Department of Electrical and Computer Engineering
North South University

Senior Design Project

ENHANCING IMAGES FOR COLOR VISION DEFICIENCY (CVD) USING DEEP LEARNING

TANEEM AHMED	**ID# 2013102042**
SIDDHARTHA SANKAR SAHA	**ID# 2011567042**
MAHBUB MORSHED RIFAT	**ID# 2011415042**

Faculty Advisor:

Dr. Mohammad Ashrafuzzaman Khan (AzK),

Associate Professor

ECE Department

Spring, 2024

LETTER OF TRANSMITTAL

June 01, 2024

To

Dr. Rajesh Palit
Chairman,
Department of Electrical and Computer Engineering
North South University, Dhaka

Subject: **Submission of Capstone Project Report on "Enhancing Images for Color Vision Deficiency (CVD) using Deep Learning"**

Dear Sir,

With due respect, we would like to submit our **Capstone Project Report** on **"Enhancing Images for Color Vision Deficiency (CVD) using Deep Learning"** as a part of our BSc program. This report presents our original research on color-deficient observers using a deep-learning approach. The project was a valuable learning experience for us, as it allowed us to apply our knowledge and skills to a challenging problem in the real world. We made every effort to produce a high-quality report that meets all of the requirements.

We will be highly obliged if you kindly receive this report and provide your valuable judgment. It would be our immense pleasure if you find this report useful and informative to have an apparent perspective.

Sincerely Yours,

....

Taneem Ahmed
ECE Department
North South University, Bangladesh

....

Siddhartha Sankar Saha
ECE Department
North South University, Bangladesh

....

Mahbub Morshed Rifat
ECE Department
North South University, Bangladesh

APPROVAL

Taneem Ahmed (ID # 2013102042), Siddhartha Sankar Saha (ID # 2011567042) and Mahbub Morshed Rifat (ID # 2011415042) from Electrical and Computer Engineering Department of North South University, have worked on the Senior Design Project titled "**Enhancing Images for Color Vision Deficiency (CVD) using Deep Learning**" under the supervision of Dr. Mohammad Ashrafuzzaman Khan partial fulfillment of the requirement for the degree of Bachelors of Science in Engineering and has been accepted as satisfactory.

Supervisor's Signature

……………………………………….

Dr. Mohammad Ashrafuzzaman Khan

Associate Professor

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

Chairman's Signature

……………………………………….

Dr. Rajesh Palit

Professor

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

DECLARATION

This is to certify that the project titled "**Enhancing Images for Color Vision Deficiency (CVD) using Deep Learning**" is entirely our own work. We have not submitted any part of this work to any other institution, in whole or in part, for the award of any other degree or diploma. All project-related information will be kept confidential and will not be shared without the express permission of the project supervisor. All relevant previous works cited in this report have been duly acknowledged and cited. We have adhered to the plagiarism policy outlined by the supervisor.

Students’ names & Signatures

- -

1. Taneem Ahmed

- -

2. Siddhartha Sankar Saha

- -

3. Mahbub Morshed Rifat

ACKNOWLEDGEMENTS

The authors would like to express their heartfelt gratitude towards their project and research supervisor, Dr. Mohammad Ashrafuzzaman Khan, Associate Professor, Department of Electrical and Computer Engineering, North South University, Bangladesh, for his invaluable support, precise guidance and advice pertaining to the experiments, research and theoretical studies carried out during the course of the current project and also in the preparation of the current report.

Furthermore, the authors would like to thank the Department of Electrical and Computer Engineering, North South University, Bangladesh for facilitating the research. The authors would also like to thank their loved ones for their countless sacrifices and continual support.

ABSTRACT

Enhancing Images for Color Vision Deficiency (CVD) Using Deep Learning

In a stunning world full of colors, color vision deficiency is the most common difficulty every human being faces. This problem still has no treatment. The subject of this research is to expose artificial intelligence methods as profound studies that aim at material, tangible visualization through images of color blindness in humanity, hence becoming a hope that technology will eventually fill the gap in color perception. The two AI models are the basis for creating and utilizing the project.

The first model is Daltonization, which simulates the perception of color by different kinds of CVD and hence can create image transformations that specifically cater to one's eye problems. Secondly, A CNN-based Autoencoder model trains on various images taken under normal vision color conditions and color vision deficiency (CVD) conditions. Therefore, the aim is to create an autoencoder that will convert any image into a better-colored one that is clearly visible to individuals suffering from CVDs.

The triumph of this task is determined by how well it performs in terms of two criteria. Quantitative measurements allow for an analysis of the degree to which the latest picture reproduces original images through methods like SSIM (Structural similarity index). User testing and feedback obtained by people suffering from CVD constitute quality appraisals for verifying the acceptability of changes made to their visual perception.

We ensure that data collection and utilization are done per strict ethical principles to avoid violating privacy rights or obtaining participants' permission. The primary objective of using converted images is to adhere to and also maintain established accessibility standards for people with Color Vision Deficiency (CVD), therefore rendering them more manageable.

The project focuses on user feedback and iterative design methods aimed at guaranteeing that the produced tools are intuitive and user-friendly for persons with low vision (CVD). These AI models work well on web browsers, mobile devices, or picture-editing tools. This project looks not only at

improving the accessibility of images but also examines possible uses for the models in this field within augmented reality and virtual reality, amongst other emerging technologies. By improving how they see things while learning or working with it every day during their hours of relaxation at home, we hope that through our efforts, those suffering from CVD would experience an enhanced quality of life."

The exploration of AI will demonstrate how deep learning could come in handy in solving accessibility issues, hence pushing the field further. A user-friendly tool will be developed through this project, which should integrate its software or applications with existing platforms and workflow easily.

The project asserts higher levels of inclusivity in visual design by increasing awareness and promotion. This project combines leading-edge AI tech and user-centered design principles, potentially changing how people with color vision deficiencies see objects.

TABLE OF CONTENTS

Contents

LETTER OF TRANSMITTAL 2

APPROVAL 4

DECLARATION 5

ACKNOWLEDGEMENTS 6

ABSTRACT 7

TABLE OF CONTENTS 9

LIST OF FIGURES 11

LIST OF TABLES 12

Chapter 1 Introduction 13

1.1 Background and Motivation 13

1.2 Purpose and Goal of the Project 14

1.3 Organization of the Report 16

Chapter 2 Research Literature Review 17

2.1 Existing Research and Limitations 17

Chapter 3 Methodology 21

3.1 System Design 21

3.2 Hardware Components/Software Components 23

3.3 Implementation 24

Chapter 4 Investigation/Experiment, Result, Analysis and Discussion 30

Chapter 5 Impacts of the Project 37

5.1 Impact of this project on societal, health, safety, legal and cultural issues 37

5.2 Impact of this project on environment and sustainability 39

Chapter 6 Project Planning and Budget 41

Chapter 7 Complex Engineering Problems and Activities.. 42

7.1 Complex Engineering Problems (CEP).. 42

7.2 Complex Engineering Activities (CEA) 44

Chapter 8 Conclusion.. 45

8.1 Summary.. 45

8.2 Limitation 47

8.3 Future Improvement 47

References 48

LIST OF FIGURES

Figure 1: System Design	22
Figure 2: Data Preprocessing	24
Figure 3: Training and Validation Data Loader	25
Figure 4: Loss Function and Optimizer	25
Figure 5: CNN-Based Autoencoder Model with Skip Connections	27
Figure 6: Training Loop & Validation Loop	28
Figure 7: Protanopia Result (Visualized)	29
Figure 8: Deuteranopia Result (Visualized)	31
Figure 9: Original Image	31
Figure 10: Deuteranopia Simulation	32
Figure 11: Deuteranopia Daltonization	32
Figure 12: Deuteranopia Daltonization & Simulation	33
Figure 13: Loss Curve (1-MSSSIM)	33
Figure 14: SSIM Accuracy Curve	34
Figure 15: Gannt Chart	41
Figure 16. Budget table.	41

LIST OF TABLES

Table I. List of Software Tools	23
Table II. List of Hardware Tools	23
Table III. Complex Engineering Problem Attributes	44
Table IV. Complex Engineering Problem Activities	46

Chapter 1 Introduction

1.1 Background and Motivation

Color vision deficiency (CVD) has been a common worldwide phenomenon in recent years. Being a genetic disorder, it is increasing day by day. Currently, 300 million people are living with color vision problems. Everyone in twelve men (8%) and one in two hundred women (0.5%) have CVD.
In 1974, John Dalton first described the color blindness problem. He and his brother were also color-blind. They often confused red with green and pink with blue. John Dalton instructed to examine his eyes perfectly after his death. Later, DNA extraction of his preserved eye tissue revealed that he was deuteranope, which means lacking the middle wave photopigment of the retina. However, in the early 20th century, everyone started to pay attention.

