**A Major Project Report (ETMJ600)**

on

**DEEP LEARNING-BASED IMAGE COLORIZATION**

Submitted to



**Amity University Kolkata**

for partial fulfillment of the requirements for the award of the degree of

**MASTER OF COMPUTER APPLICATION**

by

**SUDARSANA ACHARJEE**

**A914145023009**

Under the guidance of

**Dr. Anwesha Law**

**AMITY INSTITUTE OF INFORMATION TECHNOLOGY, KOLKATA AMITY UNIVERSITY, KOLKATA**

**April, 2025**

**DECLARATION**

I hereby declare that the dissertation entitled “**DEEP LEARNING-BASED IMAGE COLORIZATION**” submitted by me in partial fulfillment of the requirements for the award of the Degree of “**MASTER OF COMPUTER APPLICATION”** to the **“AMITY UNIVERSITY KOLKATA**” is based on the experiments and studies carried out by me. This work is original and has not been submitted in part or full for any other degree or diploma of any university or institution.

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Place: Kolkata Sudarsana Acharjee

A914145023009

**Amity Institute of Information & Technology**

**Amity University Kolkata**

**CERTIFICATE**

The Project work embodied in this dissertation entitled “**DEEP LEARNING-BASED IMAGE COLORIZATION**” submitted by **SUDARSANA ACHARJEE**, bearing enrollment number **A914145023009** in partial fulfillment of the requirements for the award of the Degree of **“MASTER OF COMPUTER APPLICATION”** to the **“AMITY UNIVERSITY, KOLKATA”** is based on the experiments and studies carried out by her. This work is original and has not been submitted in part or full for any other degree or diploma of any university or institution.

Date:

Place: Kolkata

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Anwesha Law

Assistant Professor

Amity Institute of Information Technology

Amity University Kolkata

**ACKNOWLEDGMENT**

I would like to express my heartfelt gratitude to all those who supported and guided me throughout the completion of this project titled “**DEEP LEARNING-BASED IMAGE COLORIZATION***."*

First and foremost, I am deeply thankful to “**ANWESHA LAW**”, whose expertise, valuable feedback, and encouragement were instrumental in shaping the direction of this work. Their continuous guidance helped me overcome various challenges during the research and development process.

I also wish to thank **“AMITY UNIVERSITY, KOLKATA”** for providing the necessary resources and a conducive environment for learning and experimentation. The access to computing facilities and research materials played a vital role in carrying out the practical aspects of this project.

My sincere appreciation extends to my peers and friends for their helpful discussions, suggestions, and moral support throughout the journey. I am also grateful to my family for their constant motivation and understanding, which kept me focused and determined.

Lastly, I acknowledge the contributions of the wider research community, whose work in the field of deep learning and image processing has been a valuable source of inspiration and knowledge.

This project has been a rewarding experience, allowing me to explore innovative technologies and apply them to real-world problems, and I am truly thankful to everyone who played a part in making it possible.

Sincerely,

Sudarsana Acharjee

**AMITY UNIVERSITY, KOLKATA**



**FEEDBACK BY EXAMINERS**

1. **Comment From Seminar Guide**
2. **Comment From External Examiner**

**ABSTRACT**

Image colorization is the process of adding plausible colors to grayscale images, transforming them into visually appealing and more informative versions. This project explores a deep learning-based approach to automate the colorization task using convolutional neural networks (CNNs). Traditional methods often rely on manual input or reference images, which can be time-consuming and less adaptable to diverse datasets. In contrast, deep learning models can learn complex patterns and color distributions directly from large volumes of training data.

The proposed system is trained on a dataset of RGB images, where the network learns to predict the color channels from the grayscale input. The architecture is designed to capture both low-level textures and high-level semantic features, enabling more accurate and context-aware colorization. The results are evaluated using both qualitative visual assessments and quantitative metrics such as PSNR and SSIM to ensure fidelity and realism.

This work demonstrates the effectiveness of deep learning in addressing the limitations of traditional colorization techniques. By automating the process, it has potential applications in restoring historical photographs, enhancing medical imaging, and improving visual media content. The model showcases how neural networks can be leveraged to bring grayscale images to life with minimal human intervention.

**CONTENT**

* Declaration……………………………………………………………… i
* Certification…………………………………………………………….. ii
* Acknowledgement……………………………………………………… iii
* Abstract…………………………………………………………….…… iv
* Contents………………………………………………………………… v
* List of Figures…………………………………………………………... vii
* List of Abbreviations…………………………………………………… viii

**CHAPTER 1** INTRODUCTION………………………………………………............. 1

**CHAPTER 2** LITERATURE SURVEY………………………………………………... 2

**CHAPTER 3** METHODOLOGY………………………………………………………. 3

**3.1**. Data Collection and Preprocessing…………………………………….. 5

**3.2.** Model Architecture…………………………………………………….. 7

**3.3.** Training the Model…………………………………………………….. 8

**3.4.** Inference and Post – Processing……………………………………… 10

**CHAPTER 4** APPLICATIONS OF RECOMMENDATION SYSTEM……………….. 12

**4.1.** User-Guided Color Suggestions………………………………………. 12

**4.2.** Reference Image Selection……………………………………………..13

**4.3.** Artistic Style Recommendations……………………………………….14

**4.4.** Personalized Colorization Models……………………………………..14

* 1. **.** Enhanced Workflow in Creative Industries……………………………15

**4.6.** Educational and Historical Restoration Tools…………………………16

**CHAPTER 5** CHALLENGES AND LIMITATIONS…………………………. 17

**5.1.** Inherent Color Ambiguity……………………………………………. 17

**5.2.** Dependence on Data Quality and Diversity………………………….. 18

**5.3.** Loss of Detail and Texture…………………………………………… 19

**5.4.** High Computational Requirements…………………………………… 20

**5.5.** Limited User and Customization……………………………………… 21

**5.6.** Generalization to Unseen Content…………………………………….. 21

**5.7.** Difficulty in Evaluation……………………………………………….. 22

**5.8.** Ethical and Cultural Sensitivity Concerns…………………………….. 23

Conclusion……………………………………………………………….

**CHAPTER 6** RECOMMENDATION SYSTEMS' FUTURE SCOPE………..

**6.1.** Personalized Colorization Experiences………………………………… 24

**6.2.** Intelligent Color Palette Suggestions…………………………………… 25

**6.3.** Context-Aware Reference Image Selection…………………………….. 26

**6.4.** Collaborative Filtering for Community-Based Colorization…………… 27

**6.5.** Adaptive Learning and Real-Time User Feedback……………………… 28

**6.6.** Multi-Modal Recommendation Systems………………………………… 29

**6.7.** Support for Diverse Cultural and Historical Contexts…………………… 30

**6.8.** Incorporation into Creative and Educational Environments……………..30

Conclusion…………………………………………………………………

**CHAPTER 7** CONCLUSION……………………………………………………

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Name** | **Page No.** |
| **Figure 1.1** | **colorization layers** | **7** |
| **Figure 7.1** | **Image Comparison Building** | **32** |
| **Figure 7.2** | **Image Comparison car** | **32** |
| **Figure 7.3** | **Image Comparison Moon** | **32** |
| **Figure 7.4** | **Image Comparison Street at night** | **33** |
| **Figure 7.5** | **Image Comparison scene** | **33** |
| **Figure 7.6** | **Image Comparison Bird** | **34** |
| **Figure10.1:** | **Start Window** | **48** |
| **Figure10.2:** | **Result Window** | **48** |
| **Figure10.3:** | **List of unitary comparison** | **48** |

**LIST OF ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| **ABBREVIATIONS** | | **EXPANSION** |
| AI | Artificial Intelligence | |
| ANN | Artificial Neural Network | |
| BGR | Blue-Green-Red (Color Space) | |
| CNN | Convolutional Neural Network | |
| DNN | Deep Neural Network (OpenCV Module) | |
| GPU | Graphics Processing Unit | |
| LAB | Lightness (L), A (green–red), B (blue–yellow) Color Space | |
| MSE | Mean Squared Error | |
| OpenCV | Open Source Computer Vision Library | |
| Re LU | Rectified Linear Unit | |
| RGB | Red-Green-Blue (Color Space) | |
| SGD | Stochastic Gradient Descent | |
| VGG | Visual Geometry Group (Deep Learning Model) | |
| Res Net | Residual Network | |
| ILSVRC | ImageNet Large Scale Visual Recognition Challenge | |
| MAE | Mean Absolute Error | |
| Lab\* | A Color Space separating Lightness (L) and Color Opponents (a and b) | |
| Io U | Intersection over Union | |

# CHAPTER 1

## **INTRODUCTION**

Coloring black-and-white photographs to vibrant, realistic pictures has long intrigued artists, researchers, and historians. Traditionally, this was done by hand, taking a great amount of time, effort, and artistic talent. Artists would carefully add colors based on contextual hints, historical documents, and artistic license. While the outcome was at times stunning, hand colorization was limited by the availability of skilled labor and the subjective cues of the colorist. However, with the advent of artificial intelligence (AI), particularly deep learning technologies, the landscape of image colorization has undergone a profound transformation. Deep learning has enabled the development of highly effective, automatic methods that can restore and enrich monochromatic photographs with remarkable realism and efficiency.

Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) have demonstrated exceptional ability in learning complex visual patterns. CNNs, with their hierarchical mechanism of feature extraction, can classify and represent textures, shapes, and spatial hierarchies of images. GANs complement this even more by introducing the concept of an adversarial structure in which a generator network tries to learn producing very realistic colorizations and a discriminator network is concurrently trying to tell apart true colored images and those that were generated. By this competitive process, GANs can generate outputs that not only are statistically coherent with natural images but also perceptually engaging. These deep learning models are generally trained on large collections of color images, allowing them to learn a semantic representation of objects, materials, and scenes. After training, they are able to forecast reasonable color values for every pixel in a black-and-white image, frequently generating results that are both aesthetically pleasing and semantically correct.

