closeup. And model is struggling to colorize visible body parts when there are multiple subjects in a image.. This highlights the model's limitations in handling images with multiple people without closeup.

Table 4.9: Worst Case Scenario: Multiple people in image

Grayscale Image	Ground Truth	Colorized Image	PSNR	SSIM	LPIPS
			25.0	0.96	0.0783
			24.9	0.9458	0.1580

4.2.2 Quantitative Metrics

We visualized the training and validation accuracy, PSNR, and SSIM metrics over the epochs to understand the model's learning process and performance.

4.2.2.1 Initial Experiments 1, 2 and 3

Accuracy

The following graph shows the training and validation accuracy over the epochs for each experiment.

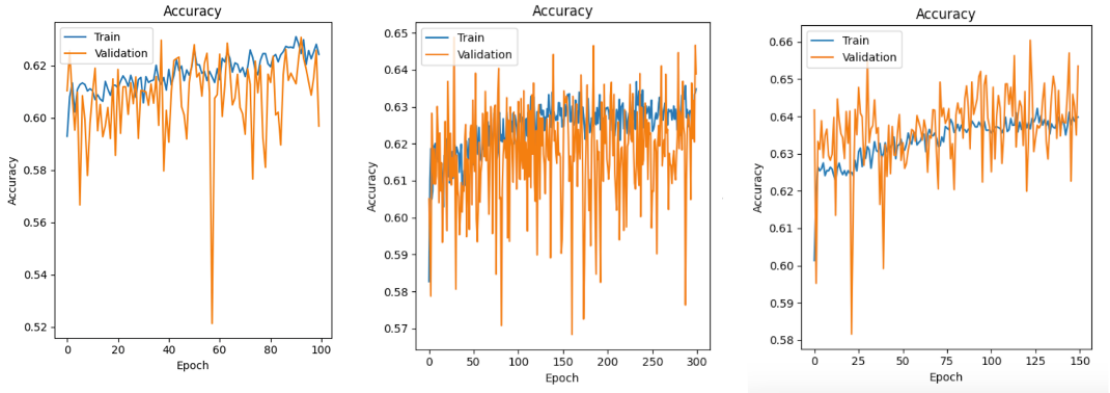


Figure 4.13: Accuracy Graph of Experiment 1, 2 and 3 respectively

The accuracy plots indicate jitteriness in the validation accuracy for both Experiment 1 and Experiment 2. This jitteriness suggests that the model was experiencing overfitting due to the limited dataset size and the presence of outliers. In Experiment 3, there is noticeably less jitteriness, attributed to the removal of outlier datasets and the increased number of images, which together helped reduce overfitting.

Additionally, it shows limitations of using accuracy as a metric for colorization tasks. While accuracy can indicate how well the model predicts the correct class labels, it does not fully capture the quality of the colorization, especially for elements like clothes and background colors. These areas often exhibit a wide range of colors and are less consistent than skin tones, leading to less reliable accuracy metrics.

Loss Function

The following figures show the training and validation MSE over the epochs for each experiment.

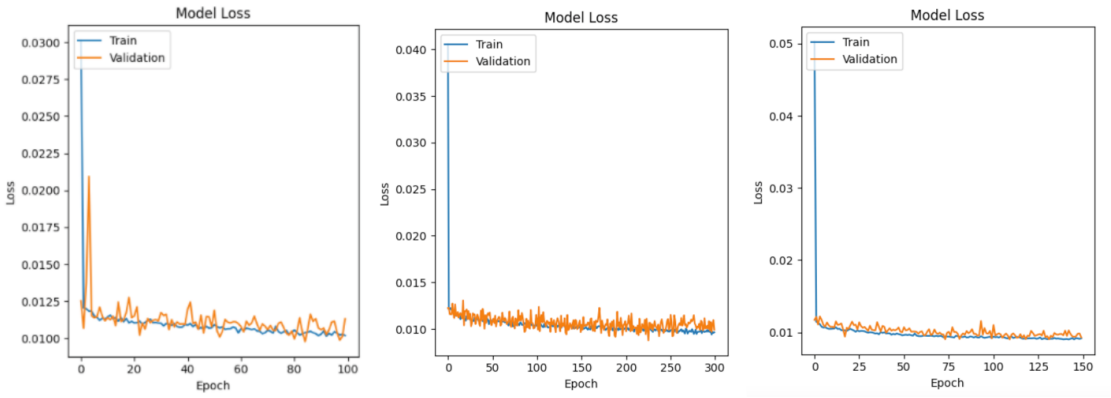


Figure 4.14: MSE Graph of Experiment 1, 2 and 3 respectively

The graphs illustrate the training and validation MSE for the three experiments. Experiment 3, with 2400 images, showed the best overall performance with lower MSE, indicating better model generalization and colorization quality. Experiment 2 also showed significant improvement due to the longer training duration, while Experiment 1 showed moderate improvement over 100 epochs.

PSNR

The following graph shows the PSNR values over the epochs for each experiment.

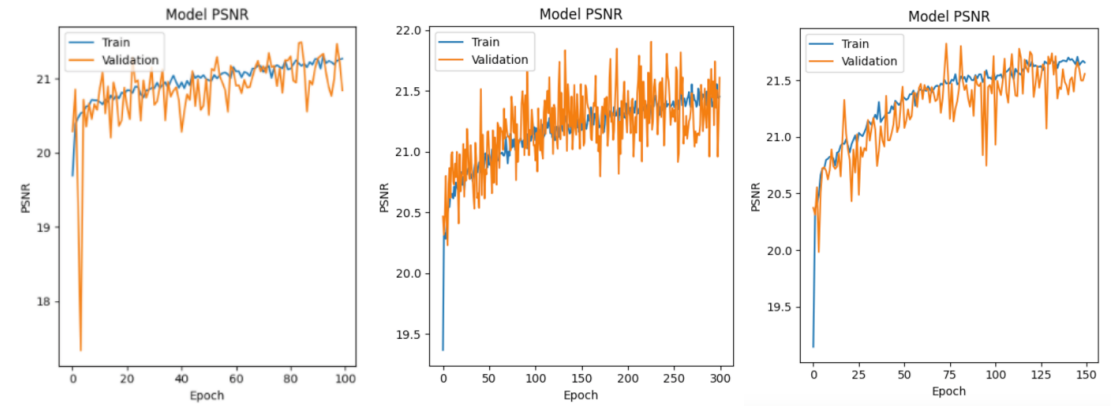


Figure 4.15: PSNR Graph of Experiment 1, 2 and 3 respectively

The PSNR graph shows that Experiment 3, with 2400 images, achieved higher PSNR values consistently, indicating better image quality and more accurate colorization. Experiment 2 also showed good PSNR values due to the longer training duration, while Experiment 1 showed lower PSNR due to fewer epochs. Experiment 1 and 2 have more

fluctuating validation plot of PSNR but the experiment 3 have a bit less fluctuation after removing outliers from dataset and increasing dataset number.

SSIM

The following graph shows the SSIM values over the epochs for each experiment.

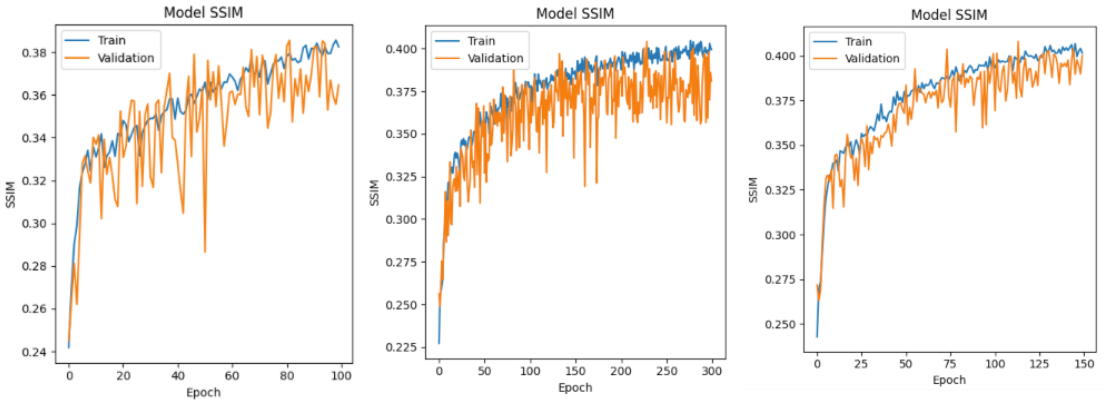


Figure 4.16: SSIM Graph of Experiment 1, 2 and 3 respectively

The SSIM values improved with the larger dataset in Experiment 3, showing better structural similarity in the colorized images. Experiment 2 also showed improved SSIM values due to the extended training period, while Experiment 1 had lower SSIM due to fewer epochs and a smaller batch size. Experiment 1 and 2 have more fluctuating validation plot of SSIM but the experiment 3 have a bit less fluctuation after removing outliers from dataset and increasing dataset number.