The research aims to improve people's lives with color vision deficiency, which motivates us. Difficulty in distinguishing color hampers everyone's daily life. It pushes them to depend on others. They require assistance comprehending traffic lights but no information about color blindness. Generally, there exist three classes of color blindness.

Red-Green Color Blindness: This is the most commonly seen case where people cannot distinguish reds from greens.
Blue-Yellow Color Blindness: This type comes next after blue-yellow color blindness because it entails an inability to differentiate between blues and yellows.
Complete Color Blindness (**Achromatopsia):** Few individuals experience total achromatopsia despite its scarcity; hence, they see the world in gray color. [2]

Human color vision is accomplished through cone cells in the retina. A human with typical vision will have three types of cells sensitive to a different light wavelengths. The L–cones catch the Long–wavelength (~red), M–cones catch Medium wavelength (~green) and S–cones catch Short–wavelength (~blue). Their responsiveness of across the light spectrum is given below:

Cone-related CVD types:

- **Monochromacy**: This is the worst form of CVD. In this, only a single type of cone is working, or no cone is working at all. The people having monochromacy see a gray world.
- **Dichromacy**: It is more common than the previous one. A person will be having two types of cones. About these, there are several types of dichromacy according to which type of cone is missing:
    - **Protanopia**: Absence of L-type cone (red).
    - **Deuteranopia**: Absence of M-type cone sensitivity (green).
    - **Tritanopia**: Absence of S-type cone sensitivity (blue).
- **Anomalous Trichromacy**: This is the most common form of CVD; the sensitivity of one of the three normal cones is changed. Following are the types of anomalous trichromacy according to the changed cone:
    - **Protanomaly**: L-cone sensitivity is different.
    - **Deuteranomaly**: M-cone sensitivity is different.
    - **Tritanomaly**: S-cone sensitivity is different.

However, personal experiences, social impact, design challenges, technology advancements, awareness, educational value, entrepreneurial spirit, and social impact have driven us to choose color vision deficiency. It is a passion to find solutions and make a change due to the challenges faced by individuals with color vision deficiency. The potential for broad social impact and inventive solutions can result in a more inclusive society. Color blindness comes with unique design problems, which may motivate individuals who are into creativity in solving problems. Lessening the effect of CVD and supporting inclusiveness will reduce the stigmatization around it while fostering an all-encompassing society.[3]

1.2 Purpose and Goal of the Project

Daltonization is a technique that adjusts image colors to sustain people with color vision deficiency (CVD). This research aims to demonstrate AI methods like deep learning to make images more tangible to people with color vision deficiency. The goal of employing models like the neural network CNN-based Autoencoders model on daltonization is to create personalized image representations that cater to their specific visual needs. Our research paves the way for a future where technology bridges the gap in color perception.

The overarching aim of this project is to contribute to the enhancement of color vision deficiency (CVD) research and practice by addressing the following specific goals:

Dataset Collection: Gather diverse image datasets from Kaggle, hugging-face, and many websites with images like Unsplash and Pixels. The project used a customized dataset to mix all the collected images.

Neural Network Models: Different generative models were used, like CGAN, Stable Diffusion, VAE, and AE, to differentiate the daltonize image quality.

Performance Evaluation: Measure the performance of the developed model using the MSSIM loss function to find the training loss and validation loss. On the other hand, SSIM performance functions to find the training and validation performance.

Interpretability and Explainability: To improve the developed model's interpretability and transparency, apply explainable artificial intelligence approaches. This phase aims to illuminate the models' decision-making processes, enabling academics and professionals to comprehend and believe in the outcomes.

Implications and Applications: Discuss the impact of the research findings for color vision deficiency research and practice. The project aims to contribute to the development of personalized interventions and support systems for researching CVD treatments in the future.

The project's success will enhance the visualization power of people with CVD. They will start to live a fully colorful life. On the other hand, the CVD people will be independent. They will no longer need help selecting clothes, reading maps, or distinguishing traffic lights.[6]

1.3 Organization of the Report

This report aims to give a thorough overview of the research conducted on a system designed to modify and enhance image colors for individuals with color vision deficiency, specifically protanopia, deuteranopia, and tritanopia. The system uses a Deep Learning architecture like CNN-Based Autoencoder to help these individuals distinguish colors effectively. The report is organized as follows:

1) Chapter 1 introduces the topic of Color Vision Deficiency (CVD) covered in the report. This includes background information, objectives, scope, and an outline of the report's contents.

2) Chapter 2 reviews relevant literature, highlighting recent research and its limitations.

3) Chapter 3 outlines the methodology, describing the research methods, techniques, or approaches used in this project.

4) Chapter 4 presents the research findings or results and discusses them.

5) Chapter 5 explores the project's impacts on societal health, environment, and sustainability.

6) Chapter 6 details the project planning and budget.

7) Chapter 7 addresses complex engineering problems and activities related to the project.

8) Chapter 8 summarizes the main findings, key points, and references discussed in the project.

Chapter 2 Research Literature Review

2.1 Existing Research and Limitations

In recent years, researchers have made significant strides in developing automated systems for compensation and evaluation for color vision deficiency (CVD). These advancements aim to enhance the color perception and accessibility of digital images and videos for individuals with CVD. The literature reviews presented in this section aim to offer a comprehensive overview of the existing approaches, techniques, and advancements in the field of Daltonization.

2.1.1: A content-dependent Daltonization algorithm for color vision deficiencies based on lightness and chroma information

A team of researchers led by Xuming Shen at Hangzhou Dianzi University unveiled a noble content-dependent Daltonization algorithm to help people with color vision deficiencies better sense color through lightness and chroma information. The developed algorithm was focused on previous challenges with the need to increase contrast and, at the same time, ensure consistency in natural colors. Comparison with other methods proved that the algorithm provided better color consistency characteristics for those with CVD, which is of utmost significance in terms of color perception accuracy.

However, some of the limitations of the algorithm were also observed in the study. It was reported that lightness modifications may lead to unexpected color shifts, like a green pepper appearing as blue to a protanomal subject. Another thing observed was that, due to its dependence on clustering for the purpose of image processing, this algorithm would not be suitable for video applications because color discontinuities may result from the variability of the clustering. Although the algorithm is promising for the correction of color perception of CVD users, future studies should be conducted toward perfecting lightness modifications to avoid color changes and finding alternative image processing that would enable real-time video processing. In addition, this study has not touched upon image segmentation improvement, thereby opening a possible future study in extracting image regions, enhancing clustering efficiency, and identifying interactions of elements of perceptual learning for CVD users. [1]

2.1.2: Intelligent modification for the daltonization process of digitized paintings

Christos-Nikolaos Anagnostopoulos, George Tsekouras, Ioannis Anagnostopoulos, and Christos Kalloniatis proposed an intelligent image processing way of enhancing visualization of digital images to individuals with protanopia, a manifestation of color-blindness where red and green shades are confused. This procedure, known as daltonization, is meant to change colors to increase the perception of colors in views that are color deficient. This method changes the regular RGB color space into its dichromatic counterpart, mimicking protanopic vision using standard linear transformations. The algorithm's new technique distributes the information lost to the protanopes, improving their perception of color in digital pictures. An important aspect of the method is the RGB similarity checking module, where the color content of the original image is examined, and the adaptation parameters are chosen intelligently, avoiding trial-and-error adaptations.

The RGB similarity-checking module sets up a color tolerance threshold around every color. It is the error matrix that is changed by this algorithm to avoid color matching between the original and daltonized images in case a given color falls within this threshold of the original image color. For example, a color tolerance of 21 would consider all combinations of colors contained in C (100±10, 150±10, 130±10) as being close enough to and not considerably different from C2 (100, 150, 130).

One of the most serious drawbacks of the algorithm is its computational complexity; thus, each iteration requires 2.1 seconds in a 300x300 24-bit image on a Pentium IV 3.2 GHz processor. As a consequence, it is unsuitable for processing video images in real-time. While the principles of the algorithm can be adapted to all types of color vision deficiencies, the present implementation has been developed explicitly for protanopia. Future work could focus on optimizing the algorithm, increasing its efficiency to make it work in real-time and its extension for use in other color vision deficiencies. [7]

2.1.3: An Adaptive Fuzzy-Based System to Simulate, Quantify, and Compensate Color Blindness

Jinmi Lee and Wellington dos Santos (2017) reported on an adaptive fuzzy-based simulation tool designed to enhance color perception for users with deficiencies in color vision. The tool evaluates the kind and intensity of color blindness with a tailored version of the Ishihara test known as DaltonTest. It then applies adaptive filtering to enhance color perception in digital images. Two methods are proposed: Method A corrects the absolute color blindness and applies fuzzy rules for further adjustments. In parallel, Method B uses linear transformations for adaptive correction based on fuzzy degrees of protanomaly and deuteranomaly.