In contrast to conventional rule-based or algorithmic colorization methods—which tend to involve explicit human input, handcrafted features, or pre-specified color mappings—deep learning methods are mostly independent. They do not need manual intervention at inference time and can generalize across various image types and styles. One of the major strengths of colorization using deep learning is its ability to understand images at a cognitive level. As an example, a trained model can identify something like sky, leaves, cloth, or skin and assign contextual colors accordingly. This semantic-level understanding allows creating natural images even in the total absence of any color information from the past.

The uses of deep learning-based colorization are extensive and significant. In historical research and preservation, colorization through automation brings new life to archival photos and old films, rendering historical material more accessible and engaging for contemporary audiences. In the entertainment sector, it offers strong tools for reimagining classic films, documentaries, and media artifacts in full color. In addition, within machine learning itself, colorization operations can act as a method of self-supervised learning that strengthens feature representations for downstream tasks like image classification, object detection, and segmentation.

However, for all its powerful abilities, deep learning-based colorization has challenges. One of the built-in challenges is the inherent ambiguity of the colorization problem—many objects could plausibly be a variety of colors. Without adequate contextual information, models can generate erroneous or inconsistent colorizations. In addition, the generalization capability and performance of a colorization model largely rely on the quality and diversity of training data. Pictures with rare objects, specialized fields, or uncommon lighting conditions are usually major challenges for current systems.

To overcome these limitations, recent studies investigate different approaches, such as the addition of user guidance, reference image usage, and the construction of more advanced model architectures like transformer-based networks. These efforts seek to enhance the accuracy, control, and flexibility of the colorization process, rendering it more robust and versatile for a broader range of applications.

In summary, deep learning has transformed the art and science of colorizing black-and-white photographs. By marrying automation with semantic knowledge and creative ability, it has made a historically time-consuming practice an intelligent, data-driven process. As research keeps advancing, deep learning-based colorization is on the verge of unlocking even more thrilling possibilities in the fields of art, history, culture, and technology.

# CHAPTER 2

# **LITERATURE SURVEY**

The method of colorizing monochrome images has gone through extensive enhancement in the past few years, particularly after deep learning methods came into action. Traditional practices involved the usage of manual input or heuristic techniques, which commonly proved to be time and knowledge-dependent. But deep neural networks brought about tremendous transformation in the practice by offering automation along with natural-looking colors.

One of the first influential papers in deep learning-based colorization is by Zhang et al. (2016), who proposed an automatic image colorization algorithm with a deep convolutional neural network. Their method posed colorization as a classification task in the ab color space of the CIELAB color model. Through the application of class-rebalancing techniques, the model could predict realistic colors and better deal with rare color distributions. This work laid the foundation for subsequent improvements in both accuracy and visual quality.

Expanding on this, Iizuka et al. (2016) presented a fully automatic colorization network that incorporated both global and local features. Their architecture employed two branches: global context learning and local texture, which were later combined for the ultimate color prediction. This two-branch approach allowed the model to not only know what was present in the image but also where it was, greatly improving the realism of the color produced.

Another significant innovation was achieved through the application of Generative Adversarial Networks (GANs). Isola et al. (2017) proposed the idea of image-to-image translation with conditional GANs (cGANs), in which grayscale images were translated into colored images via adversarial learning. The method enabled the generator to generate more realistic colors by learning how to deceive a discriminator network that was trained to identify real and colorized images.

More enhancements were observed with the addition of user guidance. Zhang et al. (2017) created a model that enabled user-inputted color hints or points, which the network propagated throughout the image. This interactive colorization method provided more control over the output, merging the benefits of automatic techniques with manual imagination.

Recent work has therefore aimed at making colorization models more robust and generalizable. For example, Su et al. (2020) investigated attention mechanisms in order to further concentrate on the most important areas of an image so that color predictions are more contextually relevant. Simultaneously, transformer models have also begun to attract attention with their success on other vision tasks, though they remain relatively nascent in the task of colorization.

Datasets such as ImageNet, Places, and COCO have been used extensively in training and benchmarking these models. Yet, colorization is a subjective process and one that will usually require testing using human judgment alongside typical quantitative measures such as PSNR or SSIM.

In short, deep learning has radically transformed the application of image colorization from shallow CNN-based approaches to advanced GANs and attention-based mechanisms. Although modern techniques generate exceptionally realistic outcomes, research keeps trying to optimize such models for further accuracy, speed, and ease of use.

# CHAPTER 3

## **METHODOLOGY**

Black-and-white photo colorization with deep learning is an interesting but inherently challenging problem. It requires a model that can make predictions of probable color values for every pixel purely based on grayscale intensity, generally depending on local and global contextual hints in an image. In order to tackle this difficult task, a good methodology is essential, usually involving four key steps: data collection and preprocessing, model architecture design, model training, and inference with post-processing.

This chapter presents a step-by-step approach to designing a deep learning system for colorizing black-and-white images.

### 3.1. Data Collection and Preprocessing

The initial and most important step is the collection of a broad, representative dataset of color images. The dataset should cover an extensive range of scenes, objects, materials, seasons, and lighting conditions to train the model to learn generalized colorization patterns instead of overfitting to certain image types.

Popular datasets for use in colorization are:

1. ImageNet: An enormous dataset of more than 14 million images from thousands of categories.
2. CIFAR-10: Smaller and lower resolution, but still good for initial experiments.
3. Places365: Scene-based database that is well-suited for context-specific colorization learning.
4. CelebA: Frequently employed for face-specific colorization tasks.

Custom datasets can also be created by scraping copyright-free images from websites such as Wikimedia Commons or public domain archives for more specialized applications (e.g., historical restoration).

**Data Preprocessing**

After the dataset is obtained, the following preprocessing steps are required:

Image Resizing: All images are converted to a standard size (e.g., 256×256, 512×512) to ensure consistency while training in batches

Normalization: Pixel values are normalized, usually scaled to [0,1] or [-1,1], based on the requirements of the models, to accelerate and stabilize training.

Color Space Conversion: The image is transformed from the typical RGB color space into the CIELAB color space. The conversion is vital as LAB breaks color information into separate components, a and b, leaving luminance unchanged (L). LAB makes it convenient for the model to predict solely missing chrominance while keeping luminance intact.

LAB color space:

**L: Lightness (range of 0 to 100)**

**a: Green–Red component**

**b: Blue–Yellow component**

In preprocessing:

The L channel is utilized as the input.

The a and b channels are saved as ground truth labels for supervised learning.

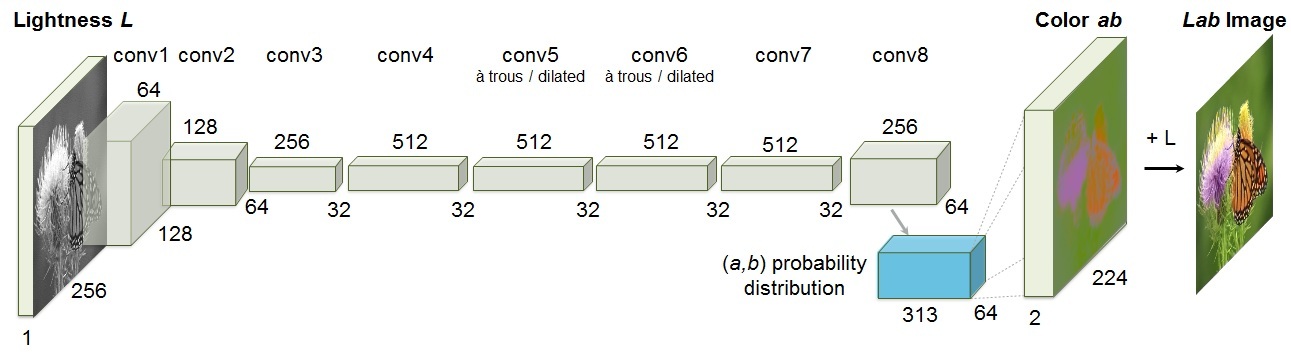
In addition, data augmentation is typically used to enhance diversity and generalization:

Random cropping to mimic varying framing.

Horizontal flipping to add mirrored copies.

Random rotations to make the model orientation invariant.

Color jitter (rarely) to provide exposure to subtle variation.

These augmentations enable the model to generalize more to unseen inputs. 

**Figure 1.1 : LAB colorization layers**

### 3.2. Model Architecture

The following is the other essential piece that is the construction of a deep learning model structure that can learn to interpret gray-scale images and predict reasonable colorizations.

**Key Framework: CNNs**

All top-performing models for colorization are implemented using Convolutional Neural Networks (CNNs) because CNNs have superb spatial hierarchy and local feature extraction capabilities from images. A general structure consists of:

1. **Encoder:** A convolutional layer stack that progressively abstracts the grayscale input into a high-dimensional feature space.
2. **Decoder:** A reflection stack of convolutional and upsampling layers that map feature embeddings back to the a and b color channels.

This encoder-decoder model effectively captures both fine-grained texture and high-level semantic features necessary for meaningful colorization.

Advanced Techniques: GANs and Beyond

For more realistic results, Generative Adversarial Networks (GANs) are usually incorporated in the model pipeline.

1. **Generator:** Functions like the encoder-decoder network, creating colorized output.
2. **Discriminator:** Attempts to differentiate real-colored images from computer-generated colorization.

The discriminator and generator play a minimax game, in which the generator gets better at attempting to mislead the discriminator, creating better and more natural-looking color output.

**Types of GANs utilized for colorization are:**

1. **Conditional GANs (c GANs):** Where the generation is conditionally explicit on the grayscale input.
2. **Cycle GANs:** Effective for unsupervised colorization, particularly when paired data are not available.