4.2.2.2 Latest Experiment - Large Scale Training and Fine Tuning

PSNR

The graph below shows the PSNR values over the epochs for both the initial training phase and the fine-tuning phase.

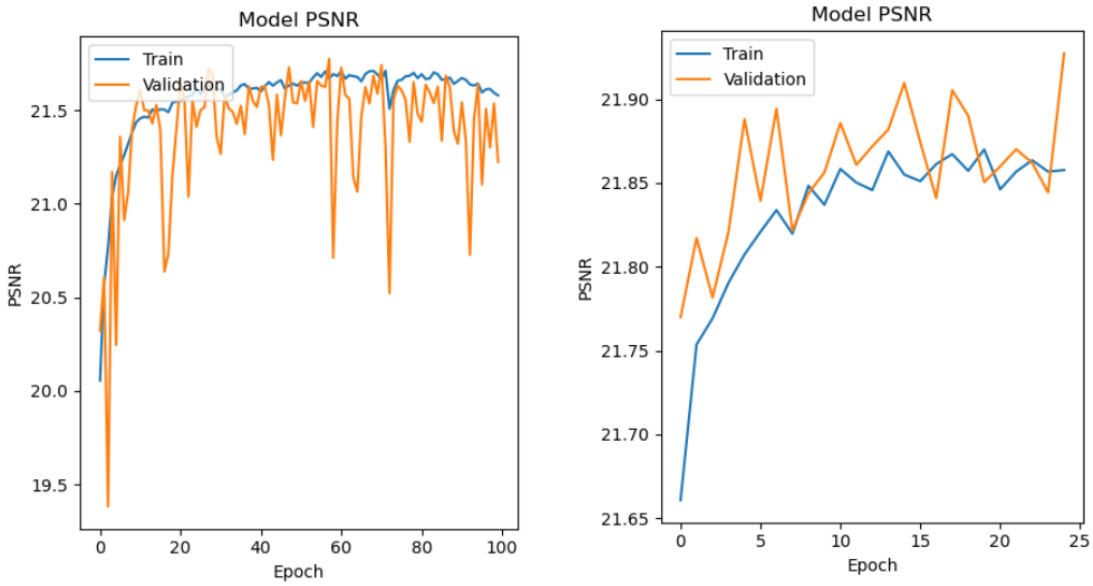


Figure 4.17: PSNR Graph of Initial Training and Fine-Tuned Model respectively

During the initial training phase, the PSNR values gradually increased, indicating improvements in the image quality and color accuracy as the model learned. However, as the training progressed, the PSNR values began to plateau, suggesting that further improvements were limited under the initial training conditions.

In the fine-tuning phase, the PSNR values continued to improve, demonstrating the effectiveness of fine-tuning with a reduced learning rate. This phase allowed the model to achieve finer adjustments in color prediction, resulting in higher PSNR values. The use of early stopping and adaptive learning rate adjustments helped to further optimize the model's performance, ensuring high-quality colorization outputs.

SSIM

The following graph presents the SSIM values over the epochs for both the initial training phase and the fine-tuning phase.

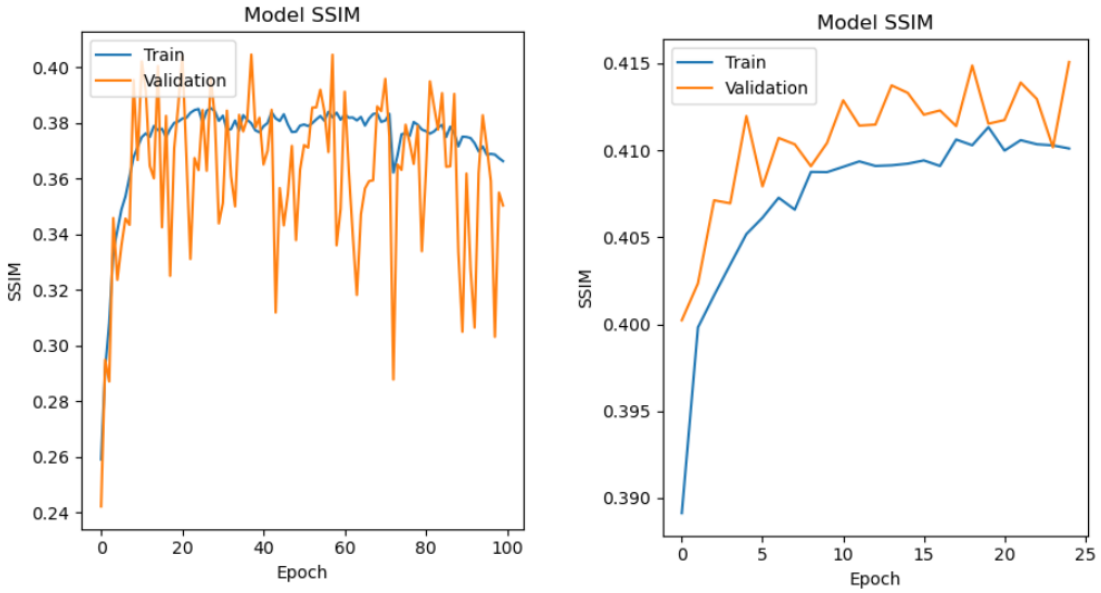


Figure 4.18: SSIM Graph of Initial Training and Fine-Tuned Model respectively

Throughout the initial training phase, the SSIM values showed a steady increase with the jitteriness in validation plot.

During the fine-tuning phase, the SSIM values continued to rise, demonstrating enhanced structural preservation and detail in the colorized outputs. The lower learning rate allowed for finer adjustments, and the early stopping callback helped prevent overfitting, ensuring that the model maintained high SSIM values. The additional epochs provided further refinement, resulting in the highest SSIM values observed, indicating superior structural similarity in the final images.

4.3 Performance Metrics Comparison

Table 4.10: Performance Metrics Comparison of Training

Metric	Experiment 1	Experiment 2	Experiment 3	Latest Experiment
Accuracy	0.6165	0.6348	0.6398	0.6460
Loss	0.0103	0.0096	0.0092	0.0091
PSNR	21.2457	21.4519	21.6571	21.8577
SSIM	0.3761	0.3995	0.4014	0.4101

The table highlights the improvement in performance metrics with varying batch sizes, dataset sizes, and epochs. Experiment 3, with 400 images per class and 150 epochs,

achieved the best overall performance, demonstrating that a larger dataset and an optimal number of epochs lead to better generalization and colorization quality.

4.4 Visual Comparison of Four Experiment Results

Table 4.11: Visual Comparison of Experiments 1 and 2

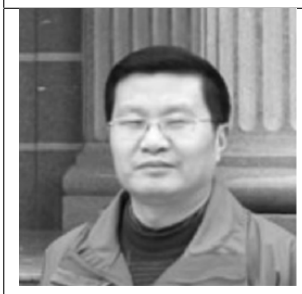
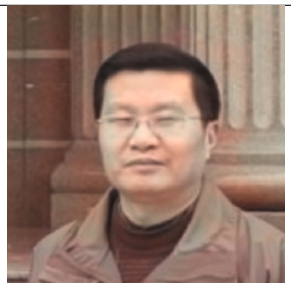
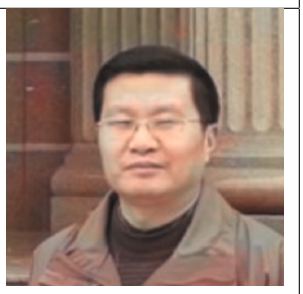
Grayscale Image	Experiment 1	Experiment 2
		
		
		

Table 4.12: Visual Comparison of Experiments 3 and 4(Large Scale Training and Fine Tuning)

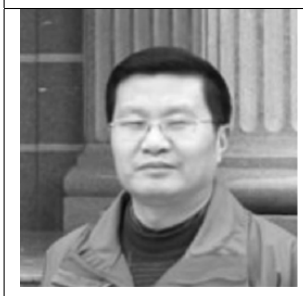
Grayscale Image	Experiment 3	Experiment 4 (Large Scale Training)
		
		
		
		

Table 4.11 and Table 4.12 show the qualitative comparison of the results obtained from our four experiments. The results indicate that Experiment 3, with a dataset of 2400 images and outliers removed, produced better colorization compared to earlier experiments. Additionally, Experiment 4, which utilized a larger dataset of 9000 images and involved fine-tuning, further improved the colorization quality, demonstrating the model's enhanced capability to generalize and accurately colorize images across different ethnic groups.