The tool was tested with color-blind and non-color-blind users. Preliminary study Four color-blind volunteers assessed 40 images, which were corrected using Method A, with and without histogram equalization. The results indicate that correction within the RGB domain with histogram equalization better understand the image. Moreover, correction without equalization produced fewer distortions of the original colors. For non-color-blind users Ten basic images were used to create simulations for protan and deuteran color blindness at different intensities 0%, 25%, 50%, 75% and 100%. The best performance was shown by Method B with histogram equalization with obtaining 47.3% of positive answers for protan color blindness.

The greatest limitation of this study is the relative absence from the literature of similar mathematical methods of adaptive color blindness compensation. In addition, scant real color blindness cases, especially those mild, hinder the development and verification of such tools. However, the promising results in this study point to the potential of fuzzy-based approaches to improve color perception for individuals with color vision deficiencies. Further research should be done on developing these methods and making them more applicable to other types of color blindness. [4]

2.1.4: Color vision deficiency datasets & recoloring evaluation using GANs

The Paper Introduces a new method that will help people with color vision deficiency perceive better colors. The dataset contains 5,699 images, all preprocessed and then recolored by using an optimization-based algorithm and a self-adapting rule-based algorithm. After the removal of duplicates and incorrectly recolored images, the dataset contains 1,859 pairs of original and recolored images from the optimization-based algorithm and 1,225 pairs from the self-adapting algorithm.

The effectiveness of the dataset is tested by training three well-known GAN networks for recoloring images towards CVD: pix2pix-GAN, Cycle-GAN, and Bicycle-GAN. The results indicate that thepix2pix-GAN outperforms the other two in qualitative and quantitative evaluations since it achieves the highest SSIM scores. On the other hand, Cycle-GAN performed the worst, and Bicycle-GAN introduced large distortions.

The study has demonstrated the potential of deep learning for image recolorization in CVD. Some limitations are acknowledged: first, the dataset is relatively small, consisting of only nine groups of color pairs. Further, the self-adapting recoloring algorithm, though quite effective, may change colors into something that CVD patients cannot accurately interpret. Moreover, the evaluation has been mainly done on a quantitative level by applying SSIM and PSNR, which cannot represent the whole perceptual experience of colorblind individuals. Future work may account for these limitations by gathering an expanded dataset, refining the recoloring algorithms, and involving more subjective evaluations.[5]

Chapter 3 Methodology

3.1 System Design

Designing an Image Daltonization System using CNN-Based Autoencoder neural network model

Problem Statement: The goal is to develop a system that can modify and enhance image colors for specific types (protanopia, deuteranopia, and tritanopia) of color-deficient observers using Deep learning architecture like CNN-Based Autoencoder to help them distinguish the colors effectively.

Data Collection: For this project, we used Landscape Pictures dataset from Kaggle comprising of 4,319 images of pictures of natural landscapes like mountains, deserts, seas, beaches, islands and cities, which has been created with 7 research from the website Flickr (https://www.flickr.com/)

Creating Custom Dataset: We used the Daltonization algorithm on the Kaggle dataset to make the three different types (protanopia, deuteranopia, and tritanopia) daltonized versions. Along with the 4319 original images, our custom dataset contains 4319 + 4319*3 = 17,276 images.

Data Preprocessing: Applying all preprocessing steps to the collected data: First, ensure that the input image is in torch.float32, the standard type for PyTorch tensors. Then, normalize pixel values to the range between 0 and 1, followed by rescaling by 0.5. Further, our transformation pipeline will resize the image to 256x256 pixels, transform it into a PyTorch tensor, and perform normalization again by subtracting the mean—0.5 for each color channel—and dividing by the standard deviation—0.5. Our standardized format is important for effective learning in most image-based deep-learning tasks.

Model Architecture: In generating daltonized images using deep learning, we have looked at Stable Diffusion models, GANs, Conditional GANs, and VAE. We select the most suitable architecture based on the visualization and SSIM accuracy curve balance for the final project. This chosen backbone architecture was based on a CNN-based Autoencoder with skip connections that would form the backbone for our daltonized generating model, ensuring accurate performance and practicality for deployment in health applications, especially on resource-constrained devices

Model Modification: We've customized our CNN-based Autoencoder architecture to align with the demands of creating high resolution of daltonized images, which involves skip connections to the network so that it enhances performance by directly transferring information between encoder and decoder layers. It helps to alleviate the vanishing gradients during training, reconstruct high-frequency details, and allows better hierarchical learning of the features.

Training: We have split the dataset into train and validation datasets as 80:20. The modified autoencoder model, based on types, is trained by the train set. In the training process, the weights of the model are optimized for minimizing the (1-MSSIM) score, that is, how dissimilar the generated and target color-blind images are, by using Adam optimization algorithms.

Model Evaluation: We have evaluated the trained model using the validation set. Measuring performance metrics such as SSIM (Structural Similarity Index Measure) to assess the model's performance.

Fine-tuning and Hyperparameter Tuning: we have fine-tuned the model by adjusting hyperparameters such as learning rate, batch size, and regularization techniques. This process helps further optimizing the model's performance.

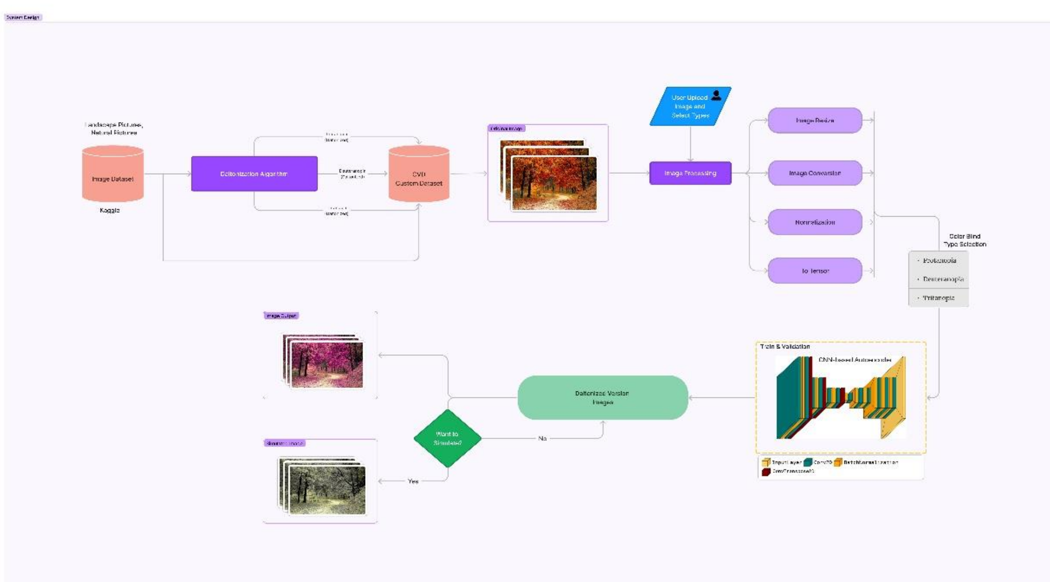

Figure 1: System Design

3.2 Hardware Components/Software Components

Dataset: https://www.kaggle.com/datasets/arnaud58/landscape-pictures

This Dataset has 4,319 landscape images.