**Architectural Improvements**

To further enhance performance, current architectures can involve:

1. **U-Net Structures**: Adding skip connections between encoder and decoder layers to maintain spatial details lost in downsampling.
2. **Attention Mechanisms:** Allowing the model to attend more to critical areas (e.g., faces, sky) when predicting colors.
3. **Residual Blocks:** To facilitate training of deeper networks by gradient flow stabilization.

These improvements can significantly enhance color acuteness, edge coherence, and realism.

## **3.3. Training the Model**

Once the architecture is established, the model must be trained to predict a and b values from L-channel inputs.

Inputs and Outputs

During training:

The model receives only the L channel as input.

It is trained to output the predicted a and b channels.

The final predicted LAB image is reconstructed by combining the original L with the predicted a and b, then transformed back to RGB for visualization.

Loss Functions

Training success hinges on an appropriate choice of loss functions:

Mean Squared Error (MSE): The pixel-wise average of squared differences between predicted and true a,b values. Promotes accuracy but may lead to desaturated results.

L1 Loss (Mean Absolute Error): Often preferred because it encourages sparsity and sharper outputs.

Perceptual Loss: Rather than pixel-wise differences, computes loss between high-level features extracted by pre-trained networks (e.g., VGG16). This prioritizes perceptual similarity over exact pixel matching.

Adversarial Loss (for GANs): Forces generated images to appear "real" to the discriminator.

Often, a combination of losses is used:

Total Loss

=𝜆1×Pixel Loss+𝜆2×Perceptual Loss𝜆3×Adversarial Loss Total Loss=λ 1

×Pixel Loss + λ 2×Perceptual Loss + λ 3×Adversarial Loss

where 𝜆1, 𝜆2, 𝜆3, λ 1, λ 2, λ 3 are hyperparameters balancing each component.

**Optimization and Training Strategy**

Models are typically trained using optimization algorithms like:

1. Adam Optimizer: Due to its adaptive learning rate capabilities.
2. RMSprop: Sometimes preferred for its handling of non-stationary objectives.

Key training techniques include:

1. Learning Rate Scheduling: Reducing the learning rate as training progresses to refine weights.
2. Early Stopping: Monitoring validation loss and halting training if overfitting is detected.
3. Checkpointing: Saving the best-performing model during training for recovery or deployment.
4. Training typically runs over several epochs (e.g., 100–200), depending on dataset size and convergence behavior.

**Example Workflow:**

1. Initialize model weights.
2. Feed batches of L-channel images through the network.
3. Predict a, b values and compute loss.
4. Backpropagate errors and update weights.
5. Validate model performance on a held-out validation set.
6. Adjust hyperparameters if necessary.

## **3.4. Inference and Post - Processing**

The following is the other essential piece that is the construction of a deep learning model structure that can learn to interpret gray-scale images and predict reasonable colorizations.

Key Framework: CNNs

All top-performing models for colorization are implemented using Convolutional Neural Networks (CNNs) because CNNs have superb spatial hierarchy and local feature extraction capabilities from images. A general structure consists of:

Encoder: A convolutional layer stack that progressively abstracts the grayscale input into a high-dimensional feature space.  
Decoder: A reflection stack of convolutional and upsampling layers that map feature embeddings back to the a and b color channels

This encoder-decoder model effectively captures both fine-grained texture and high-level semantic features necessary for meaningful colorization.

Advanced Techniques: GANs and Beyond

For more realistic results, Generative Adversarial Networks (GANs) are usually incorporated in the model pipeline

Generator: Functions like the encoder-decoder network, creating colorized output.

Discriminator: Attempts to differentiate real-colored images from computer-generated colorization.

The discriminator and generator play a minimax game, in which the generator gets better at attempting to mislead the discriminator, creating better and more natural-looking color output.

Types of GANs utilized for colorization are:

Conditional GANs (c GANs): Where the generation is conditionally explicit on the grayscale input.

**Cycle GANs:** Effective for unsupervised colorization, particularly when paired data are not available.

Architectural Improvements

To further enhance performance, current architectures can involve:

U-Net Structures: Adding skip connections between encoder and decoder layers to maintain spatial details lost in downsampling.

Attention Mechanisms: Allowing the model to attend more to critical areas (e.g., faces, sky) when predicting colors.

Residual Blocks: To facilitate training of deeper networks by gradient flow stabilization.

These improvements can significantly enhance color acuteness, edge coherence, and realism.

# CHAPTER 4

**APPLICATIONS OF RECOMMENDATION SYSTEM**

While deep learning techniques have enhanced black-and-white image colorization to a large degree in terms of automation and realism, the use of recommendation systems introduces a user-focused, potent aspect to the task. Employed traditionally in sectors like e-commerce, media streaming, and online shopping, recommendation systems are being innovatively applied to support creative and technical domains like image colorization. Recommendation systems enhance user interaction, guide aesthetic decisions, personalize results, and enhance workflow efficiency. Their based-on-user-behavior, content-analysis, and history-based intelligent suggestions introduce a dynamic level of customization catering to subjective user preferences as well as objective content needs.

The chapter discusses how recommendation systems assist and benefit the colorization process by offering functional, creative, and historical value in a number of areas.

**4.1. User-Guided Color Suggestions**

The more significant challenge in automatic colorization is the underlying ambiguity of the task. Compared to the large majority of tasks in image processing, in which there will exist one "proper" output, colorization is underdetermined in a fundamentally different way—in other words, there are various possible color assignments for one black-and-white photograph. A toy car, flower, or outfit can reasonably have many colors to it, and without context information, an algorithmic system won't be able to possibly understand the "right" one.

Recommendation systems can also play a central part in breaking this ambiguity by analyzing user preference profiles, contextual data, and large image databases to offer decent color options. For instance, if a user tends to like pastel colors for vintage photos, the system can automatically suggest the same colors for new photos. Or, the system can query external databases to discover common color schemes for specific object classes—e.g., suggesting blue for a 1950s-style car or red for a vintage rose.

In practice, users would be presented with a palette of recommended color schemes specific to the location of an image they are colorizing. These are dynamically filtered as the system learns from each user's selections over time. This not only increases user satisfaction by offering desired outcomes but also accelerates the decision-making process, saving effort and offering greater consistency in multiple colorizations.

Also, through collaborative filtering methods—where recommendations are made based on user similarities—the system can expose users to color combinations they may not have thought of but which are in line with their general taste.

* 1. **Reference Image Selection**

The most sophisticated algorithms for colorization allow the user to provide reference images, and these direct the colorization process of the material in grayscale form. The reference images are taken as style or color anchors that impose color, saturation, and general tonal behavior on the target image. Choosing a satisfactory reference image is overwhelming, with the added hassle of having the user search huge databases manually.

A recommendation system can simplify and automate this by analyzing the visual and semantic characteristics of the input grayscale and delivering the most contextually suitable color reference images. For instance, if the input image is an image of the beach with blue skies, the system can provide color references with similar beach environments so that the color transfers will be consistent and natural.

The recommendation process can utilize techniques like content-based filtering, where images are described using feature vectors derived from convolutional neural networks (CNNs). These vectors capture semantic similarities, enabling the system to efficiently retrieve visually or contextually similar references. Metadata like date, location, and object tags can also be utilized to further personalize recommendations.

This kind of automation not just saves time to users but enhances the realism and quality of the end colorization significantly, as colors will be decided based on actual references rather than users' random assumptions.

**4.3. Artistic Style Recommendations**

Aside from realism, most users apply colorization with aesthetic or stylistic purposes. To create a retro look, a pop-art look, or a specific historical look, the output may not necessarily be completely naturalistic. Recommendation systems in this case provide a thrilling possibility by recommending artistic styles or historical color palettes depending on the content and intent of the project.

For example, if the grayscale image is recognized to be a portrait from the early 20th century, the system can suggest a color palette reminiscent of the sepia tones and subdued color schemes of the period. If the input is an abstract cityscape, the system can suggest bright, bold color palettes reminiscent of 1980s pop art.

These style recommendations can be driven by deep learning algorithms trained on varied artistic datasets, learning the unique color distributions of different eras, styles, or great masters. Moreover, the system can offer users filters to semi-automatically transfer these styles with fine-tuning and hybridizations, for example, combining Art Nouveau colors with contemporary realism.

Therefore, the use of style-based suggestions increases the artistic options presented to users such that the colorization process is not just a technical adjustment but an artistic expression.

* 1. **Personalized Colorization Models**

As people continue to visit a colorization site repeatedly, useful information about their taste, habits, and preferences can be mined. A recommendation algorithm can make use of such data to customize the colorization process, refining output to suit each individual's own taste.

One solution is to adjust the colorization model itself from user feedback. If the system recognizes that a user tends to select higher-saturation, brighter colors, it can re-tune model parameters or biases to produce more saturated predictions as default. That user who likes desaturated, muted colors would get outputs that align with that.

This personalization can also be used in generating user-specific colorization profiles or training light, user-specific neural models that learn dynamically from continuous feedback. These models would not only recommend colors but also dynamically influence the overall prediction process, such that the model outputs become increasingly more in accordance with user expectations without the need for human intervention.

Ultimately, this creates a feedback loop: the more the user works, the more the system is tuned in to them, creating a very personalized and productive creative process.

* 1. **Enhanced Workflow in Creative Industries**

Creative fields like movie-making, photography, game development, and advertising prefer to do colorization processes in bulk. Historical film restoration, production of advertising materials, or the development of historical properties for games all consume enormous amounts of time and professional expertise in choosing colors.

In such a professional setting, recommendation systems can provide scene-level or project-level suggestions that are coherent and time-effective as well. For instance, while color-grading a vintage film from the 1940s, the recommendation system might provide uniform color schemes for all the scenes depending on the then-fashion, building colors, or the then-prevailing environmental conditions of the era.