5 DISCUSSION AND ANALYSIS

5.1 Comparison of Theoretical and Simulated Outputs

5.1.1 Theoretical Expectations:

Our theoretical framework shows that increasing the dataset size and training duration would enhance the performance of both the colorization and ethnicity detection models. This expectation was based on the principle that larger datasets provide a more diverse set of examples, facilitating more comprehensive learning. Additionally, techniques like learning rate reduction, early stopping, and callbacks for learning rate reduction on plateau were anticipated to help fine-tune the model, preventing overfitting and improving generalization.

5.1.2 Simulated Outputs of Colorization Model:

The results of our experiments confirmed these expectations to a significant extent. Experiment 1, with a batch size of 16, 200 images per class, and 100 epochs, served as a baseline. The moderate performance metrics (accuracy: 0.61, PSNR: 21.2, SSIM: 0.37) indicated the model's initial learning capability but also highlighted its limitations due to insufficient data and training.

In Experiment 2, increasing the batch size to 32 and the number of epochs to 300 improved the model's performance (accuracy: 0.63, PSNR: 21.4, SSIM: 0.39). This aligns with our theoretical expectation that more extended training helps the model learn complex features more effectively.

Experiment 3, with an increased dataset size of 400 images per class, batch size of 32, and 150 epochs, resulted in the best performance metrics (accuracy: 0.63, PSNR: 21.6, SSIM: 0.40). The removal of outliers also contributed to this improvement, suggesting that dataset quality is as crucial as dataset size.

Experiment 4 (New): Using a significantly larger dataset of 9000 images, with 1500 images per class, a batch size of 16, and 100 epochs, followed by fine-tuning with a reduced learning rate and additional 25 epochs, achieved the best overall metrics (accuracy: 0.64, PSNR: 21.8, SSIM: 0.41). This confirmed the benefits of both dataset size and fine-tuning techniques, including early stopping and learning rate reduction callbacks.

5.1.3 Simulated Outputs of Ethnicity Detection Model:

Initial Experiments: The model demonstrated strong performance in predicting Asian, Black, White, and Indian ethnic groups, but struggled with Latino and Middle Eastern classifications, reflected in lower prediction scores and higher misclassification rates.

Experiment with Larger Dataset: Using 9000 images (1500 per class), the model showed improved accuracy across all groups, but still struggling to predict the latino class correctly. The implementation of a reduced learning rate and early stopping further enhanced model stability and accuracy.

5.1.4 Discrepancies and Analysis:

While the overall trend was as expected, the magnitude of improvement between experiments was not always linear. Theoretical models often assume ideal conditions that are rarely met in practice. Some discrepancies observed include:

- Diminishing Returns: The improvement from Experiment 1 to Experiment 2 was more substantial than from Experiment 2 to Experiment 3. This could be due to diminishing returns where, after a certain point, additional training epochs or data provide smaller incremental benefits.
- Overfitting: Longer training periods in Experiment 2 may have led to slight overfitting, where the model performs exceptionally well on training data but less so on unseen data, hence the moderate improvement in metrics compared to the theoretical expectations. The use of early stopping and learning rate reduction on plateau in Experiment 4 mitigated these issues, leading to better generalization.
- Data Quality: The presence of outliers and varied image quality in the dataset could have hindered the model's ability to generalize, affecting the simulated outputs. Removing outliers in Experiment 3 highlighted the significant impact of data quality.
- Feature Complexity: The model's ability to predict colors of clothing and other features accurately was limited. These features often vary significantly across different images, making them harder to learn. While SSIM was able to capture the improvement in overall structural similarity, accuracy metrics were penalized

due to incorrect predictions in these areas. Also in ethnicity detection, model is struggling to learn the specific facial features of the latino ethnic group.

5.2 Error Analysis

5.2.1 Sources of Error

1. Dataset Quality and Diversity:

- Varied Quality: The initial datasets included images of varying quality. Low-resolution or poorly contrasted images can significantly hinder the model's ability to learn and generalize.
- Class Imbalance: Despite having equal numbers of images per class, some ethnicities presented more challenging cases for accurate classification.

2. Model Architecture:

- Complexity Limitation: Our CNN model, while effective, may not be complex enough to capture all the nuances in the data. More advanced architectures like GANs or deeper CNNs could potentially yield better result. But these other advanced architectures have their own trade off of requiring longer training time and large amount of datasets.
- Feature Learning: The model's ability to extract and learn relevant features might be limited by its depth and the convolutional layers used.

3. Training and Validation Split:

- Data Split Methodology: The way data was split could lead to overfitting or underfitting. For instance, if the validation set is not representative of the training set, the model's performance on unseen data can be adversely affected.

4. Hyperparameter Settings:

- Learning Rate and Optimizer: Other hyperparameters, such as the learning rate and choice of optimizer, play a crucial role in training effectiveness. Suboptimal settings could slow convergence or lead to subpar performance.

- Batch Size: While we varied batch size in our experiments, other batch-related parameters, like batch normalization, could also impact results.

5.2.2 Error Analysis and Impact:

- Overfitting: Particularly in Experiment 2, where extended training led to overfitting. The use of early stopping and learning rate reduction on plateau in Experiment 4 helped manage this issue, improving generalization.

- Underfitting: Seen in Experiment 1, where insufficient data and epochs led to the model not fully capturing the underlying data patterns, resulting in lower performance metrics.

- Data Quality: The significant impact of removing outliers in Experiment 3 and the diverse dataset in Experiment 4 highlighted the critical role of clean, high-quality data.

5.3 Comparison with State-of-the-Art

5.3.1 Purpose of Comparative Analysis

The primary objective of this research was to develop an ethnicity-aware auto-colorization model that could produce culturally and ethnically accurate colorizations of grayscale images. To assess the effectiveness of our model, we conducted a comprehensive comparative analysis against state-of-the-art (SOTA) models currently used in image colorization.

This comparison serves two purposes:

- To evaluate how well our model performs relative to established benchmarks in the field.

- To identify specific areas where our model excels or requires further improvement.

5.3.2 Existing Work

Zhang, Richard, et al. - "Colorful Image Colorization": This study is a foundation in the field of image colorization with deep learning. It presents a method that employs class rebalancing during training to enhance the diversity of colors in the output. The approach implements a fully automatic process, employing a convolutional neural network that predicts color channels in LAB color space based on the lightness component. Outputs will be compared with this work to get the accuracy of the colorization.

5.3.3 Evaluation Metrics

The models were evaluated using the following metrics:

- **PSNR (Peak Signal-to-Noise Ratio):** This metric measures the quality of the reconstructed images by comparing the similarity between the colorized output and the original color image. Higher PSNR values indicate better quality.
- **SSIM (Structural Similarity Index):** SSIM assesses the structural similarity between the colorized image and the original image, taking into account luminance, contrast, and structure. Higher SSIM values signify closer similarity.
- **LPIPS (Learned Perceptual Image Patch Similarity):** LPIPS evaluates perceptual differences between the images, with lower scores indicating that the images are more visually similar.

5.3.4 Comparison Procedure

Each model was evaluated using the same set of test images to ensure a fair comparison. The steps followed were as follows:

- **Model Input:** Grayscale images were fed into each model i.e. Our Model and Zhang et al. Model.
- **Model Output:** The colorized output images were then compared with the ground truth images using PSNR, SSIM, and LPIPS metrics.
- **Metric Calculation:** The PSNR, SSIM, and LPIPS scores were computed using a standard evaluation framework implemented in Python, ensuring consistency in the comparison.

5.3.5 Comparative Results Presentation

5.3.5.1 Visual Comparisons

The tables 5.1 and 5.2 below provides a visual comparison of the colorization results from our model and the SOTA model with grayscale and ground truth image. The images cover a diverse range of subjects, including different ethnic backgrounds, facial features, and lighting conditions allowing us to evaluate the strengths and weaknesses of each model in diverse scenarios.