Software Components:

Table I. List of Software Tools

Tools	Functions	Other Similar Tools (If any)	Why selected this tool?
Python	For Deep Learning	R	It has excellent libraries and frameworks for building and training models
PyTorch	Deep Learning Library	Tensorflow	Develop CNN model
Google Colab	Coding ,Visualization	Kaggle Notebook	Interactive computing environment
Matplotlib	Visualizing data distributions, model performance		To visualize model performance

Hardware Components:

Table II. List of Hardware Tools

Tools	Functions	Other Similar Tools (If any)	Why selected this tool?
GPU	Accelerate the training of learning models	CPU	Reduce the complexity of learning models
SSD	Store dataset , trained models and other project related files	HDD	Storing files

3.3 Implementation

A. **Data Preprocessing**: We define a dataset class in PyTorch, which helps us to easily train a model for either color blindness correction or simulation. We loaded pairs of images from a specified directory, where each pair contains the original image and its modified daltonized version that simulates one of the possible types of color blindness. We perform transformations such as resizing, conversion to tensors, and normalization to prepare the images for the model. The dataset can now be easily used in training loops to feed the model with the original and colorblind image pairs for learning the transformations needed to correct or simulate color vision deficiencies.

```
def preprocess_image(image):
    if image.dtype != torch.float32:
        image = image.to(torch.float32)

    image = (image + 1) * 0.5

    return image

class ColorblindDataset(torch.utils.data.Dataset):

    def __init__(self, data_dir, colorblind_types, transform=None):
        self.data_dir = data_dir
        self.colorblind_types = colorblind_types
        self.transform = transforms.Compose([
            transforms.Resize((256, 256)),
            transforms.ToTensor(),
            transforms.Normalize([0.5, 0.5, 0.5], [0.5, 0.5, 0.5])
        ])
        self.image_paths = self.load_data()

    def load_data(self):
        image_paths = []
        for colorblind_type in self.colorblind_types:
            type_dir = os.path.join(self.data_dir, colorblind_type)
            original_dir = os.path.join(self.data_dir, "original")
            for filename in os.listdir(type_dir):
                if filename.endswith((".jpg", ".png", ".jpeg")):
                    original_path = os.path.join(original_dir, filename)
                    colorblind_path = os.path.join(type_dir, filename)
                    image_paths.append((original_path, colorblind_path))
        return image_paths

    def __len__(self):
        return len(self.image_paths)

    def __getitem__(self, index):
        original_path, colorblind_path = self.image_paths[index]

        # print(f"Image loaded: {original_path}")
        # print(f"Colorblind_image Type: {colorblind_path}")

        original_image = Image.open(original_path).convert("RGB")
        colorblind_image = Image.open(colorblind_path).convert("RGB")

        # print("Original Image Shape (after loading):", original_image.size)
        # print("Colorblind Image Shape (after loading):", colorblind_image.size)

        if self.transform:
            original_image = self.transform(original_image)
            colorblind_image = self.transform(colorblind_image)
            # print("Original Image Shape (after preprocessing):", original_image.
                shape)
            # print("Colorblind Image Shape (after preprocessing):", colorblind_image
                .shape)

        return original_image, colorblind_image
```

Figure 2: Data Preprocessing

B. **Data Loader**:

```
data_dir = "data/train"
    dataset = ColorblindDataset(data_dir, colorblind_types, transform)
    dataloader = torch.utils.data.DataLoader(
        dataset, batch_size=batch_size, shuffle=True)

val_data_dir = "data/val"
    val_dataset = ColorblindDataset(val_data_dir, colorblind_types, transform)
    val_dataloader = torch.utils.data.DataLoader(
        val_dataset, batch_size=1, shuffle=False)
```

Figure 3: Training and Validation Data Loader

C. **Loss Function**: As the loss function, MS-SSIM guides the training of neural networks toward the synthesis of images that are perceptually similar to target images. The MS-SSIM measures the dissimilarity—1-MSSSIM—of the generated image from the target image across multiple scales, hence encouraging the optimization of the network in the generation of local details and structure of the image as a whole, which results in outputs closer to human perception.

D. **Optimizer**: The weights of the model in training are optimized using the Adam optimizer. Essentially, it is used to tune the parameters of the model by considering the gradient in the loss function in an incremental manner to improve the color feature detection capability of the model.

```
ssim = torchmetrics.StructuralSimilarityIndexMeasure(data_range=1.0).to(device)
ms_ssim_value = MS_SSIM(data_range=1, size_average=True, channel=3)

def calculate_loss(model_output, colorblind_image):
    return 1-ms_ssim_value(model_output, colorblind_image)

optimizer = optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)
```

Figure 4: Loss Function and Optimizer

E. **Model Architecture**: In our Autoencoder architecture, we start out with an encoding journey where we progressively down-sample the input image through convolutional layers, namely, enc1 through enc5, followed by batch normalization and ReLU activation. The process finally distills the image into a compact latent representation. Finally, up-sampling the latent space is done via the transposed convolutions, dec5 through dec1, in a manner that maintains a structure similar to the traditional autoencoder, which is followed by the un-padding at each decoding step to refine the quality of the reconstruction. After concatenating corresponding encoder feature maps to the decoder input, a final sigmoid activation pushes the output into the range of [0, 1], which is the final reconstruction looking very similar to the original input.

```
class Autoencoder(nn.Module):
    def __init__(self):
        super(Autoencoder, self).__init__()

        # Encoder
        self.enc1 = nn.Sequential(
            nn.Conv2d(3, 32, kernel_size=3, stride=2,
                      padding=1),  # Output: 32x128x128
            nn.BatchNorm2d(32),
            nn.ReLU(True)
        )
        self.enc2 = nn.Sequential(
            nn.Conv2d(32, 64, kernel_size=3, stride=2,
                      padding=1),  # Output: 64x64x64
            nn.BatchNorm2d(64),
            nn.ReLU(True)
        )
        self.enc3 = nn.Sequential(
            nn.Conv2d(64, 128, kernel_size=3, stride=2,
                      padding=1),  # Output: 128x32x32
            nn.BatchNorm2d(128),
            nn.ReLU(True)
        )
        self.enc4 = nn.Sequential(
            nn.Conv2d(128, 256, kernel_size=3, stride=2,
                      padding=1),  # Output: 256x16x16
            nn.BatchNorm2d(256),
            nn.ReLU(True)
        )
        self.enc5 = nn.Sequential(
            # Output: 512x8x8 (Latent Representation)
            nn.Conv2d(256, 512, kernel_size=3, stride=2, padding=1),
            nn.BatchNorm2d(512),
            nn.ReLU(True)
        )

        # Decoder (with skip connections)
        self.dec5 = nn.Sequential(
            nn.ConvTranspose2d(512, 256, kernel_size=4,
                               stride=2, padding=1),  # Output: 256x16x16
            nn.BatchNorm2d(256),
            nn.ReLU(True)
        )
        self.dec4 = nn.Sequential(
            # Output: 128x32x32 (Concatenated: 512 -> 256)
            nn.ConvTranspose2d(256 * 2, 128, kernel_size=4,
                               stride=2, padding=1),
            nn.BatchNorm2d(128),
            nn.ReLU(True)
        )
        self.dec3 = nn.Sequential(
            # Output: 64x64x64 (Concatenated: 256 -> 128)
            nn.ConvTranspose2d(128 * 2, 64, kernel_size=4,
                               stride=2, padding=1),
            nn.BatchNorm2d(64),
            nn.ReLU(True)
        )

        self.dec2 = nn.Sequential(
            # Output: 32x128x128 (Concatenated: 128 -> 64)
            nn.ConvTranspose2d(64 * 2, 32, kernel_size=4, stride=2, padding=1),
            nn.BatchNorm2d(32),
            nn.ReLU(True)
        )
        self.dec1 = nn.Sequential(
            # Output: 3x256x256 (Concatenated: 64 -> 32)
            nn.ConvTranspose2d(32 * 2, 3, kernel_size=4, stride=2, padding=1),
            nn.Sigmoid()
        )

    def forward(self, x):
        enc1 = self.enc1(x)
        enc2 = self.enc2(enc1)
        enc3 = self.enc3(enc2)
        enc4 = self.enc4(enc3)
        enc5 = self.enc5(enc4)

        dec5 = self.dec5(enc5)
        dec4 = self.dec4(torch.cat([dec5, enc4], 1))  # Skip connection
        dec3 = self.dec3(torch.cat([dec4, enc3], 1))  # Skip connection
        dec2 = self.dec2(torch.cat([dec3, enc2], 1))  # Skip connection
        dec1 = self.dec1(torch.cat([dec2, enc1], 1))  # Skip connection
        return dec1
```