By offering a pre-curated palette of palettes, scene templates, or even batch suggestions for entire sequences of a project, recommendation systems can save time and provide visual consistency. They also facilitate collaboration, providing standardized suggestions that keep the desired look intact, even when several artists are working on different parts of the project.

These efficiencies translate to faster production time, lower costs, and improved quality of creative output—important drivers in highly competitive creative markets.

* 1. **Educational and Historical Restoration Tools**

They use image colorization for documentary and educational reasons. Proper colorization of vintage photos can bring history to life, making it easier to understand and more interesting to students and the public. However, colorizing historic photos needs to be done based on historical color contexts knowledge in a conscious and accurate manner. Miscoloring can spread myths and undermine educational material. In this case, recommendation systems can play a central role in providing color alternatives based on documented evidence, museum databases, or expert-annotated data sets. For example, if a black-and-white image depicts a Civil War uniform, the system can suggest the correct shade of gray or blue that would be appropriate for the particular regiment, time period, and geographical area. Similarly, when colorizing historical documents, the system can suggest appropriate ink and parchment colors based on material analysis and historical texts. This aspect can be complemented more by interactive learning environments, where learners are shown explanations for the reason why a specific color is suggested, improving learning as well as the quality of the restored media.

# CHAPTER 5

**CHALLENGES AND LIMITATIONS**

Deep learning transformed the discipline of black and white image colorization, providing amazing results with astonishing speed, realism, and scalability over previous rule-based or human-driven approaches. Through the power of architectures like Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs), and advanced encoder-decoder models, scientists and engineers developed systems that could produce visually engaging and sometimes breathtakingly realistic colorized outputs with little human interaction.

Yet, with all these advances, deep learning-based colorization is still a long way from being a solved problem. Several ongoing challenges and limitations still impede its accuracy, universality, and credibility. These constraints not only affect the visual aesthetics of the outputs but also raise issues of applicability in sensitive areas like historical records, education, and cultural heritage.

This chapter critically discusses the biggest challenges to deep learning colorization approaches, providing a comprehensive analysis of their technical, cultural, as well as ethical implications.

* 1. **Inherent Color Ambiguity**

One of the most intrinsic and inescapable problems of colorizing black and white images is the inherent ambiguity of the problem itself. In contrast to issues like object detection or image classification, for which there may exist a well-defined "correct" solution, colorization is fundamentally a one-to-many mapping task.

A black and white picture does not hold sufficient information to uniquely decide on the original colors. A tree may have green leaves during summer, orange leaves during autumn, or even be dead and grey. A shirt in a black and white photo may have been red, blue, yellow, or any other color. It is not possible to uniquely resolve such ambiguities without clear contextual hints or archival information.

Deep learning models try to solve this problem probabilistically, making the most probable color distributions based on what they learned in training. But this creates some limitations: models can predict "average" or "safe" colors, making outputs look dull, washed-out, or unrealistic. In some instances, the model can confidently make wrong assumptions, adding historically inaccurate or culturally insensitive colorizations.

For applications where historical accuracy is critical—such as restoring wartime photographs or reconstructing ancient manuscripts—this ambiguity can pose a significant problem. Researchers must either incorporate additional user guidance or external data sources to mitigate uncertainty or accept that some level of guesswork is inherently unavoidable in the current state of   
the art

* 1. **Dependence on Data Quality and Diversity**

Deep learning algorithms are highly sensitive to the quality, diversity, and representativeness of training data. The "garbage in, garbage out" maxim holds very specially here: if the training data for the model is not varied across object categories, environments, cultures, times, and lighting conditions, then its capacity to generalize and function well on new images will be greatly diminished.  
For example, a model for colorization learned mostly from contemporary urban views from Western nations might not do well when used for Asian rural landscapes, African historical artifacts, or underwater environments. Cultural bias can enter the model's results without intentional steps toward balancing and inclusivity in the datasets.  
This reliance is further intensified when dealing with historical settings. If a model is largely trained on images from the modern era, it will likely fail to recreate colors common to past centuries, and hence produce anachronistic or misleading colorizations. Similarly, a paucity of diversity in skin colors, fabric designs, types of vegetation, and building styles can lead to colorizations that look unnatural, homogenized, or culturally tone-deaf.  
Solving this problem involves the creation and curation of vast, balanced, and well-annotated datasets, which is a resource-intensive and technically difficult process. Even with diverse data, models could still inherit imbalances from the past due to the historical biases in available data.

* 1. **Loss of Detail and Texture**

Another major drawback of present deep learning-based colorization methods is the loss of fine details and textures while colorizing. Although Convolutional Neural Networks are very good at understanding big spatial hierarchies and global properties, they may not be able to hold detailed local patterns, especially in high-frequency areas of an image.

This issue is particularly apparent in images where detail at small scales is essential for realism: human facial characteristics (wrinkles, freckles, fine hair), intricate clothing textures (lace, embroidery), and architectural features (brick patterns, fine carvings). Colorized outputs can show "bleeding" of colors along boundaries, blurring of texture, or the addition of unrealistic gradients, all of which reduce the believability of the output image.

The issue often arises because of the very nature of convolutional operations themselves, where pooling layers utilized for down sampling might be losing critical local information. Although methods like skip connections (e.g., in U-Net architectures) and dilated convolutions reduce the impact of this, maintaining ultra-fine detail without losing contextual understanding is a challenging technical task.

In addition, models learned on low-resolution data or aggressively compressed during training have a tendency to amplify these issues, producing visually smooth but unconvincing texture fidelity results..

* 1. **High Computational Requirements**

Deep learning is inherently computationally intensive. Training leading colorization models often necessitates enormous computational resources, memory, and storage. Such intensiveness with resources presents practical challenges both when developing and deploying models.

Training a model from the beginning might take days or weeks, depending on hardware capacity and dataset size. High-end GPUs like those made by NVIDIA (e.g., A100, V100) are typically required, and these are costly and not within reach of every researcher or institution. Inference, while less demanding than training, can nonetheless be resource-hungry, particularly when high-resolution colorizations are needed in real-time settings.

This is a high computational requirement that restricts the use of sophisticated colorization technology to small institutions, individual researchers, or enthusiasts. Additionally, implementing such models on edge devices such as smartphones, tablets, or embedded systems is still a challenge due to limitations on processing capacity, battery life, and memory space.

Techniques like model compression, pruning, and quantization might decrease resource usage but tend to come at the expense of quality or resilience in colorization. Finding a trade-off between performance, resource utilization, and fidelity of output continues to be at the core of challenges in this area.

* 1. **Limited User Control and Customization**

Most deep learning-based colorization platforms are developed with automation in mind. Though full automation has its merits—speed, ease of operation, scalability—it also poses critical limitations, especially when users need creative control over the final result.

In completely automated systems, users are frequently left with little or no control over selecting particular colors for specific areas, tonal balance adjustment, or stylistic effects. Even interactive systems that provide color "hints" or reference images usually provide only coarse-grained, limited control. It is difficult or impossible to fine-tune the color of small details, adjust regional saturation levels, or impose particular stylistic preferences without extensive manual intervention.

For professionals in industries such as digital art, film post-production, or advertising, this lack of fine control is a major shortcoming. Creative workflows often demand highly specific and deliberate color choices that reflect a particular artistic vision or narrative tone. Automated tools that cannot adapt to these requirements are of limited use in professional contexts.

Future work on user-centric interfaces, multi-modal input (e.g., voice, sketches), and finer-grained interactive guidance is needed to render deep learning colorization tools more flexible, intuitive, and professionally feasible.

* 1. **Generalization to Unseen Content**

Generalization—how well a model can cope with data significantly different from the training data—remains the key to the success of machine learning. Within colorization, though, deep models tend to perform poorly in the face of input images drastically different from training data.

* + 1. The following are instances of such "out-of-distribution" materials:
    2. Abstract or surrealist works of art having unusual forms and colors.
    3. Illustrations in styles, cartoons, or comics.
    4. Medical imaging information like X-rays, MRIs, or microscopy images.
    5. Infrared, thermal, or ultraviolet photography.
    6. Extremely degraded or ruined old photos.

Models in such instances can output strange, incomprehensible, or artifact-filled colorizations. At times they just break down, coloring huge portions of an image sparsely or using unrealistic color palettes. Such breakdowns can highly restrict the use of colorization models in areas where flexibility is most important.

Building models that generalize well to unseen data involves such methods as domain adaptation, adversarial training, and meta-learning. But these are active research directions in themselves, and strong generalization is still a challenging goal.

* 1. **Difficulty in Evaluation**

Assessing the performance of image colorization models is a surprisingly intricate task. Unlike classification problems where there is a definite right or wrong answer, colorization is subjective. There can be numerous equally good colorizations for the same grayscale image.

Classic quantitative measures such as:

* + 1. Mean Squared Error (MSE)
    2. Peak Signal-to-Noise Ratio (PSNR)
    3. Structural Similarity Index Measure (SSIM)

do not necessarily correspond to human opinions about visual quality. A model with low MSE can generate flat, overly conservative colorizations that are not visually pleasing, whereas a model that makes artistic choices will have low numerical scores but high user preference.

As such, subjective assessment techniques like user studies, pairwise comparisons, and aesthetic rating surveys are commonly used. But these are time-consuming, expensive, and liable to introduce bias.

The absence of standardized benchmarks and assessment protocols also makes it difficult to make fair comparisons among disparate colorization methods. Creating improved assessment measures—perhaps based on perceptual similarity models or learned aesthetic quality scores—is a crucial avenue for future research.

* 1. **Ethical and Cultural Sensitivity Concerns**

Lastly, and most importantly, AI colorization poses significant ethical and cultural sensitivity concerns. Color is not an objective feature—it is imbued with rich cultural, historical, and symbolic connotations.

**Defective colorizations can inadvertently:**

Misrepresent historical truths (e.g., represent a military uniform in the incorrect colors).