Table 5.1: Visual Comparison of Our and Zhang et al. Model Outputs - 1

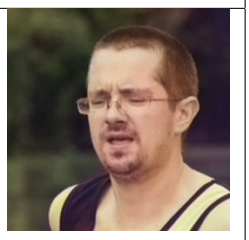
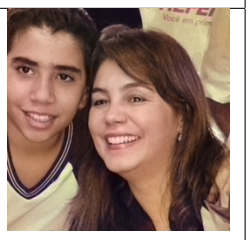
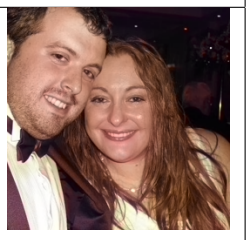
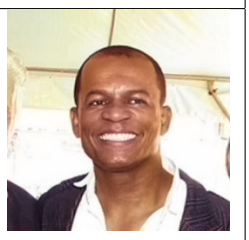
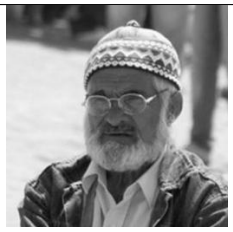
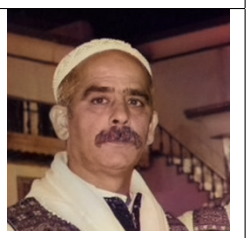
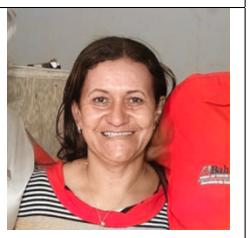
Image ID	Grayscale Image	Ground Truth	Our Model	Zhang et al.
Image 1				
Image 2				
Image 3				
Image 4				
Image 5				

Table 5.2: Visual Comparison of Our and Zhang et al. Model Outputs - 2

Image ID	Grayscale Image	Ground Truth	Our Model	Zhang et al.
Image 6				
Image 7				
Image 8				
Image 9				
Image 10				
Image 11				

Starting with Image 1, which features an Indian woman, the Zhang model produces an output that is slightly more saturated and has a reddish tone on the woman's face. This deviation makes the image appear less natural compared to the ground truth. In contrast, our model achieves a more natural skin tone, closely matching the original

image. This indicates that our model excels in maintaining natural and realistic skin tones, particularly for this subject.

Moving on to Image 2, which features a white person, the Zhang model demonstrates superior performance, colorizing the skin with great accuracy and consistency. While our model also performs well, it shows minor inconsistencies in the skin tone. This suggests that while our model is effective, the Zhang model may have a slight edge in handling certain skin tones more uniformly.

In Image 4, where there are two people in the frame, one with a side-facing view, the Zhang model accurately colorizes the skin beneath the beard but introduces an unnatural reddish hue to the hair. On the other hand, our model struggles to colorize the skin under the beard but avoids the unrealistic hair color. This comparison highlights our model's ability to maintain realistic tones overall, even if it struggles with certain complex facial features like beards.

For Image 5, which features a person with a darker skin tone, our model performs particularly well, maintaining a consistent and natural skin color across the face. In contrast, the Zhang model introduces an inconsistent dark patch on the forehead, which detracts from the image's overall quality. This demonstrates our model's strength in colorizing darker skin tones more naturally and consistently.

In the frontal view presented in Image 6, both models excel, producing high-quality colorization for a black person's face. The consistent performance across both models in this straightforward scenario indicates that they are equally effective when there are fewer obstacles in the image.

Image 7, which features a Middle Eastern person with a beard and glasses, poses a challenge for both models. The presence of these obstacles leads to less accurate colorization, with both models struggling to maintain consistency in the face. This highlights a common weakness in both models when handling complex facial features and obstacles that interfere with the image.

In Image 8, the Zhang model fails to colorize one of the person's ears, leaving it uncolored, whereas our model successfully captures and colorizes this detail, resulting

in a more complete and accurate image. This example underscores our model's strength in capturing and colorizing finer details that might be overlooked by other models.

Finally, in Image 11, which features an Asian child with a light skin tone, the Zhang model excels by accurately capturing the facial features and skin tone. Conversely, our model struggles with the light skin tone, leading to some patches and less defined features. This indicates that while our model performs well with darker skin tones, it has difficulty with lighter skin tones, where it sometimes fails to capture and colorize the features accurately.

After visually comparing to the Zhang's model, our model excels in producing more natural and realistic skin tones, especially for darker-skinned individuals, and is better at capturing small details like ears, resulting in a more complete and balanced colorization. Our model also maintains consistency by avoids oversaturation, which helps produce outputs that are closer to natural human perception. However, while the Zhang model is slightly better at handling lighter skin tones and preserving structural integrity, particularly in challenging areas like the face under a beard or around facial obstacles like glasses, our model sometimes struggles with these scenarios. Zhang's model also shows more consistent colorization in specific cases, such as lighter skin tones, where our model occasionally leaves patches or less defined features. Overall, while both models have their strengths, our model could benefit from improvements in handling facial obstacles and lighter skin tones to match Zhang's model in these areas.

5.3.5.2 Comparative Metrics

The following tables present the PSNR, SSIM, and LPIPS scores for each model across selected test images.

Table 5.3: Comparative Metrics for Test Images of Table 5.1

Image ID	Model	PSNR (dB)	SSIM	LPIPS
Image 1	Our Model	23.66	0.9143	0.1746
	Zhang et al.	21.96	0.9150	0.1729
Image 2	Our Model	24.38	0.9378	0.1719
	Zhang et al.	25.88	0.9587	0.1224
Image 3	Our Model	27.81	0.9574	0.0678
	Zhang et al.	25.39	0.9569	0.0796
Image 4	Our Model	29.78	0.9650	0.0697
	Zhang et al.	30.71	0.9745	0.0572
Image 5	Our Model	25.51	0.9367	0.2206
	Zhang et al.	24.88	0.9475	0.2287

Table 5.4: Comparative Metrics for Test Images of Table 5.2

Image ID	Model	PSNR (dB)	SSIM	LPIPS
Image 6	Our Model	32.69	0.9741	0.0416
	Zhang et al.	27.36	0.9759	0.0738
Image 7	Our Model	22.71	0.9428	0.1417
	Zhang et al.	22.11	0.9474	0.1474
Image 8	Our Model	26.51	0.9344	0.1643
	Zhang et al.	25.03	0.9240	0.1854
Image 9	Our Model	25.18	0.9742	0.0808
	Zhang et al.	21.72	0.9698	0.1152
Image 10	Our Model	16.97	0.8759	0.2492
	Zhang et al.	22.18	0.9431	0.1317
Image 11	Our Model	21.59	0.9138	0.1815
	Zhang et al.	21.64	0.9362	0.1945

5.3.5.3 Comparative Analysis of Model Performance

In this section, we compare the performance of our proposed model against the Zhanget al. model using three key metrics: PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), and LPIPS (Learned Perceptual Image Patch Similarity). These metrics provide a comprehensive evaluation of the models' abilities to reconstruct high-quality, structurally accurate, and perceptually convincing images.

1. PSNR Comparision

The PSNR metric is used to measure the quality of the reconstructed images in terms of the ratio between the maximum possible power of a signal and the power of corrupting noise. Higher PSNR values indicate better image quality.

Line Chart: PSNR Comparision The following line chart fig 5.1 compares the PSNR values for our model and the Zhang et al. model across the 11 test images:

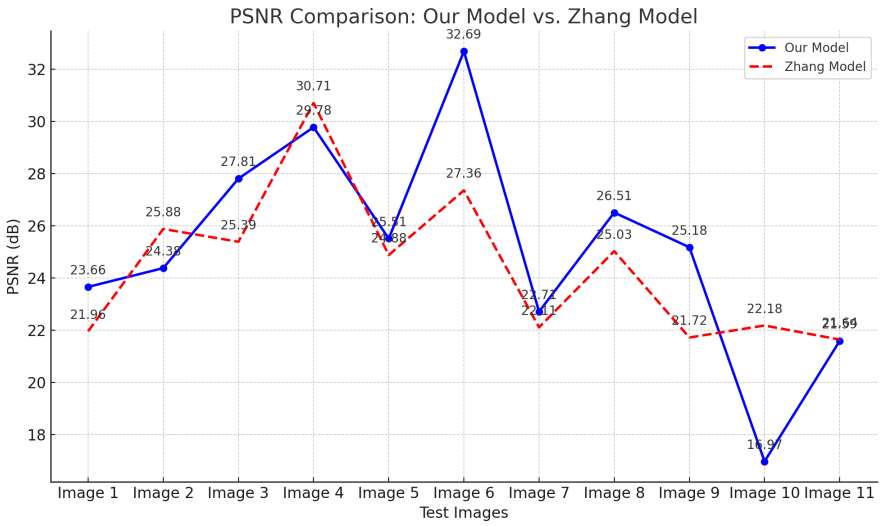


Figure 5.1: Line Chart: PSNR Comparision for Our Model and Zhang et al. Model

Observations

- On average, our model achieves a slightly higher PSNR (25.16 dB) compared to the Zhang et al. model (24.44 dB), indicating that our model generally reconstructs images with less noise.