Figure 5: CNN-Based Autoencoder Model with Skip Connections

F. **Training Loop & Validation Loop**: In the training process, we run over the number of epochs specified. We load images from our dataset in batches in each epoch and feed them into the autoencoder. After a forward pass, it generates the images reconstructed; then, we calculate the loss with the corresponding colorblind images. The Adam optimizer is then used to tune the model parameters based on calculated loss. We calculate average loss and SSIM at each training epoch. We then test the model performance on the separate validation dataset while calculating the average loss and SSIM. Images in both the original and colorblind formats, along with the reconstructed images for each kind of colorblindness, are shown throughout both training and validation. We have, by the end of this process, an autoencoder model trained to restore color vision.

```
# Hyperparameters
num_epochs = 50
batch_size = 64
learning_rate = 0.0001

# Create the Autoencoder model
model = Autoencoder().to(device)
# print(model)
optimizer = optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)


colorblind_types = ["protanopia", "tritanopia", "deuteranopia"]

transform = transforms.Compose([
    transforms.Resize((256, 256)),
    transforms.ToTensor(),
    transforms.Normalize([0.5, 0.5, 0.5], [0.5, 0.5, 0.5])
])

# Training Loop
total_iterations = 0
train_losses_epoch = []  # Average loss per epoch (training)
train_ssim_epoch = []    # Average SSIM values per epoch (training)
val_losses_epoch = []    # Average loss per epoch (validation)
val_ssim_epoch = []      # Average SSIM values per epoch (validation)

for epoch in range(num_epochs):

    data_dir = "data/train"
    dataset = ColorblindDataset(data_dir, colorblind_types, transform)
    dataloader = torch.utils.data.DataLoader(
        dataset, batch_size=batch_size, shuffle=True)

    model.train()
    train_loss = 0
    train_ssim = []

    for i, (image, colorblind_image) in enumerate(dataloader):
        original_images = preprocess_image(image).to(device)
        colorblind_images = preprocess_image(colorblind_image).to(device)
        target_colorblind_type = colorblind_types[i % len(colorblind_types)]
        # Zero the parameter gradients
        optimizer.zero_grad()

        # print("Original Image Range:", original_images.min(), original_images.max()
            )
        # print("Target Image Range:", colorblind_images.min(), colorblind_images.max
            ())
        # Forward pass (reconstruction)
        output_images = model(original_images)

        # print("Output Image Shape:", output_images.shape)
        # print("Output Image Range:", output_images.min(), output_images.max())

        # Compute loss
        loss = calculate_loss(output_images, colorblind_images)
```

```
        # Backward pass and optimize
        loss.backward()
        optimizer.step()
        total_iterations += 1

        train_ssim_value = ssim(output_images, colorblind_images)

        # Print loss and SSIM for each iteration (not stored)
        print(f"Epoch: {epoch + 1}/{num_epochs}, Type: {target_colorblind_type}, Iteration: {
            i+1}, Training Loss: {loss.item():.4f}, SSIM: {train_ssim_value.item():.4f}")

        train_loss += loss.item()
        train_ssim.append(train_ssim_value.item())

    # Calculate average loss and SSIM for the epoch (training) and store for plotting
    avg_train_loss = train_loss / len(dataloader)
    avg_train_ssim = sum(train_ssim) / len(train_ssim)
    train_losses_epoch.append(avg_train_loss)
    train_ssim_epoch.append(avg_train_ssim)

    visualize_images(original_images, colorblind_images,
                     output_images, target_colorblind_type, total_iterations-1)

    val_data_dir = "data/val"
    val_dataset = ColorblindDataset(val_data_dir, colorblind_types, transform)
    val_dataloader = torch.utils.data.DataLoader(
        val_dataset, batch_size=1, shuffle=False)

    model.eval()  # Set model to evaluation mode
    val_loss = 0
    val_ssim = []
    with torch.no_grad():  # No need to calculate gradients for validation
        for image, colorblind_image in val_dataloader:
            original_images = preprocess_image(image).to(device)
            colorblind_images = preprocess_image(colorblind_image).to(device)
            output_images = model(original_images)
            loss = calculate_loss(output_images, colorblind_images)

            val_ssim_value = ssim(output_images, colorblind_images)

            val_loss += loss.item()
            val_ssim.append(val_ssim_value.item())

        target_colorblind_type = colorblind_types[i % len(colorblind_types)]
        visualize_images(original_images, colorblind_images,
                         output_images, target_colorblind_type, 0)

    avg_val_loss = val_loss / len(val_dataloader)
    avg_val_ssim = sum(val_ssim) / len(val_ssim)
    val_ssim_epoch.append(avg_val_ssim)
    val_losses_epoch.append(avg_val_loss)
    print(f"\nEpoch: {epoch + 1}/{num_epochs}, Training Loss: {avg_train_loss:.4f}, Training SSIM: {
        avg_train_ssim:.4f},\nValidation Loss: {avg_val_loss:.4f},Validation SSIM: {avg_val_ssim:.4f}\n")
```

Figure 6: Training Loop & Validation Loop

G. **Model Saving**: If the training gets done and the model behaves satisfactorily, then it is saved to a file. Such a saved model can be later used for other deployment opportunities.

In general, it is an implementation pipeline designed to consider data preparation, model architecture, training, evaluation, and model saving. It looks to utilize the power of deep learning in generating enhanced daltonized images that will benefit color-deficient people.

Chapter 4 Investigation/Experiment, Result, Analysis and Discussion

A. **Experiment**: The experiment, therefore, aims at developing an autoencoder model for color vision deficiency correction. We did this by training a dataset containing images that represented various scenes as seen through the eyes of people with various types of color blindness: protanopia, deuteranopia, and tritanopia, on a deep convolutional neural network. Our preprocessing consisted of resizing images and applying normalization to enhance the model's generalization ability. The architecture of the autoencoder includes skip connections that improve the reconstruction quality, learning a compact representation of the input images to generate the color-corrected versions. The model is evaluated using the structural similarity index measure—a perceptual quality assessment.

B. **Results** (Visualized):

i) **Protanopia** (Missing or malfunctioning L-cones (Red)):

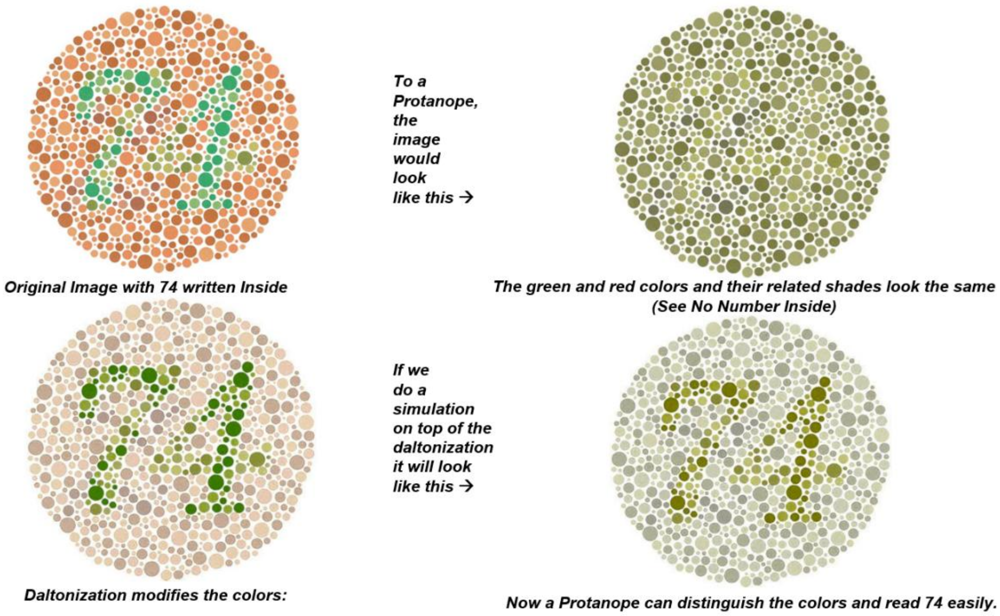

Figure 7: Protanopia Result (Visualized)

ii) **Deuteranopia** (Missing or malfunctioning M-cones (Green)):

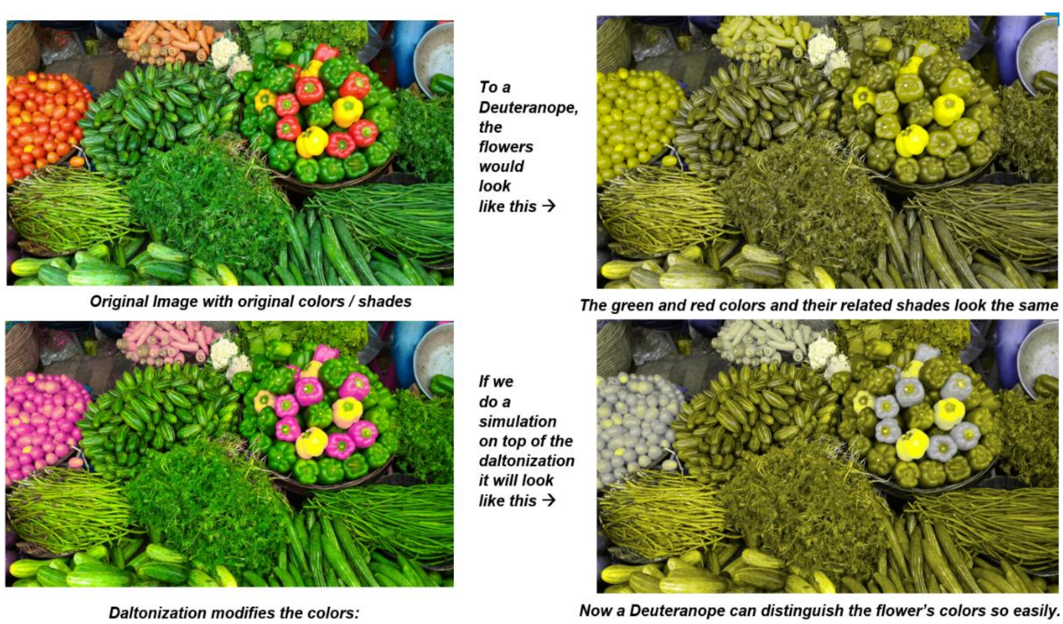

Figure 8: Deuteranopia Result (Visualized)

If we look at results with various shades of color, it is then easy to realize how hard it is to tell the difference in colors for a person with color deficiency. The shades often mix together. The human eye can detect approximately 10 million shades of color. So, our generated results suggest that an enhanced range of shades according to the perception of the color-deficient observer will allow them to pick out colors more easily.