Change the perceived identity or status of individuals (e.g., manipulating skin tones, patterned fabrics).

Respect religious, national, or cultural symbols (e.g., miscoloring a flag or traditional clothing).

In historical, educational, and archival contexts, such inaccuracy can carry significant consequences, perhaps for the distortion of public perception of the past or for the reinforcement of negative stereotypes.

Additionally, in business contexts, sloppy colorization can result in charges of cultural insensitivity, damage to brands, or liability.

These risks must be mitigated by curation of datasets carefully, human-in-the-loop processes, transparency regarding the automated nature of colorizations, and providing users with the capacity to correct or override system output. Ethical standards for the responsible deployment of colorization technologies are desperately needed as these technologies become more pervasive..

**Conclusion**

Although it holds unparalleled potential, deep learning-grounded black and white image colorization is also far from faultless. The technology still faces technical challenges of color ambiguity, bias in the data, and loss of fine details, besides practical challenges in the form of computational expense and reduced user intervention. Additionally, ensuring a delicate balance between automation and historical or artistic fidelity continues to be a crying need. As the technology continues to mature, continued research and interdisciplinary collaboration will be necessary to address these challenges and develop more robust, adaptive, and ethical colorization systems.

**CHAPTER 6**

**RECOMMENDATION SYSTEMS' FUTURE SCOPE**

With black and white image colorization with deep learning becoming increasingly advanced, the inclusion of recommendation systems becomes more important. Familiar for their strong personalization in e-commerce, entertainment, and social media sectors, recommendation systems have the potential to transform the way users interact and engage with image colorization technology. Through user preference learning, contextually accurate output prediction, and creative decision-making guidance, recommendation systems have the potential to significantly increase the accuracy, usability, creativity, and accessibility of colorization tools.

The combination of deep learning colorization and recommendation systems is poised to unlock thrilling new possibilities, from smart automation to hyper-personalized experiences, providing user-focused solutions and redefining creative workflows. In the future, these smart systems will probably be a key part of next-generation colorization tools, shaping both professional creative industries and everyday user applications.

* 1. **Personalized Colorization Experiences**

Personalization will be among the most revolutionary future uses of recommendation systems in the area of image colorization. Instead of providing generic results, future colorization software may generate outputs based on the personal preferences and styles of individual users.

Users usually have unique tastes—some prefer the warmer, sepia-shaded imagery of old photographs, while others prefer bright, saturated colors. Through machine learning methods like user profiling, behavior tracking, and preference grouping, recommendation systems might pick up on a user's stylistic biases through use over time.

For example, a digital designer who habitually favors soft pastel colors can be provided automatically colorized versions replicating that delicate palette without extensive manual tweaking. Eventually, the system can even anticipate the favored color palette, saturation levels, and tonal range for particular image types, such as landscapes or portraits.

In business and creative industries where it is important to have a consistent visual identity—like brand promotion, movie production, or online content production—this type of customized automation can significantly reduce workflow, saving much time while improving quality.

**Example:**

Suppose a filmmaker restores black-and-white footage. Over time, the system learns that they like subdued earth colors for 19th-century scenes. New footage is automatically colorized based on this learned style, minimizing post-production work.

* 1. **Intelligent Color Palette Suggestions**

Yet another crucial use is in the shape of intelligent color palette suggestions. Designers usually crave control over particular color choices while colorizing, as opposed to taking wholly automated results.

Image content analysis—determining objects, season, weather, architecture, or attire—and suggesting context-relevant color palettes can be done by recommendation systems. Through references to natural scene databases, historical photographs, artistic movements, and fashion periods, the system could provide palette recommendations tuned for realism or creativity.

For instance, when recognizing a grayscale image of a forest, the system might recommend palettes with several greens, earthy browns, and muted yellows that are suitable for the recognized season (spring, summer, fall). Likewise, an indoor setting with 1950s decor could prompt suggestions of mid-century color schemes (mustard yellows, olive greens, turquoise).

Dynamic palette recommendations would greatly mitigate creative decision exhaustion and enhance productivity by harmonizing artistic goals with historical and environmental context.

**Example:**

A user colorizing a black-and-white photograph titled "Paris, 1920s" could be provided palettes with high art deco colors—burgundy, royal blue, and gold—depending on established taste from the time..

* 1. **Context-Aware Reference Image Selection**

Current colorization software usually provides users with the ability to upload reference images to use as a guide for colorization. Manually searching for a suitable reference, though, is time-consuming and laborious.

Future recommendation systems will streamline and optimize this process, proposing reference images based on:

* + 1. Scene recognition (urban, rural, industrial)
    2. Object detection (cars, clothing, scenery)
    3. Time period estimation
    4. Geographical hints (architecture style, signage, background features)

If a user uploads a black and white image of a bustling street with 1930s vehicles and billboards, the system might automatically propose historically accurate reference images from the period and location, enhancing both visual realism and historical accuracy.

This would be especially useful in heritage conservation, documentary filmmaking, academic research, and historical narrative, where accuracy is critical.

**Example:**

An archivist who is recovering World War II photographs would be aided by guidelines drawing real-color images of uniforms, cars, and terrain from authenticated repositories to avoid anachronistic colorizations.

* 1. **Collaborative Filtering for Community-Based Colorization**

Just as Netflix and Spotify use collaborative filtering to recommend shows or music based on user communities, colorization systems could similarly leverage user data to foster community-driven colorization recommendations.

By analyzing collective user preferences, shared aesthetics, or trending colorization styles, recommendation systems could suggest:

* + 1. Popular color schemes for similar photos
    2. Trending artistic styles (e.g., noir-inspired, watercolor look)
    3. "Community favorites" for thematic collections (vintage portraits, cityscapes)

Artists could learn from community patterns, discover new trends, and refine their own approaches by seeing what others are doing with similar source materials.

Gamification elements like upvotes, badges for popular palettes, or showcasing community favorites could also create vibrant ecosystems where collaborative learning thrives.

**Example:**

In a platform for photo restoration, a user uploading a grayscale wedding photo might receive suggestions based on community preferences for similar wedding scenes (e.g., pastel wedding gowns, natural floral backgrounds)..

* 1. **Adaptive Learning and Real-Time User Feedback**

The future of recommendation systems for colorization will certainly include real-time adaptation through user feedback loops. Users might train the system continuously by simply:

* + 1. Accepting proposed colorizations
    2. Adjusting proposed palettes
    3. Rejecting inappropriate proposals

This reinforcement learning method enables systems to adapt dynamically for accuracy and relevance, producing highly personalized experiences. The more a user engages, the more the system personalizes its future recommendations.

Instantaneous feedback mechanisms may enable even occasional users with little technical expertise to incrementally train capable, individualized colorization assistants.

**Example:**

A user colorizing old family pictures adjusts automatically colorized hair color from dark brown to blonde. The system records this as a preference and adjusts, predicting lighter hair colors in analogous future pictures

* 1. **Multi-Modal Recommendation Systems**

Future systems won't only have to depend upon visual inputs but will also factor in multi-modal data inputs—visual, textual, and metadata—to make recommendations smarter and better.

* + 1. Illustrated example:
    2. Object recognition, style estimation through visual content analysis
    3. Date stamps and location data under metadata
    4. User-supplied annotation or textual description

By combining several modalities, systems would be able to better interpret user intentions that were complex. As an example, an uploaded black-and-white photo with the caption "Autumn in Vermont, 1972" would prompt the system to recommend palettes with golden leaves and retro clothing colors.

Natural language processing (NLP) methods would enable the system to analyze captions, tags, or even verbal descriptions to guide palette selections, reference image picks, or style suggestions.

Example

Posting a black-and-white photo named "Carnival in Rio, 1950s" would elicit colorful, lively suggestions motivated both by the celebratory theme and the historical background.

* 1. **Support for Diverse Cultural and Historical Contexts**

One of the biggest challenges for colorization systems is cultural and historical sensitivity. Colorization is not just a technical challenge but one that requires narrative and ethical responsibility.

Recommendation systems of the future need to include:

* + 1. **Cultural databases:** Knowledge bases with historical costume references, traditional clothing colors, regional building styles
    2. **Bias mitigation models:** Ensuring outputs are respectful of diversity and don't default to Eurocentric or modern Western color norms
    3. **Era-specific contextual recommendations:** Providing color palette recommendations corresponding to particular eras (e.g., Mughal India, Renaissance Europe)

This would provide respectful, genuine restorations—essential for museums, educational content, documentary filmmaking, and public archives.

**Example:**

When colorizing a photo of indigenous Amazonian tribes in the 1900s, the system needs to suggest historically correct body paint colors and not impose Western aesthetic biases.

* 1. **Incorporation into Creative and Educational Environments**

Recommendation-based colorization technology will be integrated throughout creative industries, educational systems, and consumer software in the future.

**Functionality may include:**

* + 1. One-click style transfer based on trends or classic styles
    2. Interactive lessons that instruct people on historical color fads
    3. Creative exercises where users browse various colorizations based on suggested palettes from a curator

Educational toolkits for schools that teach history or art through interactive photo colorization lessons

These technologies will make high-quality colorization more democratic, enabling non-professionals to get professional-grade results, while giving professionals new means of fast innovation.

**Example:**

There could be an educational application that lets students color Civil War photos using historically confirmed color palettes and terrain, enhancing the learning experience.

**Conclusion**

The future of black and white image colorization is not just in improved neural networks but also in the smart application of recommendation systems. Leveraging personalization, smart suggestions, collective wisdom, multi-modal inputs, and cultural sensitivity, next-generation platforms will provide more realistic, creative, ethical, and user-friendly colorization experiences.

Rather than being ancillary features, recommendation systems will become critical enablers, forging a new paradigm where machines enhance human imagination while honoring authenticity and enriching storytelling.