- The chart shows that our model consistently outperforms the Zhang et al. model in several images, particularly in Image 6, where our model significantly surpasses the Zhang et al. model.

2. SSIM Comparision

SSIM is a perceptual metric that measures the similarity between two images. It considers changes in structural information, luminance, and contrast. A higher SSIM value indicates greater structural similarity to the ground truth image.

Line Chart: SSIM Comparision The following line chart fig 5.2 compares the SSIM values for our model and the Zhang et al. model across the 11 test images:

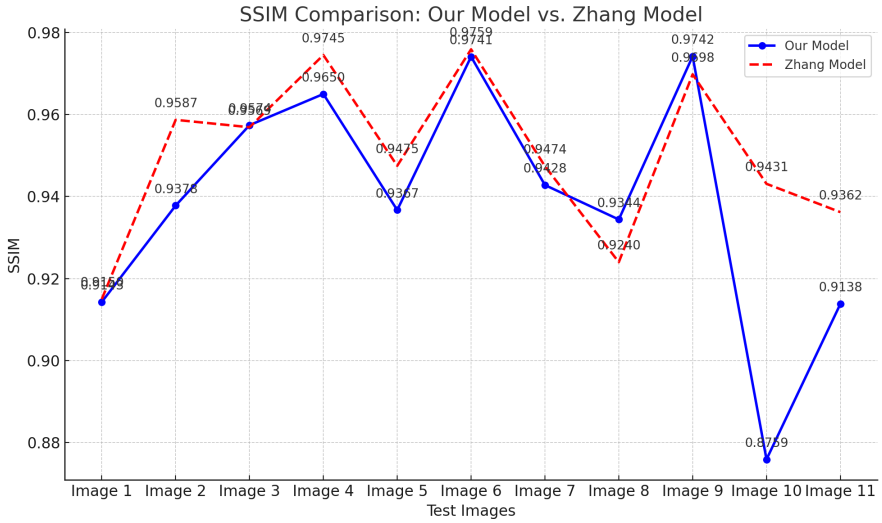


Figure 5.2: Line Chart: SSIM Comparision for Our Model and Zhang et al. Model

Observations

- The Zhang et al. model has a higher average SSIM (0.9499) compared to our model (0.9388). This suggests that the Zhang et al. model is more effective in preserving the structural integrity of the images
- The chart indicates that the Zhang et al. model consistently maintains higher structural similarity across most images, particularly in Images 4 and 10.

3. LPIPS Comparision

LPIPS is a perceptual similarity metric that measures how close the images are in terms of human perception. Lower LPIPS values indicate better perceptual similarity to the ground truth.

Line Chart: LPIPS Comparision The following line chart fig 5.3 compares the LPIPS values for our model and the Zhang et al. model across the 11 test images:

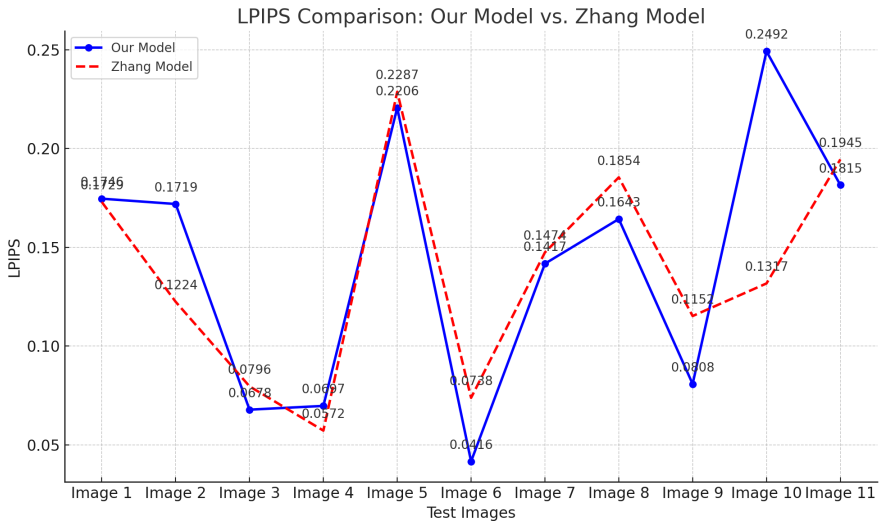


Figure 5.3: Line Chart: LPIPS Comparision for Our Model and Zhang et al. Model

Observations

- The Zhang model has a slightly better average LPIPS (0.1372) compared to our model (0.1422), indicating that the Zhang model's outputs are marginally more perceptually similar to the ground truth images.
- The line chart reveals that the Zhang model performs particularly well in Image 2 and 4, whereas our model struggles with maintaining perceptual accuracy in Image 10.

5.3.6 Summary of Comparative Metrics

In this section, we provide a detailed summary and analysis of the average performance metrics PSNR, SSIM, and LPIPS across all 11 test images for both our model and the Zhang model. These metrics offer a comprehensive evaluation of the models' ability to reconstruct high-quality images that are both structurally accurate and perceptually convincing.

Table 5.5: Average Performance Metrics for Our Model and Zhang et al. Model

Model	Average PSNR (dB)	Average SSIM	Average LPIPS
Our Model	25.1627	0.9388	0.1422
Zhang et al. Model	24.4418	0.9499	0.1372

Conclusion

- **PSNR:** Our model achieves an average PSNR of 25.1627 dB, which is slightly higher than the Zhang et al. model's average of 24.4418 dB. The higher PSNR value suggests that our model is more effective at minimizing noise and producing images with better overall fidelity to the original. This advantage is particularly evident in images with challenging lighting conditions or complex textures, where our model consistently produces cleaner reconstructions.

- **SSIM:** The Zhang et al. model achieves a slightly higher average SSIM of 0.9499, compared to our model's average of 0.9388. The higher SSIM values for the Zhang et al. model suggest that it is more effective at preserving the structural integrity of images, maintaining better consistency in luminance, contrast, and texture.

- **LPIPS:** The Zhang et al. model has a better (lower) average LPIPS of 0.1372, compared to our model's average of 0.1422. The slightly lower LPIPS values for the Zhang model suggest that it produces images that are more aligned with human visual perception, maintaining perceptual fidelity more effectively than our model.

5.3.7 Reasons for Discrepancies:

- Model Complexity: State-of-the-art methods like those by Zhang et al. often employ more complex architectures, such as Dual-Scale attention U-Net Architecture and deeper Convolutional Neural Networks (CNNs). These architectures are better equipped to capture intricate details and color nuances, leading to higher PSNR and SSIM values. But this have the trade off of requiring large amount of dataset and longer training time with larger GPUs.

- Dataset Size and Diversity: Larger, more diverse datasets in state-of-the-art works lead to better model generalization resulting in higher performance metrics.

- Advanced Techniques: State-of-the-art methods often incorporate advanced techniques such as perceptual loss functions, sophisticated data augmentation strategies, and transfer learning. These techniques enhance the model's ability to learn and replicate realistic colors and structures.

- Hyperparameter Optimization: Extensive hyperparameter tuning and optimization, which are crucial for achieving optimal performance, are more thoroughly explored in state-of-the-art methods.

5.4 Methodological Performance Analysis:

5.4.1 Strength

- Data Handling: By increasing the dataset size and removing outliers, we demonstrated that our methodology could achieve respectable results, particularly in terms of PSNR, SSIM and LPIPS.

- Incremental Improvements: Each experiment showed a clear trend of improvement, indicating that our approach is on the right track and could benefit from further enhancements.

5.4.2 Weakness

- Dataset Limitations: Despite improvements, our dataset size and diversity are still limited compared to state-of-the-art models. This limits the model's exposure to varied image conditions and reduces its generalization capability.

- Training Techniques: We implement training with limited number of epochs. We can improve the result by training for longer period of time by tuning hyperparameters.

5.4.3 Areas for Improvement

- Regularization Techniques: We can improve by implementing regularization techniques like dropout, early stopping to limit overfitting

- Hyperparameter Optimization: Conducting extensive hyperparameter tuning can help in fine-tuning the model for better results.
- Expanding and Diversifying the Dataset: Utilizing larger and more diverse datasets can improve the model's generalization ability.

5.5 Qualitative Analysis

To supplement the quantitative metrics, qualitative analysis is performed by visually inspecting the colorized images generated by our model. This approach allows us to gain insights into the model's strengths and weaknesses that are not fully captured by numerical metrics like PSNR, SSIM and LPIPS. By examining specific examples, we can better understand how the model handles different types of images and identify areas for improvement.