- If we could see the picture, there are many shades of colors that the human eye can perceive in the flower. Some of its main shades are shown in the suggested color palette. There are also a lot of other shades of these colors.

Figure 9: Original Image

- Suppose we simulate the image to show how a person with deuteranopia, that is, red/green color blindness, can see the colors and shades of the flowers. Then we will see that, for them, the colors and shades are quite similar, overlapping, and therefore indistinguishable.

Figure 10: Deuteranopia Simulation

- The Daltonization algorithm works wonders in balancing color while holding the same content meaning. It recolors most of the indistinguishable colors into a form that allows color-deficient people to see most colors within their color palettes.

Figure 11: Deuteranopia Daltonization

- Finally, running the simulation of the Daltonized version, we observe that more colors are seen within the palettes. The missing cones in their retinas do not permit them to visualize the colors, and they need to resort to medical treatment. However, they can observe more colors and they will discern colors, objects, and letters better than usual.

Figure 12: Deuteranopia Daltonization & Simulation

C. **Results** (Curves):

a. **Loss Curve**:

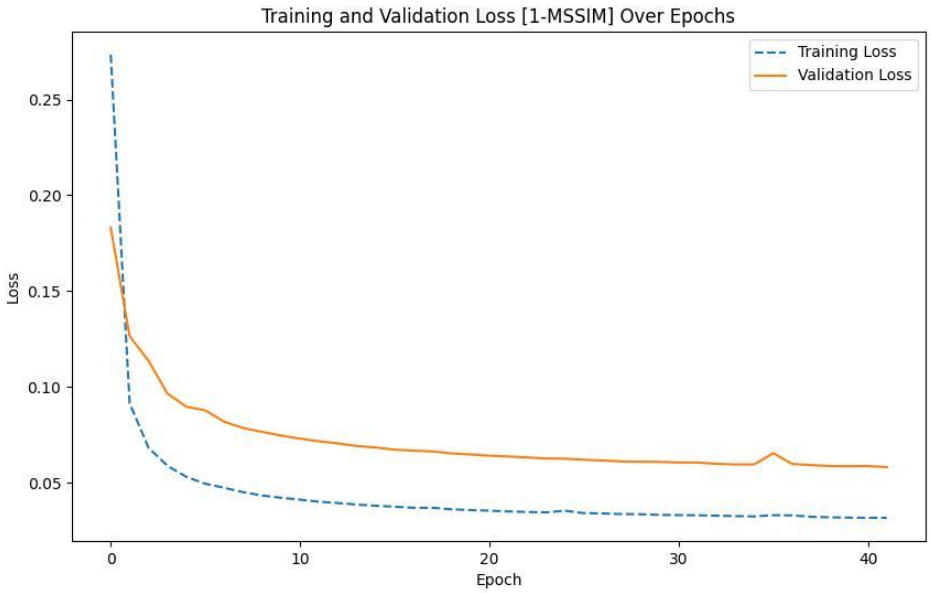

Figure 13: Loss Curve (1-MSSSIM)

Training Loss: **3.16%**

Validation Loss: **5.81%**

Analysis: The graph shows the model's learning over the 40 epochs measured by loss (1-MSSSIM). The loss value decreases as the model is learning better for the reconstruction of the images. During training, the loss drops from 0.26 to 0.04 within the first 10 epochs, showing rapid learning on the training data. Validation loss decreases from 0.19 and gradually reaches 0.06 around epoch 30, which shows the slower and consistent improvement on unseen data. A continuous decline in both losses depicts that the model is effective in learning the reconstruction of images with increasing similarity to the original ones.

b. **SSIM Curve**:

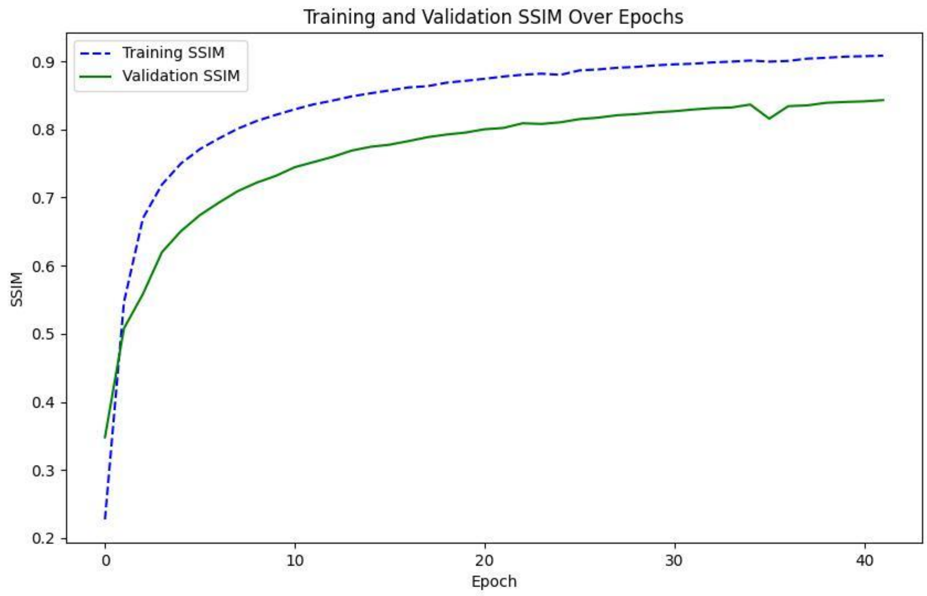

Figure 14: SSIM Accuracy Curve

Training SSIM: **90.83%**

Validation SSIM: **84.30%**

Analysis: The graph depicts training and validation Structural Similarity Index Measure over 40 epochs. Training SSIM starts off at about 0.25 and increases to 0.9 within the first 10 epochs, showing that the model is really quick to learn how to reconstruct images that are similar to the original training data. The validation SSIM also starts at about 0.35 and rises steadily, though it flattens out at about 0.85 after 30 epochs; that suggests the model has reached its limit in being able to generalize to unseen data, in this case, the validation set. Overall, the trend of the SSIM scores shows the effectiveness of the model in terms of image reconstruction, whereby the training SSIM always outperforms that of the validation.

D. **Discussion**: Our research has focused on the potentials of autoencoders—types of deep learning models—on enhancing the quality of images that are subjected to color blindness. The integration of skip connections enabled our model to capture and reconstruct the minute details in the images; this is evidenced by the achievement of high SSIM scores. The plots from the training process of the model showed a rapid drop in 1-MSSIM loss, depicting that our model was able to quickly grasp the inherent patterns and structures from the input data.

Though our results are promising, still there is scope for improvement. The slight difference between the training and validation loss curves indicates that minor overfitting could occur, and this could be rectified by the use of proper regularization techniques or by data augmentation. Experimentation with other loss functions or architecture changes may further improve the performance of the model.

Chapter 5 Impacts of the Project

5.1 Impact of this project on societal, health, safety, legal and cultural issues

Societal Impact:

Changing the curriculum: The new design of educational materials, textbooks, and digital resources will make them inclusive, ensuring equal access to information for colorblind students.

Training of teachers: Teachers will undergo intense training on color blindness to help them build open learning environments and change teaching methods where necessary.

Helping devices: Schools will start using supportive devices such as software that enables people with color vision deficiency to use them or optical tools to enhance colors when studying materials developed in different colors.

Change in thought: Color blindness will be seen by society as the natural difference in human eye vision rather than a disability, therefore cherishing differences and promoting togetherness.