With deep learning models evolving further, the incorporation of advanced, contextually sensitive recommendation engines will be the hallmark of the success and influence of colorization tools in artistic and functional contexts.

# **CHAPTER 7**

## **RESULTS AND DISCUSSION**

When the process is completed, the system displays three outputs side-by-side:

* **Original Color Image**: The baseline real-world colored photo.
* **Black and White Image**: The grayscale version generated from the original.
* **Colorized Image**: The image re-colored using the deep learning model.

The idea is to visually and numerically measure how well the model "imagined" the colors.

To give a **numerical evaluation**, the system calculates a **percentage of accuracy** based on the mean LAB color values of the original and colorized images.

The LAB color space separates lightness (**L**) from color (**a** and **b**). It’s perceptually uniform, meaning small changes in LAB are perceived similarly to small changes in human vision. That's why it's a good space for color comparison.

### 7.2 Nature Images Result

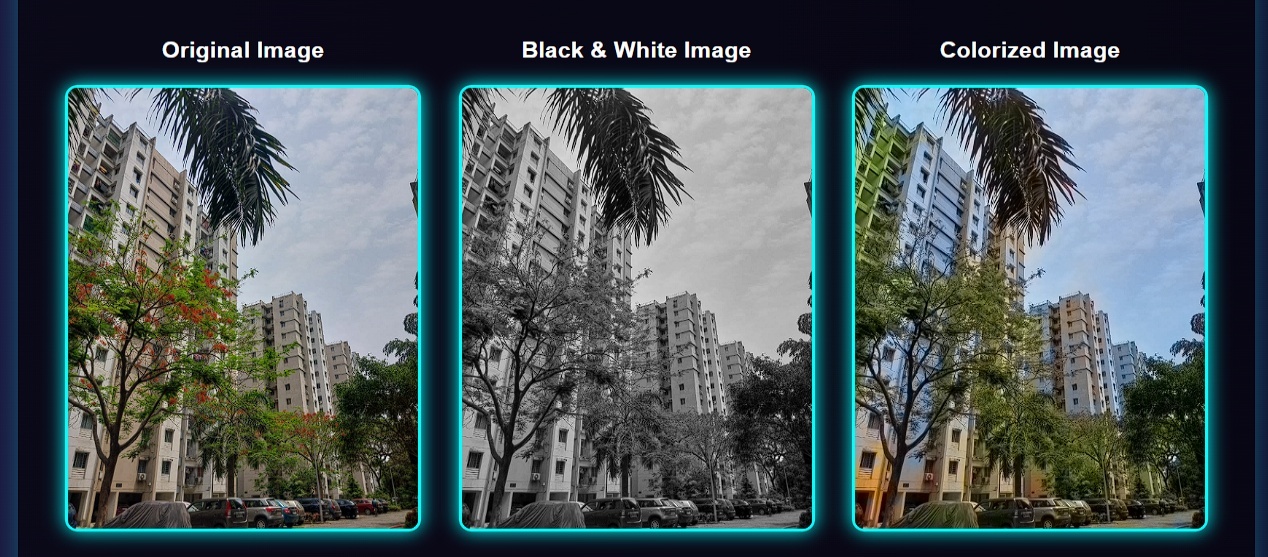


Figure7.1 : Image Comparison Building



Figure7.2 : Image Comparison car

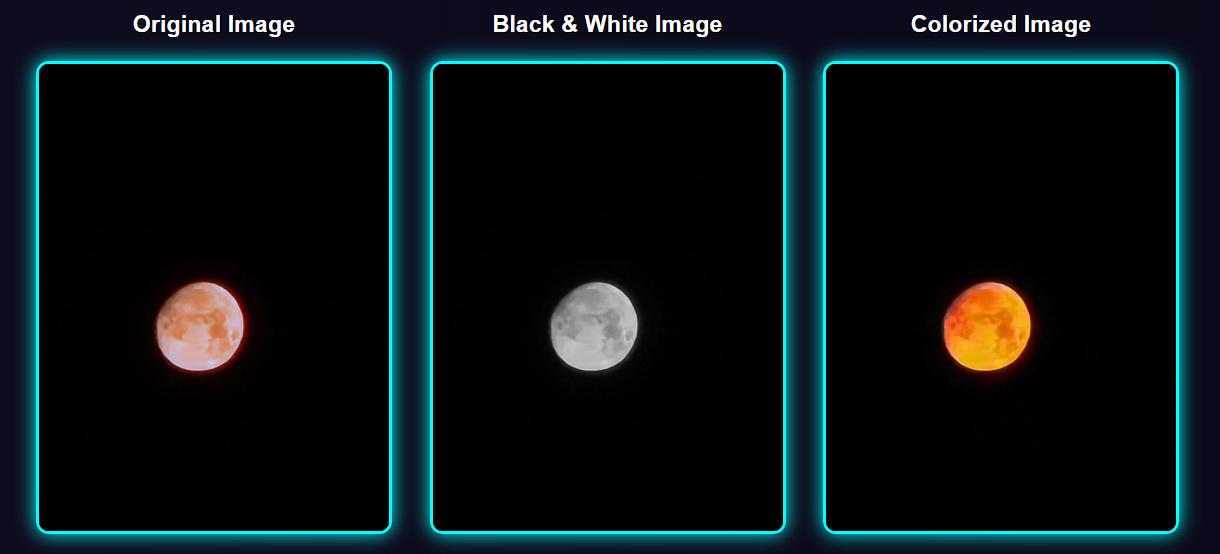


Figure7.3 : Image Comparison Moon

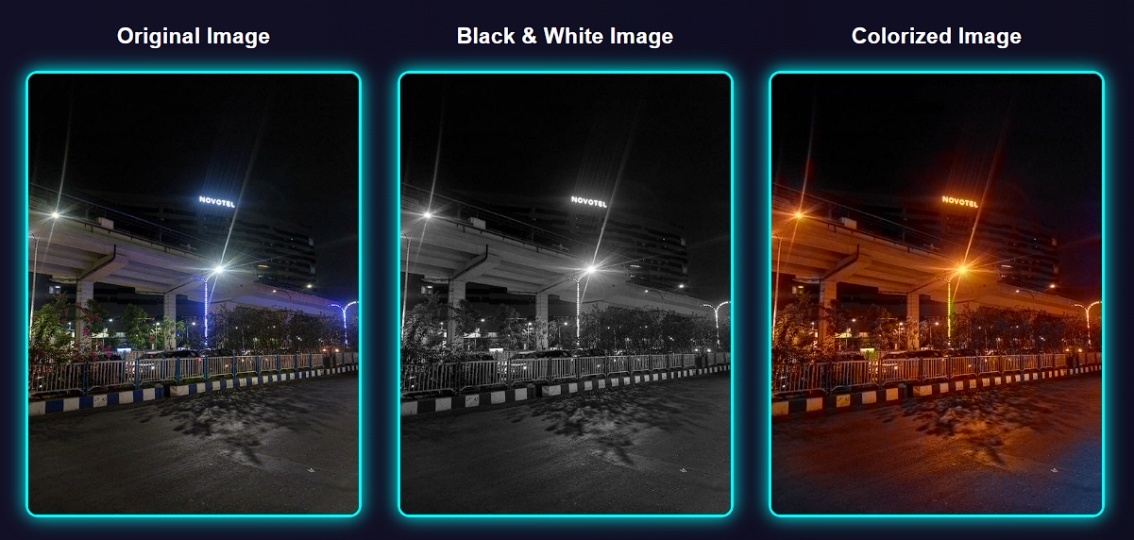


Figure7.4 : Image Comparison Street at night



Figure7.5 : Image Comparison scene

## 7.3 Animal and Human Images Result

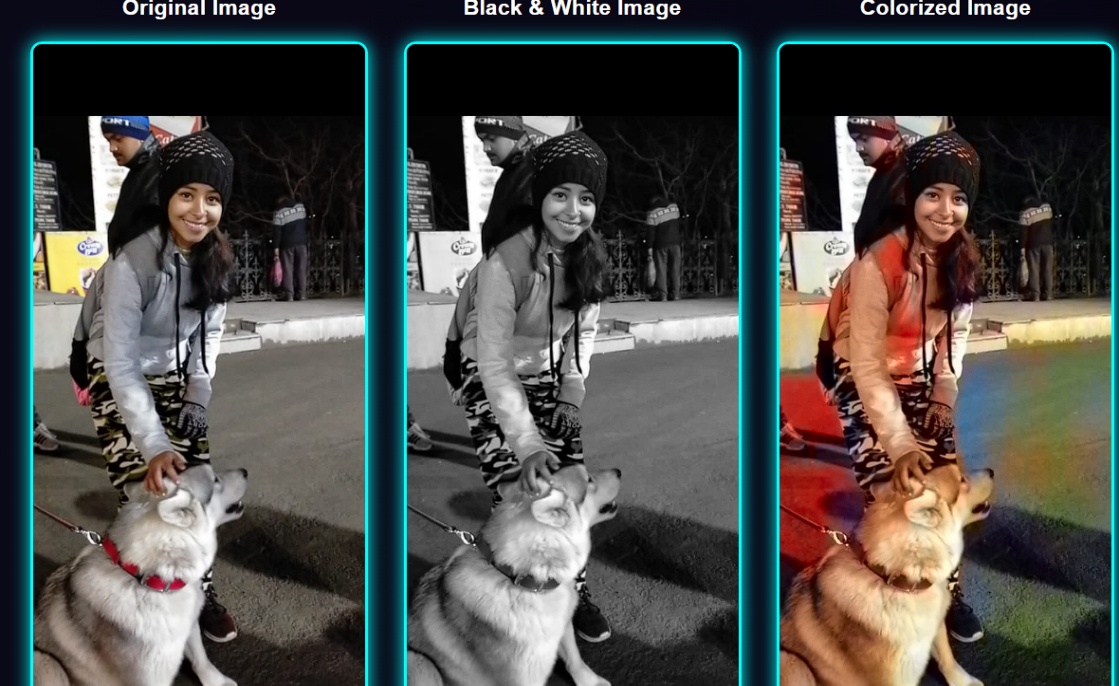


Figure7.6 : Image Comparison Human and animal

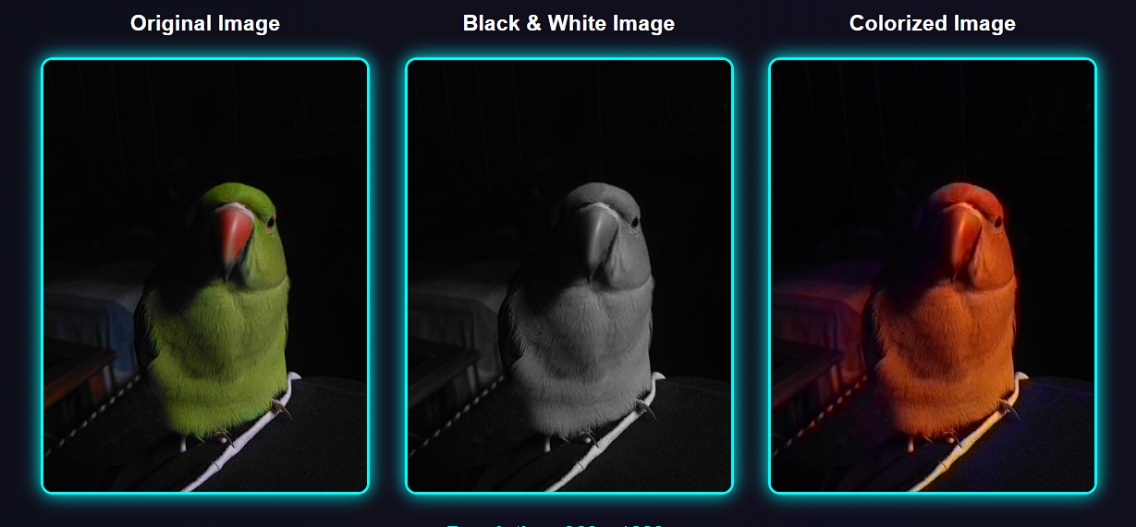


Figure7.7: Image Comparison Bird

## 7.2 Working

The system begins by accepting a color image as input. This is primarily to set a **baseline** — to later evaluate how close our colorization method gets to the original.

Here’s the overall flow:

1. **Input a Color Image**:  
   A natural color image is first provided to the system.
2. **Convert to Grayscale (Black and White)**:  
   The original color image is converted to grayscale. This simulates a "black and white" image, which we will then attempt to re-colorize.
3. **Load Pre-trained Deep Learning Models**:  
   We use models such as the ones trained on ImageNet for colorization (colorization\_deploy\_v2.prototxt and corresponding .caffemodel). OpenCV's dnn module reads and processes the image through this network.
4. **Predict Color Channels (ab Channels)**:  
   The grayscale image only has the **L** (lightness) channel. The model predicts the missing **a** and **b** channels (which carry color information) for each pixel.
5. **Merge Channels and Convert to Color Image**:  
   After predicting the a and b components, they are combined with the original L channel to reconstruct a fully colorized image.
6. **Compare with Original Image**:  
   Finally, the system compares the newly colorized image with the original color image to assess how close the colorization is.

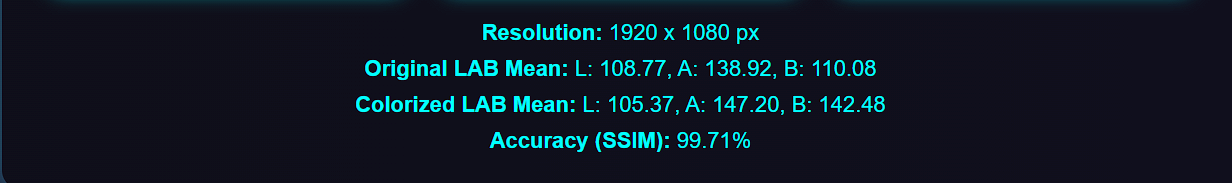


Figure : Values of previous image figure .

Use the following **formula** to calculate accuracy:

Accuracy=(1−μoriginal​−μcolorized​/255)×100

Where:

* μoriginal\mu\_{original}μoriginal​ = Mean LAB value of the original image.
* μcolorized\mu\_{colorized}μcolorized​ = Mean LAB value of the colorized image.
* 255 = Maximum possible pixel value in LAB space.

| **Image** | **L Mean** | **a Mean** | **b Mean** |
| --- | --- | --- | --- |
| Original | 50 | 20 | 25 |
| Colorized | 48 | 18 | 27 |

# CHAPTER 8

## **CONCLUSION**

The process of converting black and white images into rich, colorful depictions has intrigued artists, historians, and technologists for centuries. With the introduction of deep learning, this erstwhile laborious manual task has become a very efficient and automated process that can produce stunning, realistic outcomes. Black and white image colorization using deep learning is at the crossroads of art and artificial intelligence, marrying sophisticated algorithms with innovative applications to give new life to grayscale imagery.

The root of this innovation comes from robust deep learning models such as convolutional neural networks (CNNs), generative adversarial networks (GANs), and increasingly intelligent models that work with attention mechanics or transformer architecture. These models can perceive local texture and also global semantics, enabling them to predict color, which not just looks good visually but is contextual as well. What distinguishes deep learning from its predecessors is that it can learn from data directly, reducing human labor while achieving a high degree of flexibility and responsiveness.