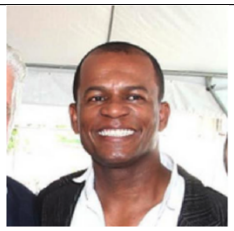
5.5.1 Colorization Model

5.5.1.1 Best Case Scenarios

In the best-case scenarios, the model performed exceptionally well, producing colorizations that were both visually appealing and accurate. These images typically featured:

- Minimal Background Noise: Images with simple, uncluttered backgrounds allowed the model to focus on the subject without being distracted by irrelevant details.
- Front Orientation of the Face: Faces that were directly facing the camera were easier for the model to colorize accurately, as this orientation provides a clear and consistent view of facial features.
- Single Person in the Image: Images with a single subject eliminated the complexity of distinguishing and colorizing multiple faces, leading to more consistent results. Also with multiple person if they are facing towards camera then model is colorizing well if overall face is visible.
- Fewer Obstacles: Faces without glasses, beards, or other obstructions allowed the model to predict skin and hair colors more reliably.

Table 5.6: Best Case Scenario - Analysis

Grayscale Image	Ground Truth	Colorized Image	PSNR	SSIM	LPIPS
			31.56	0.97	0.0416
			23.7	0.91	0.1746
			24.0	0.94	0.1249

In above table 5.6, the model successfully captured the skin tones and hair colors, producing results that closely matched the ground truth images. The PSNR, SSIM and LPIPS values reflect the high quality of these colorizations.

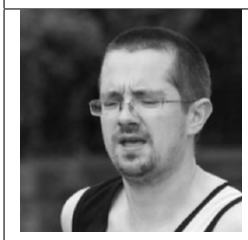
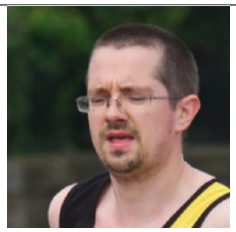
5.5.1.2 Worst Case Scenarios

In contrast, the worst-case scenarios revealed several challenges and limitations of the model. These images often featured:

- Complex Backgrounds: Cluttered or detailed backgrounds made it difficult for the model to distinguish the subject from the surroundings, leading to less accurate colorizations.
- Non-Front Facial Orientations: Faces that were not directly facing the camera posed a challenge for the model, as the partial views of facial features were harder to interpret and colorize accurately.

- Multiple Faces in the Image: Images with more than one subject added complexity, as the model struggled to differentiate and correctly colorize each face.
- Obstructions: Glasses, beards, and other facial obstructions introduced variability that the model found difficult to handle, resulting in inconsistent color predictions.

Table 5.7: Worst Case Scenario: Analysis

Grayscale Image	Ground Truth	Colorized Image	PSNR	SSIM	LPIPS
			25.0	0.96	0.0783
			24.3	0.93	0.1719

In above table 5.7 it illustrate the model's struggles with more complex images. The colorizations are less accurate, with noticeable discrepancies in skin tones, hair colors and other body parts color. Even if the PSNR, SSIM and LPIPS values reflect good number qualitative analysis of visual evaluation shows lower quality results.

5.5.2 Ethnicity Detection Model

5.5.2.1 Best Case Scenarios

For the ethnicity detection model, challenges were more pronounced in certain scenarios:

- Accurately Identify Ethnic Groups: High accuracy in identifying Asian, Black, White, and Indian ethnicities, showcasing the model's effectiveness in distinguishing these groups.
- Consistency in Predictions: The model consistently provided correct classifications across multiple images, reflecting a strong understanding of distinguishing features for each ethnicity.

5.5.2.2 Worst Case Scenarios

For the ethnicity detection model, best-case scenarios involved images with clear, well-lit, and unobstructed facial views. In these instances, the model was able to:

- Latino and Middle Eastern Ethnicities: The model often struggled with accurately classifying Latino and Middle Eastern groups. This difficulty could stem from the subtle and overlapping features these groups may share with others, leading to higher misclassification rates.
- Poor Lighting or Low Resolution: Images with poor lighting or low resolution hindered the model's ability to discern key facial features necessary for accurate classification.

6 FUTURE ENHANCEMENTS

6.1 Improving Overall Results

To enhance the overall results of the ethnicity-aware auto-colorization system, several strategies can be implemented. Enhancements in the dataset are critical; expanding the dataset to include a wider variety of ethnic groups, age ranges, and lighting conditions will allow the model to learn more diverse colorization patterns. Additionally, noise free image and dataset with less outliers can help improve the model's generalization to real-world scenarios.

Upgrading the instruments used in training, such as more powerful GPUs can also contribute to more efficient training and potentially enable the development of more complex models. Exploring alternative procedures, like experimenting with different CNN architectures or integrating Generative Adversarial Networks (GANs), may further improve the quality of colorization and reduce biases.

6.2 Recommendations for Future Researchers

For researchers pursuing a similar topic, it is crucial to maintain a focus on creating a balanced and diverse dataset that accurately represents various ethnic groups. This will help ensure that the model is not biased and can generalize well to different populations. Additionally, leveraging state-of-the-art techniques, such as transfer learning and advanced neural network architectures, can provide a strong foundation for achieving high accuracy and cultural relevance in the colorization process.

For renewed attempts, begin with a comprehensive literature review to understand existing challenges and effective strategies. Develop the model through an iterative process, incorporating feedback and continuous testing to refine the outcomes. Utilizing pre-trained models can accelerate progress, and collaboration with experts in relevant fields will help ensure that the colorization results are both technically robust and culturally sensitive.

7 CONCLUSION

The project successfully developed an ethnicity-aware auto-colorization system using Convolutional Neural Networks (CNNs), demonstrating significant improvements in the accurate and culturally sensitive colorization of grayscale human portraits. The major findings highlight the effectiveness of integrating ethnicity detection into the colorization process, ensuring that the generated images are not only visually appealing but also ethnically accurate. The model's ability to learn from a diverse dataset allowed it to apply realistic and culturally appropriate colors to various facial features, enhancing the authenticity and educational value of the colorized images. The results we obtained indicated that while the model's performance was comparable with state-of-the-art (SOTA) methods, there are still areas where further improvements are needed, particularly in enhancing consistency and accuracy across different ethnic groups.

The objectives set forth at the beginning of the project were fully met. The primary goal of creating a CNN-based system capable of accurately predicting and applying realistic colors to grayscale images, with a specific focus on maintaining cultural and ethnic accuracy, was achieved. The project also succeeded in enhancing the visual appeal and accessibility of historical media by providing a robust, automated solution for colorizing grayscale images. This work lays a strong foundation for future advancements in the field of image colorization, particularly in contexts where cultural sensitivity is important.