Innovation Movements: Therefore, by inspiring future innovations, the project might serve as a template for addressing other visual or sense-making disabilities, considering its achievements.

Healthcare:

Recent Breakthroughs in Research: The proper execution could speed up the research into the genetic or neurologic causes of colorblindness, bringing forth possible remedies and preventative measures.

Psychological Support for Mental Health: Assisting people to live with this condition psychologically could provide different counseling options, such as details of how one can get in touch with a counselor in an emergency.

Safety:

Universal Design: In public spaces, transportation systems, and workplaces, colorblind-friendly design are what would be, and this will significantly reduce the potential accidents and injuries in them.

Emergency Preparedness: Ensure the evacuations and communications flow effectively among CVD individuals, as well as emergency response procedures and signage.

Legal Implications:

Legal and Policy Landmark Legislation: Landmark legislation mandating different industries should ensure that there are colorblind-friendly design and accessibility standards to help drive success.

Litigation and Advocacy: The right of every person with color vision deficiency should be advocated for legally by empowering legal practitioners to fight on behalf of these people, thus making them equally protected and have equal opportunities in life.

Culture and Arts:

Representative and Awareness: This project has the potential to start a significant change in media, arts, and entertainment that would lead to more excellent representation and acceptance when it comes to colorblindness.

Creative Streams: The colorblind artists would open new borders by using innovative techniques challenging old concepts about both color and beauty.

5.2 Impact of this project on environment and sustainability

Social Equity: The programs we design towards simulating, quantifying, and compensating for color blindness while promoting inclusiveness in technology. It will be possible for those who lack color vision to participate fully in digital worlds; it also leads to less discrimination and friendlier social relations where equal chances exist.

Reducing Electronic Waste: This involves accommodating color blindness through software solutions that potentially help us do away with the requirements of specialized hardware devices or physical modifications into current technologies, thus leading to less electronic waste from outdated or replaced hardware and contributing more to environmental conservation efforts.

Enhancement of Energy Efficiency: In our projects, adaptive software technologies are designed to consume less power than their hardware counterparts. Promoting energy efficiency encourages substituting energy-saving software solutions for resource-intensive hardware by providing a digital color blindness platform.

Global Reach and Impact: Color blindness affects individuals worldwide, and our projects have the potential for widespread impact by enabling individuals with color vision deficiencies to better interact with digital content and technology. Moreover, ensuring everyone has equal access to opportunities and information regardless of visual ability will support global sustainability initiatives.

Advancement of Awareness and Education: Raising awareness about the challenges people face with color vision deficiency. However, by highlighting the importance of accessibility and providing practical solutions, our projects educate developers and users about the needs of this demographic, fostering a culture of inclusivity and sustainability in technology development.

Embracing sustainable behaviors may entail optimizing energy usage when analyzing statistics as well as using virtual collaboration programs in order not only to do away with all things concerning motion but also to save time and money. Additionally, e can help save the planet by giving people knowledge about color blindness problems and supporting CVD individuals with eco-friendly solutions from a broader perspective of provision of care and patronage.

Chapter 6 Project Planning and Budget

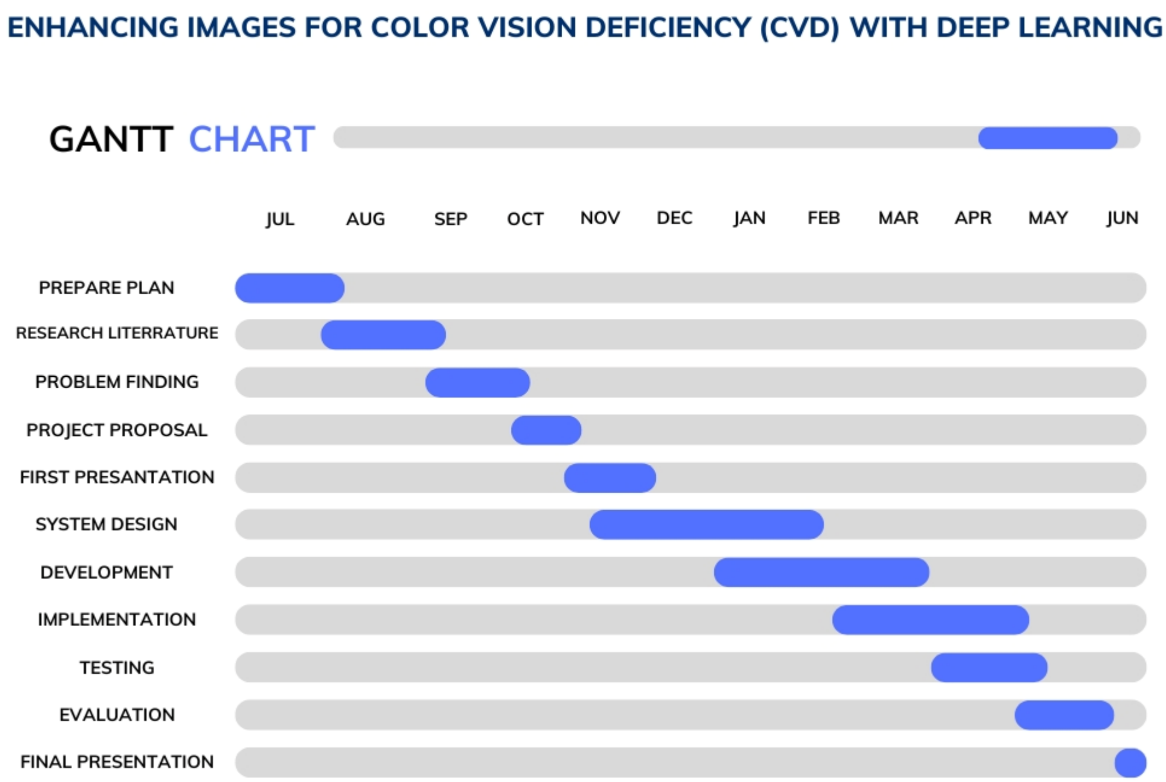

Figure 15. Gantt chart.

Component	Unit Price (in BDT)	Months	Total Cost
Colab Subscription	1350	3	4050

Figure 16. Budget table.

Chapter 7 Complex Engineering Problems and Activities

7.1 Complex Engineering Problems (CEP)

Table III. Complex Engineering Problem Attributes

Attributes		Addressing the complex engineering problems (P) in the project
P1	Depth of knowledge required (K3-K8)	This research project necessitates a comprehensive understanding of the problem profile of CVD (color-vision- deficiency) and the Biological explanation behind it (K3), Daltonization and Deep Learning (K4), CVD Detecting Software Tools, and Data Analysis (K5), Programming (K6), Knowledge of Dataset of Image for Three Major CVD (K7), and familiarity with Scientific Research Papers (K8).
P2	Range of conflicting requirements	In the context of this research, enhancing images for CVD and the computational complexity (computational resources) are interconnected. Strengthening the model needs a wide range of CVD-affected people.
P3	Depth of analysis required	The intricate nature of enhancing images for CVD calls for an in-depth analysis of various methodologies. This research thoroughly uses algorithms like Daltonization, deep learning models like CNN-based Auto-Encoder, and feature selection techniques for three types of CVD to choose the most suitable approach for generating images.

P4	Familiarity of issues	Familiarity issues with enhancing images for CVD always take us to the realization of the vision that a CVD faces all the time and how we can solve the issue by not being affected by the problem.
P5	Extent of applicable codes	Significant modifications have been made to the existing code to get the accepted output from the model when applying the CNN-based Auto Encoder to the Daltonized image.
P6	Extent of stakeholder involvement	Stakeholders in this research include the research team, funding agencies, and potentially academic institutions or organizations interested in the outcomes of generated images for CVD and applying the model to the different glasses and as a plugin in different software or related fields.
P7	Interdependence	The research project relies on a synergy of various interconnected components, including dataset collection, algorithm selection, model training, and result analysis. These components need to work cohesively to achieve meaningful images for CVD.

7.2 Complex Engineering Activities (CEA)

Table IV. Complex Engineering Problem Activities

Attributes		Addressing the complex engineering activities (A) in the project
A1	Range of resources	The project requires various resources, including human resources, funding, modern tools (such as simulation software and mobile apps), hardware components, and access to large datasets of images.
A2	Level of interactions	The project will involve interactions between various stakeholders, including researchers, people with CVD, drug discovery, smart glass manufacturers, software developers, and funding agencies.
A3	Innovation	The project will employ innovative engineering skills by introducing deep learning in the CVD field, which was rare in the past.
A4	Consequences to society / Environment	Develop systems to simulate, quantify, and compensate for CVD and promote inclusivity in technology. Participating in digital environments fosters social sustainability by reducing discrimination and promoting equal opportunities.
A5	Familiarity	The project requires familiarity with deep learning, such as CNN-based Auto-Encoder and Daltonization Algorithm. Additionally, the project requires familiarity with huge datasets with colorful images.