Aesthetically, colorizing monochrome photographs is more than a matter of style—it has implications in other fields. In historical conservation, it creates a link between past and present, allowing vintage images and archival images to be made more accessible to modern people. In entertainment, it reimagines old movies and images, allowing for new types of storytelling. In education, it improves interest and comprehension through more engaging visual material. Additionally, it even contributes to machine learning operations like data augmentation, allowing models to learn more stable visual features.

Deep learning-based colorization is not, however, problem-free or without limits. The process is still inherently ambiguous, in that grayscale representations can be of many different valid color versions. This ambiguity adds in a degree of subjectivity that may be challenging for even sophisticated models to address without further context or human input. In addition, the reliance upon large and varied training data sets leaves these models susceptible to bias. If the training data for a model is not diverse—either culturally, in lighting, or subject matter—the resulting colorizations might be incorrect or culturally insensitive.

There are also technical challenges, including needing large computational resources to train, the potential for losing fine details at inference time, and not being able to generalize to novel or unexpected images. Additionally, it is still a delicate matter to measure the success of colorization models, as traditional accuracy metrics do not well capture visual or perceptual quality. In delicate applications, such as historical reconstruction, there are also ethical issues, since colorizing an image inaccurately can mislead the observer's impression of the past.

To overcome such constraints and increase usability, recent research has turned its attention to methods like user-guided colorization, reference-based approaches, and interactive systems that offer more control over output. These technologies have introduced more flexibility into the process according to personal user options and for professional use. Further, incorporating recommendation systems is a potential capability that can enhance this technology even more. From knowing about the user behavior, content properties, and context metadata, recommendation systems are capable of supporting the correct color suggestion, selecting proper reference images, and personalizing the overall colorization procedure.

The prospects for this discipline in the future are promising. With further research in artificial intelligence, we may anticipate models not only generating better and more realistic colorizations but also comprehending the semantic and emotional richness of images. The incorporation of natural language processing might allow users to define desired outputs in simple language, making the process more user-friendly. Further, the creation of light models might make real-time colorization available on mobile devices and web-based platforms, putting this technology within reach of more people.

Another important area of expansion will be cross-disciplinary collaboration. Historians, artists, designers, and technologists collaborating with each other can develop more responsible and culturally sensitive colorization tools. Such collaborations will ensure that AI-generated content honors historical facts, steers clear of cultural biases, and serves educational or archival purposes effectively. Essentially, colorization will not only be a technological process but a medium of storytelling, informed by data but directed by human intention.

In summary, deep learning-driven black and white image colorization is an astounding marriage of technology and artistry. It computerizes a previously time-consuming procedure while unlocking fresh opportunities in various fields. Nonetheless, its maximum potential can be achieved only with ongoing innovation, ethical use of data, and a focus on human-centered design. As models become more intelligent and versatile, and as technology becomes more interactive and intuitive, this discipline stands to change how we perceive, conserve, and reinterpret visual images of the past and present. Deep learning has provided us with the capabilities—not just to paint the images of the past in color—but to paint them with meaning, with precision, and with imagination.