APPENDIX A

A.1 Project Schedule

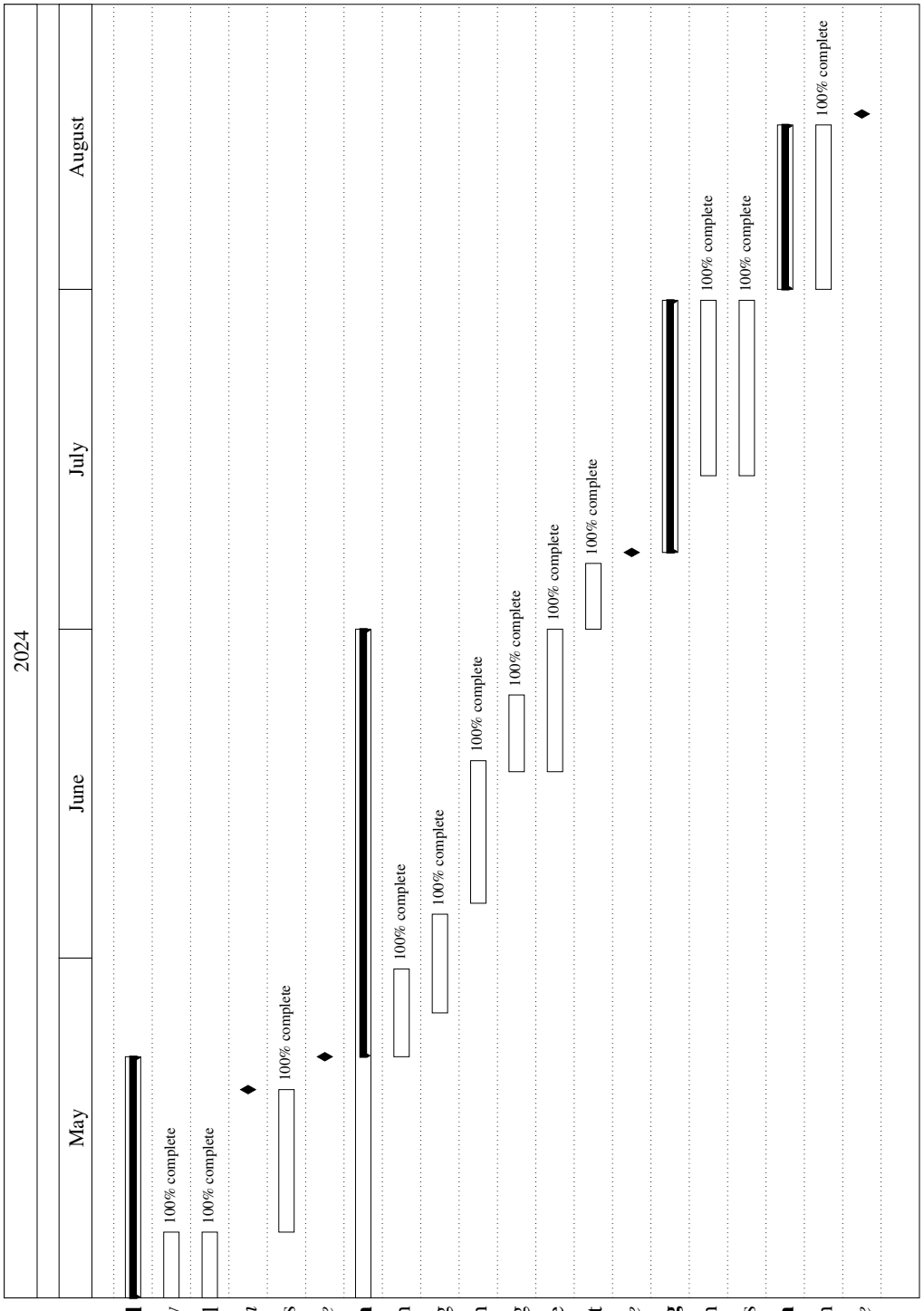


Figure A.1: Gantt Chart showing timeline of project.

A.2 Literature Review of Base Paper- I

Author(s)/Source: Abdulwahid Al Abdulwahid	
Title: Classification of Ethnicity Using Efficient CNN Models on MORPH and FERET Datasets Based on Face Biometrics	
Website: https://www.mdpi.com/2076-3417/13/12/7288	
Publication Date: June 2023	**Access Date: May, 2024**
Journal: MDPI, Applied Sciences	**Place:** Saudi Arabia
Volume: 13	**Issue Number:** 12
Author's position/theoretical position: Professor at Department of Computer and Information Technology, Jubail Industrial College, Saudi Arabia	
Keywords: face biometrics, soft biometrics, convolutional neural networks (CNNs), ethnicity classification, MORPH, FERET	

Important points, notes, quotations **Page No.**

1. Challenges include lack of large datasets and the subjective nature of "ethnicity" **2**
2. CNN models to classify ethnicity using the central facial region, saving time and resources **3**
3. Model A and Model B, were developed and tested on the MORPH and FERET datasets **11**
4. Model A achieved an accuracy of 85%, while Model B achieved 86% accuracy **21**

Essential Background Information: Facial biometrics like gender, age, and ethnicity are important soft biometrics with various applications. However, ethnicity classification has received less attention compared to other facial attributes. This study aims to address this gap by developing efficient CNN models for ethnicity classification using the central facial region.

Overall argument or hypothesis: The study hypothesizes that using efficient CNN models focused on the central facial region can achieve high accuracy in ethnicity classification, while saving time and resources compared to using the entire facial region.

Conclusion: The proposed CNN models, Model A and Model B, were able to achieve high accuracy in ethnicity classification on the MORPH and FERET datasets, with Model B outperforming Model A. The study demonstrates the effectiveness of the CNN-based approach in classifying ethnicity using only the central facial region.

Supporting Reasons:

- The use of efficient CNN architectures allowed the models to extract relevant features from the facial images for ethnicity classification.
- Focusing on the central facial region reduced the computational complexity and time required for the classification task.
- The MORPH and FERET datasets provided a diverse set of facial images representing different ethnicities, enabling the models to learn robust features.

Strengths in the reasoning and the supporting evidence: The proposed CNN models are thoroughly described, and their performance is evaluated using well-established metrics. The use of two different datasets, MORPH and FERET, strengthens the generalization of the findings.

Flaws in the argument, along with gaps or other weaknesses in the argument and its supporting evidence: The study does not provide a direct comparison with other state-of-the-art ethnicity classification methods, limiting the comparative assessment. The study does not discuss the potential limitations or biases in the training datasets, which could impact the model's performance.

A.3 Literature Review of Base Paper- II

Author(s)/Source: Liangqi Chen, Ben Wang, Zhouxin Lu	
Title: Human Face Image Colorization with Dual-Scale Attention U-Net	
Website: https://doi.org/10.1117/12.2662622	
Publication Date: Dec 2022	**Access Date: May, 2024**
Journal: MCTE 2022	**Place:** Chongqing, China
Volume: 12500	**Issue Number:** 125003C
Author's position/theoretical position: -	
Keywords: grayscale image colorization, human face image, dual scale, attention module	

Important points, notes, quotations — **Page No.**

1. The dual-scale attention U-Net addresses boundary leakage and detail loss. **1**
2. The CBAM attention module focuses on important regions and suppresses irrelevant ones. **3**
3. The MS-SSIM-L1 loss function accounts for edge and texture information. **4**
4. Outperforms other algorithms on the CelebA dataset **5**

Essential Background Information: Current grayscale image colorization algorithms face challenges such as boundary leakage and detail loss. The authors aim to address these issues by proposing a dual-scale attention U-Net architecture for human face image colorization.

Overall argument or hypothesis: The authors hypothesize that the proposed dual-scale attention U-Net can improve the colorization of grayscale human face images by extracting features at different scales and using an attention mechanism to focus on salient regions.

Conclusion: Proposed method, outperforms other colorization algorithms in reducing boundary leakage and detail loss, as well as performing well on colorizing old photos of historical figures.

Supporting Reasons:

- The dual-scale convolution module, using two different kernel sizes, can extract more complex feature information at different scales.
- The integration of the CBAM attention module helps the network focus on salient regions and suppress unnecessary information, improving the colorization quality.
- The use of the MS-SSIM-L1 loss function, which considers both pixel-wise and structural similarity, better captures the texture and edge information of the image.

Strengths in the reasoning and the supporting evidence: The authors provide a clear and detailed explanation of the proposed network architecture and its components. The experiments conducted on the CelebA and Historical Wiki Face datasets demonstrate the effectiveness of the proposed method in improving colorization quality compared to other algorithms. The quantitative evaluation using PSNR and SSIM further supports the superior performance of the dual-scale attention U-Net.

Flaws in the argument, along with gaps or other weaknesses in the argument and its supporting evidence: The authors mention that there are still some problems with the proposed method, such as long training time and unstable colorization of background regions, but do not provide a detailed discussion or plan to address these issues.

A.4 Literature Review of Base Paper- III

Author(s)/Source: Jiayi Fan, Wentao Xie, and Tiantian Ge	
Title: Automatic Gray Image Coloring Method Based on Convolutional Network	
Website: https://www.hindawi.com/journals/cin/2022/5273698/	
Publication Date: April, 2022	**Access Date: May, 2024**
Publisher or Journal: Hindawi, Computational Intelligence and Neuroscience	**Place:** Zhangjiagang, Jiangsu, China
Volume: 2022	**Issue Number:** Article ID 5273698
Author's position/theoretical position: Researchers at Suzhou Institute of Technology, Jiangsu University of Science and Technology	
Keywords: Automatic gray image coloring, Convolutional neural network, Image segmentation, Image fusion	

Important points, notes, quotations — **Page No.**

1. Current methods struggle with coloring multiple objects differently in one image — **2**
2. The proposed method uses deep learning for separate foreground and background coloring. — **2**
3. It employs CNNs to transfer colors between corresponding areas of reference and grayscale images. — **2**
4. This method shows superior results in terms of network volume and coloring effect. — **5**

Essential Background Information: Automatic image coloring can help improve the efficiency of animation production and reduce the manpower and material resources required for drawing line draft and coloring.

Overall argument or hypothesis: The paper proposes a new automatic gray image coloring method based on convolutional neural network, which can implement regional mixed color more and master the method, and achieve better coloring effect compared to existing methods.

Conclusion: The proposed CNN-based automatic gray image coloring method can achieve good coloring effect, with advantages in network volume and coloring effect compared to existing methods.