Chapter 8 Conclusion

8.1 Summary

The common genetic disorder of color vision prevents millions of people worldwide from detecting and expressing colors that change their world every day. The research looks at how artificial intelligence (AI) enhances visual accessibility among the colorblind in society. The two main AI models are Daltonization, which mimics different types of CV disorders by producing specific photo transformations, and a CNN-based Autoencoder, developed on photographs taken from healthy individuals and those suffering from color vision deficiency.

The project will gather images from various backgrounds and evaluate model performance based on quantitative evaluations and qualitative reports from individuals suffering from color deficiencies. Ethical concerns will be vital to our focus, while changed pictures will comply with accessibility rules.

We are interested in developing user-friendly tools that run on different platforms, and they should improve visual experiences during education, work, and other daily activities. The cardinal purpose of this study is not just to enhance the life standards of colorblind individuals but also to propel further use of AI in accessibility issues. We would re-write it this way. To develop platforms that simplify the process, it should aim at enriching educational, working, and life activities as far as vision is concerned, but it should also advance AI applications in accessibility for a better life of individuals affected by disability or aging.

Even though some research has shown improvement potential in enhancing color contrast and constancy, current stumbling blocks are unintended color shifts, real-time video processing capabilities, and computational complexity where enhancements are made. More so, developing and validating comprehensive tools for compensating CVD face several setbacks; they include lack of extensive real-world data and subjective evaluations." Notwithstanding these constraints, the study indicates that deep learning and fuzzy-logic-based processes match the personalities mentioned earlier and increase their accessibility.

A CNN-based Autoencoder with skip connections is meant to convert images for insiders with color vision deficiency (CVD). Therefore, around 17,276 images from a Kaggle landscape image database were altered using the Daltonization algorithm. The sequences of actions the photos underwent are normalization followed by resizing them. The model was trained in 80% of its data samples and tested in the remaining 20%. These instances of this network's functioning raised the fact that its performance is so precise (1-MSSIM) that the error function in an Adam optimizer was used for training. Fine-tuning and adjusting hyperparameters have been performed to optimize the model's performance using SSIM as an evaluation metric.

For a long time, the blindness project appeared to hold unlimited possibilities to change people's lives in many different areas, such as encouraging diversity and increasing creativity. In several idiosyncratic principles, the color-blinded suggestive design enhances union and unlocks a dormant workforce. Increasing the speed of investigation on causes of CVD will be a real impetus for improvement in medicine. Also, increasing mental health support among those with this problem will help evolve health care by all means.

Improved safety procedures in work environments and public places will reduce risks, embracing universal design principles and emergency measures considering color blindness. By law, implementing the project might result in groundbreaking legislation that would compel organizations to adopt colorblind-accessible architectural guidelines and increase efforts toward equal rights and opportunities. This project may contribute to a more acceptable attitude towards color blindness in society, inspire representation, and foster creative communication.

The use of software to help with colorblindness has many advantages. This article will explore some of these benefits: promoting social equity, depleting electronic waste, and increasing energy efficiency. Consequently, this would expand because the project eventually goes global, creating awareness among developers while enabling them to get informed about tech inclusiveness.

In our project, the Training Loss is **3.16%,** and the Validation Loss is **5.81%**

On the other hand, the Training SSIM **is 90.83%,** and Validation SSIM is **84.30**

Overall, this all-encompassing project tackles a global problem, bettering the lives of people with CVD and ensuring a future where everybody can exist sustainably and equitably.

8.2 Limitation

The specific limitation has to be maintained:

Limited Dataset: The efficiency of deep learning should be independent of the quality and variety of the training data. Obtaining a large and representative dataset of images viewed under various CVD conditions is often a difficult task, constraining the model's ability to generalize across diverse real-world scenarios.

Real-Time Performance: Utilizing AI models in real-time applications like intelligent glasses or enhanced reality often needs higher processing speed. The result requires optimization to make them efficient on devices with minimal memory.

Cost and Accessibility: AI tools might be inaccessible to people who cannot afford AI-powered tools as their development and deployment can be expensive. Action to ensure their availability and affordability to all. Also, a lower GPU computation makes the model less efficient.

The project will have a better chance at success and be more widely adopted if these address constraints. Continued work on user input and iterative development that refines their effectiveness in improving visual accessibility for people with color blindness is affected by dealing with these limitations.

8.3 Future Improvement

The neural network model we built gives hope for integration into future smart and VR glasses targeting people with color vision disorders. For example, Ray-Ban and Google have already developed various features using their smart glasses, so the implementation will not be a hitch. Further expanding their capabilities, customers can install our model as a software module or app on these devices. It has caused increasing interest in products that could help people deal with the consequences of Cardiovascular Disease (CVD), which affects 300 million individuals in this world and has never been cured.

Incorporating our model in brilliant spectacles will enable individuals with CVD to enjoy a more prosperous and colorful visual world at any given time. Our model can help such persons live better lives by providing direction on how to go about it in their day-to-day activities or viewing things around them in normality, sometimes using augmented reality functionalities.

Aside from smart glasses, we can also incorporate our model into leading photo editing applications like Adobe Photoshop. It would allow CVD users to easily modify their pictures more accurately than ever before, serving their color perception demands while providing an essential tool. Such an inbuilt capability would allow CVD individuals or groups who use these applications daily to have access to them on demand, thereby increasing the customer base for Photoshop and showing a firm resolve towards inclusive designing and universal access.

On the whole, the opportunities of our model are broad and varied and have numerous meaningful consequences for people with CVD. By capitalizing on technologies already in place while also collaborating with leaders from other industries, this is a solution we can popularize that goes beyond the ordinary; it would improve visual accessibility and thus enhance the quality for millions.

References

1. Anagnostopoulos, C., Tsekouras, G. E., Anagnostopoulos, I., & Kalloniatis, C. (2007). Intelligent modification for the daltonization process of digitized paintings. *ResearchGate*. https://www.researchgate.net/publication/37683505_Intelligent_modification_for_the_daltonization_process_of_digitized_paintings
2. *Colour Blindness on the move: A brief Literary and Cultural History - Science Museum blog*. (2019, September 6). Science Museum Blog. https://blog.sciencemuseum.org.uk/colour-blindness-on-the-move-a-brief-literary-and-cultura history/?fbclid=IwZXh0bgNhZW0CMTAAAR3FWA3qdHCvPJFfNRSXOLl2JKPJjExRz

HzQu9GfmUJxYHrzMQcl0ccbWc4_aem_AT5_PC41eI98UX2V0s3NUAAESu36BUtdL0bzudukki_NMbc4kJVHdj06M14nPIVwaQXB5SPhMavF4DBY5mpOKlGp

3. Hunt, D. M., Dulai, K. S., Bowmaker, J. K., & Mollon, J. D. (1995). The chemistry of John Dalton's color blindness. *Science*, *267*(5200), 984–988. https://doi.org/10.1126/science.7863342

4. Lee, J., & Santos, W. P. D. (2011). An adaptive fuzzy-based system to simulate, quantify and compensate color blindness. *Integrated Computer-aided Engineering*, *18*(1), 29–40. https://doi.org/10.3233/ica-2011-0356

5. Li, H., Zhang, L., Zhang, X., Zhang, M., Zhu, G., Shen, P., Li, P., Bennamoun, M., & Shah, S. a. A. (2020). Color vision deficiency datasets & recoloring evaluation using GANs. *Multimedia Tools and Applications*, *79*(37–38), 27583–27614. https://doi.org/10.1007/s11042-020-09299-2

6. National Eye Institute. (2024, April 15). *Color blindness | National Eye Institute*. https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/color-blindness?fbclid=IwZXh0bgNhZW0CMTAAAR1Iem2suaw1GPodEc9gX5fTq6RbN6EjEHrupbLFfQBRpb9_NcBaDOUwguI_aem_AT7h92k-TgWGDq_8PHJC7OPsMRy6MGt-bh7qbeH3YDWDYZuJIEO0iV-bLAcuq5CJej29U6xmeooq4vSxO3czthOi

7. Shen, X., Feng, J., & Zhang, X. (2020). A content-dependent Daltonization algorithm for colour vision deficiencies based on lightness and chroma information. *IET Image Processing*, *15*(4), 983–996. https://doi.org/10.1049/ipr2.12079