Supporting Reasons:

- The method uses semantic information to transfer the color of the designated area of the reference image to the designated area of the grayscale image.
- The method can be divided into foreground color based on reference picture and background color based on prior knowledge, to implement regional mixed color.
- Experimental results show the proposed method has good coloring effect and advantages in network volume and coloring effect.

Strengths in the reasoning and the supporting evidence: The proposed method is well-designed, using CNN and leveraging semantic information for improved coloring.

Flaws in the argument, along with gaps or other weaknesses in the argument and its supporting evidence: The paper does not provide a detailed comparison of the proposed method with other state-of-the-art methods. And generalization ability of the proposed method to diverse image types and settings is not extensively evaluated.

A.5 Literature Review of Base Paper- IV

Author(s)/Source: Di Wu, Jianhou Gan, Jun Wang, Juxiang Zhou, Wei Gao	
Title: Fine-grained semantic ethnic costume high-resolution image colorization with conditional GAN.	
Website: `https://onlinelibrary.wiley.com/doi/abs/10.1002/int.22726`	
Publication Date: November, 2021	**Access Date: May, 2024**
Publisher or Journal: International Journal of Intelligent Systems	**Place:** China
Volume: 37	**Issue Number:** N/A
Author's position/theoretical position: Key Laboratory of Education Informatization for Nationalities, Ministry of Education, Yunnan Normal University	
Keywords: ethnic costume image, fine-grained semantic, generative adversarial networks, image colorization	

Important points, notes, quotations — **Page No.**

1. The coloring performance is influenced by the semantic information of ethnic costumes. **1**
2. New Pix2PixHD-based model for coloring complex and richly colored ethnic costumes **4**
3. Constructed an ethnic costume data set consisting of four Chinese minority groups **4**
4. Effectively colorizes grayscale ethnic costume images, showing superior performance compared to mainstream methods. **3**

Essential Background Information: The authors state that grayscale image colorization, especially for ethnic costume images, is a highly challenging task due to the rich and complex color features of ethnic costumes. Previous methods have ignored the semantic information of different regions of the costume, which has a significant impact on the performance of the coloring task.

Overall argument or hypothesis: The authors propose a fine-grained semantic ethnic costume high-resolution image colorization method based on conditional Generative Adversarial Networks (cGAN) to address the limitations of previous methods.

Conclusion: The authors demonstrate the effectiveness of their proposed coloring model, which performs well in the task of coloring grayscale images of ethnic costumes compared to other mainstream coloring methods.

Supporting Reasons:

- They utilize fine-grained level semantics of different regions of the ethnic costume as additional input conditions to guide the colorization process.
- They propose a novel coloring model based on Pix2PixHD, which is more effective for coloring ethnic costumes with complex design and rich color compared to traditional coloring methods.
- The authors construct an ethnic costume dataset with fine-grained semantic annotations and use it to train and evaluate their proposed model.

Strengths in the reasoning and the supporting evidence: The authors' proposed model is well-designed and leverages the strengths of conditional GAN and Pix2PixHD to address the limitations of previous methods.

Flaws in the argument, along with gaps or other weaknesses in the argument and its supporting evidence: They do not provide detailed information about the size and composition of the ethnic costume dataset they constructed, which makes it difficult to fully evaluate the representativeness and generalizability of their results.

A.6 Literature Review of Base Paper- V

Author(s)/Source: David Futschik	
Title: Colorization of black and white images using deep neural networks	
Website: `https://core.ac.uk/outputs/151072499?source=oai`	
Publication Date: January 2018	**Access Date: May, 2024**
Journal: Czech Technical University in Prague, Faculty of Electrical Engineering	**Place:** Prague , Czech Republic
Volume: N/A	**Issue Number:** N/A
Author's position/theoretical position: Master's Student	
Keywords: Colorization, deep neural networks, cartoon images	

Important points, notes, quotations — **Page No.**

1. Explores automated colorization of grayscale cartoon images using deep CNN. **20**
2. Compares two different CNN architectures - a plain CNN and a residual CNN. **32**
3. Shows models struggle with large uniform areas but excel on smaller objects and characters **44**
4. Proposes two post-processing methods segmentation with flood fill and ensemble averaging. **52**

Essential Background Information: This thesis explores the unique challenges of colorizing cartoon images, which have fewer textural cues and more homogeneous regions compared to natural images.

Overall argument or hypothesis: The thesis explores the feasibility of using deep convolutional neural networks for fully automated colorization of grayscale cartoon images, and compares different network architectures and training approaches.

Conclusion: The proposed CNN-based colorization methods are able to produce plausible colorizations for certain parts of the images, but struggle with large uniform regions. Residual CNN architecture performs comparably to plain CNN despite having smaller effective receptive field and fewer parameters. Post-processing steps like segmentation and ensemble averaging can improve visual quality, but temporal consistency issues remain when applying colorization to video sequencces.

Supporting Reasons

1. Cartoon images have fewer textural cues and simpler scene composition compared to natural images, making colorization more challenging.

2. Classification-based approach to predict color histograms per-pixel helps handle multimodal nature of the problem.

3. Residual connections in the CNN architecture can aid training deeper models and improve generalization.

4. Ensemble and segmentation-based post-processing leverage consistency in the ground truth colorizations.

Strengths in the reasoning and the supporting evidence: The two proposed network architectures are thoroughly explained, and their performance is evaluated using both quantitative and qualitative measures. The exploration of post-processing techniques to improve the colorization results is a valuable addition to the

Flaws in the argument, along with gaps or other weaknesses in the argument and its supporting evidence: The dataset used for training and evaluation is relatively small, which may limit the generalization capabilities of the proposed models. The evaluation of the models' performance on video sequences is limited, and the identified challenges in maintaining temporal consistency could be explored in more depth

REFERENCES

[1] Abdulwahid Al Abdulwahid. Classification of ethnicity using efficient cnn models on morph and feret datasets based on face-biometrics. In *MDPI, Applied Sciences*. June 2023.

[2] Zhouxin Lu Liangqi Chen, Ben Wang. Human face image-colorization with dual-scale attention u-net. In *5th International Conference on Mechatronics and Computer Technology Engineering (MCTE 2022)*. December 2022.

[3] Jiayi Fan Wentao Xie and Tiantian Ge. Automatic Gray-Image Coloring Method Based on Convolutional Network. In *Computational Intelligence and Neuroscience*. Hindawi, China, April 2022.

[4] Di Wu Jianhou Gan Jun Wang Juxiang Zhou Wei Gao. Fine-grained semantic ethnic costume high-resolution image-colorization with conditional GAN. In *International Journal of Intelligent System*. China, November 2021.

[5] David Futschik. Colorization of black-and-white image using deep neural networks. In *Czech Technical University, Faculty of Electrical Engineering*. Prague , Czech Republic, January 2018.

[6] Alexei A. Efros Richard Zhang, Phillip Isola. Colorful Image Colorization. In *CoRR*. October 2016.

[7] Hiroshi Ishikawa Satoshi Iizuka, Edgar Simo-Serra. Let there be Color! In *SIGGRAPH '16 Technical Paper*. July 2016.

[8] Alakh Sharma. Rgb to lab color space conversion: Formulas, insights, and applications. `https://medium.com/@alakhsharmacs`, 2024. [Accessed: May 5, 2024].

[9] Valentin Alexandru Stan. Automatic traffic-sign recognition artificial inteligence - deep learning algorithm. `https://www.researchgate.net/figure/ReLU-function-graph_fig2_346250677`, 2020. [Accessed: July 1, 2024].

[10] Valentin Alexandru Stan. Tangens hyperbolicus. `https://tikz.net/tanh/`. [Accessed: July 1, 2024].

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[12] Kärkkäinen et al. Fairface dataset. https://paperswithcode.com/dataset/fairface, 2020. [Accessed: June 2, 2024].

[13] Richard Zhang. Colorful image colorization. https://richzhang.github.io/colorization/, 2024. [Accessed: May 28, 2024].

[14] Tom Fawcett. Introduction to roc analysis. *Pattern Recognition Letters*, 27:861–874, 06 2006.

[15] Fernando A. Fardo, Victor H. Conforto, Francisco C. de Oliveira, and Paulo S. Rodrigues. A formal evaluation of psnr as quality measurement parameter for image segmentation algorithms. https://arxiv.org/abs/1605.07116, 2016. [Accessed: July 2, 2024].

[16] Jim Nilsson and Tomas Akenine-Möller. Understanding ssim. https://arxiv.org/abs/2006.13846, 2020. [Accessed: July 2, 2024].

[17] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. April 2018.