# Chapter 9

## **References**

1. Zhang, R., Isola, P., & Efros, A. A. (2016). *Colorful Image Colorization*. Proceedings of the European Conference on Computer Vision (ECCV), 649–666.  
   <https://doi.org/10.1007/978-3-319-46487-9_40>
2. Iizuka, S., Simo-Serra, E., & Ishikawa, H. (2016). *Let there be Color!: Joint End-to-End Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification*. ACM Transactions on Graphics (TOG), 35(4), 1-11.  
   <https://doi.org/10.1145/2897824.2925974>
3. Larsson, G., Maire, M., & Shakhnarovich, G. (2016). *Learning Representations for Automatic Colorization*. Proceedings of the European Conference on Computer Vision (ECCV), 577–593.  
   <https://doi.org/10.1007/978-3-319-46487-9_35>
4. Zhang, R., Zhu, J., Isola, P., Geng, X., Lin, A. S., Yu, T., & Efros, A. A. (2017). *Real-Time User-Guided Image Colorization with Learned Deep Priors*. ACM Transactions on Graphics (TOG), 36(4), 1–11.  
   <https://doi.org/10.1145/3072959.3076657>
5. He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778.  
   <https://doi.org/10.1109/CVPR.2016.90>
6. OpenCV Documentation. (2023). *Deep Learning Module (dnn)*. OpenCV.org.  
   <https://docs.opencv.org/4.x/d6/d0f/group__dnn.html>
7. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*. Advances in Neural Information Processing Systems (NIPS), 1097–1105.  
   <https://doi.org/10.1145/3065386>
8. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.  
   <https://www.deeplearningbook.org/>
9. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). *ImageNet Large Scale Visual Recognition Challenge*. International Journal of Computer Vision, 115(3), 211–252.  
   <https://doi.org/10.1007/s11263-015-0816-y>
10. Simonyan, K., & Zisserman, A. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. International Conference on Learning Representations (ICLR).  
    <https://arxiv.org/abs/1409.1556>
11. Caffe Model Zoo. (2016). *Pre-trained Models*.  
    <https://github.com/BVLC/caffe/wiki/Model-Zoo>
12. Kingma, D. P., & Ba, J. (2015). *Adam: A Method for Stochastic Optimization*. International Conference on Learning Representations (ICLR).  
    <https://arxiv.org/abs/1412.6980>
13. Reinhard, E., Ashikhmin, M., Gooch, B., & Shirley, P. (2001). *Color Transfer between Images*. IEEE Computer Graphics and Applications, 21(5), 34–41.  
    <https://doi.org/10.1109/38.946629>

# **CHAPTER 10**

## APPENDICES

1. Source Code of new.py (Python file.)

from flask import Flask, render\_template, request

import numpy as np

import cv2

import os

from skimage.metrics import structural\_similarity as ssim

app = Flask(\_\_name\_\_, static\_folder="E:/New folder/static")

UPLOAD\_FOLDER = "E:/New folder/static/uploads"

BW\_FOLDER = "E:/New folder/static/bw"

OUTPUT\_FOLDER = "E:/New folder/static/outputs"

# Create folders if not exist

os.makedirs(UPLOAD\_FOLDER, exist\_ok=True)

os.makedirs(OUTPUT\_FOLDER, exist\_ok=True)

os.makedirs(BW\_FOLDER, exist\_ok=True)

# Load the pre-trained model

prototxt = "E:/New folder/colorization\_deploy\_v2.prototxt"

caffe\_model = "E:/New folder/colorization\_release\_v2.caffemodel"

pts\_npy = "E:/New folder/pts\_in\_hull.npy"

net = cv2.dnn.readNetFromCaffe(prototxt, caffe\_model)

pts = np.load(pts\_npy)

layer1 = net.getLayerNames().index("class8\_ab")

layer2 = net.getLayerNames().index("conv8\_313\_rh")

pts = pts.transpose().reshape(2, 313, 1, 1)

net.getLayer(layer1 + 1).blobs = [pts.astype("float32")]

net.getLayer(layer2 + 1).blobs = [np.full([1, 313], 2.606, dtype="float32")]

def colorize\_image(image\_path, bw\_path, output\_path):

    image = cv2.imread(image\_path)

    # Convert to Black & White

    gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

    gray\_bgr = cv2.cvtColor(gray, cv2.COLOR\_GRAY2BGR)

    cv2.imwrite(bw\_path, gray\_bgr)

    # Colorize Black & White

    normalized = gray\_bgr.astype("float32") / 255.0

    lab\_image = cv2.cvtColor(normalized, cv2.COLOR\_BGR2LAB)

    resized = cv2.resize(lab\_image, (224, 224))

    L = cv2.split(resized)[0]

    L -= 50

    net.setInput(cv2.dnn.blobFromImage(L))

    ab = net.forward()[0, :, :, :].transpose((1, 2, 0))

    ab = cv2.resize(ab, (image.shape[1], image.shape[0]))

    L\_original = cv2.split(lab\_image)[0]

    LAB\_colored = np.concatenate((L\_original[:, :, np.newaxis], ab), axis=2)

    RGB\_colored = cv2.cvtColor(LAB\_colored, cv2.COLOR\_LAB2BGR)

    RGB\_colored = np.clip(RGB\_colored, 0, 1)

    RGB\_colored = (255 \* RGB\_colored).astype("uint8")

    cv2.imwrite(output\_path, RGB\_colored)

def get\_image\_info(image\_path):

    image = cv2.imread(image\_path)

    height, width = image.shape[:2]

    lab\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2LAB)

    mean\_L = np.mean(lab\_image[:, :, 0])

    mean\_A = np.mean(lab\_image[:, :, 1])

    mean\_B = np.mean(lab\_image[:, :, 2])

    return {

        "resolution": f"{width} x {height} px",

        "lab\_mean": f"L: {mean\_L:.2f}, A: {mean\_A:.2f}, B: {mean\_B:.2f}"

    }

def calculate\_ssim(original\_path, colorized\_path):

    original = cv2.imread(original\_path)

    colorized = cv2.imread(colorized\_path)

    original\_gray = cv2.cvtColor(original, cv2.COLOR\_BGR2GRAY)

    colorized\_gray = cv2.cvtColor(colorized, cv2.COLOR\_BGR2GRAY)

    ssim\_index = ssim(original\_gray, colorized\_gray)

    return f"{ssim\_index \* 100:.2f}%"

@app.route("/", methods=["GET", "POST"])

def index():

    if request.method == "POST":

        file = request.files["file"]

        if file:

            filename = file.filename

            file\_path = os.path.join(UPLOAD\_FOLDER, filename)

            bw\_path = os.path.join(BW\_FOLDER, filename)

            output\_path = os.path.join(OUTPUT\_FOLDER, filename)

            file.save(file\_path)

            # Process the image

            colorize\_image(file\_path, bw\_path, output\_path)

            # Get image info

            original\_info = get\_image\_info(file\_path)

            colorized\_info = get\_image\_info(output\_path)

            # Calculate SSIM accuracy

            accuracy = calculate\_ssim(file\_path, output\_path)

            return render\_template(

                "index.html",

                original\_filename=filename,

                bw\_filename=filename,

                colorized\_filename=filename,

                resolution=original\_info["resolution"],

                lab\_info\_original=original\_info["lab\_mean"],

                lab\_info\_colorized=colorized\_info["lab\_mean"],

                accuracy=accuracy

            )

    return render\_template("index.html")

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

1. Source Code of index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Image Colorization</title>

    <style>

        @import url('https://fonts.googleapis.com/css2?family=Bungee+Tint&display=swap');

        body {

            font-family: 'Arial', sans-serif;

            text-align: center;

            margin: 0;

            padding: 0;

            background: linear-gradient(270deg, #0f0c29, #302b63, #24243e);

            background-size: 600% 600%;

            animation: neonBG 15s ease infinite;

            color: #ffffff;

            min-height: 100vh;

            overflow-x: hidden;

        }

        @keyframes neonBG {

            0% { background-position: 0% 50%; }

            50% { background-position: 100% 50%; }

            100% { background-position: 0% 50%; }

        }

        h1 {

            font-family: 'Bungee Tint', cursive;

            font-size: 3rem;

            margin: 40px 0 20px;

            color: #fff;

        }

        .container {

            max-width: 950px;

            margin: auto;

            background: rgba(0,0,0,0.6);

            padding: 20px;

            border-radius: 15px;

            box-shadow: 0 0 20px rgba(0,255,255,0.3);

        }

        .button-group {

            display: flex;

            justify-content: center;

            gap: 20px;

            margin-top: 20px;

            flex-wrap: wrap;

        }

        .button-32 {

            background-color: #0ff;

            border-radius: 12px;

            color: #000;

            cursor: pointer;

            font-weight: bold;

            padding: 12px 24px;

            text-align: center;

            transition: all 0.3s ease;

            border: none;

            font-size: 16px;

            box-shadow: 0 0 10px #0ff, 0 0 20px #0ff inset;

            min-width: 160px;

        }

        .button-32:hover {

            background: #00e5ff;

            box-shadow: 0 0 15px #00e5ff, 0 0 25px #00e5ff inset;

            transform: scale(1.05);

        }

        .custom-file-input {

            color: transparent;

            width: 100%;

            cursor: pointer;

        }

        .custom-file-input::-webkit-file-upload-button {

            visibility: hidden;

        }

        .custom-file-input::before {

            content: 'Select Image File';

            color: #fff;

            background-color: #0ff;

            padding: 10px 20px;

            border-radius: 8px;

            font-weight: bold;

            display: inline-block;

            font-size: 14px;

            box-shadow: 0 0 10px #0ff;

            cursor: pointer;

        }

        .image-container {

            display: flex;

            justify-content: center;

            gap: 30px;

            margin-top: 30px;

            flex-wrap: wrap;

        }

        .image-container img {

            border: 3px solid #0ff;

            border-radius: 10px;

            width: 280px;

            height: auto;

            box-shadow: 0 0 15px #0ff;

        }

        .image-info {

            margin-top: 20px;

            font-size: 1.1rem;

        }

        .image-info p {

            margin: 8px 0;

            color: #0ff;

            font-weight: 500;

        }

    </style>

</head>

<body>

<h1>Upload an Image to Colorize</h1>

<div class="container">

    <form action="/" method="post" enctype="multipart/form-data">

        <div class="button-group">

            <label class="button-32">

                <input class="custom-file-input" type="file" name="file" required>

            </label>

            <button class="button-32" type="submit">Upload & Process</button>

        </div>

    </form>

    {% if original\_filename and bw\_filename and colorized\_filename %}

        <h2>Results:</h2>

        <div class="image-container">

            <div>

                <h3>Original Image</h3>

                <img src="{{ url\_for('static', filename='uploads/' + original\_filename) }}" alt="Original Image">

            </div>

            <div>

                <h3>Black & White Image</h3>

                <img src="{{ url\_for('static', filename='bw/' + bw\_filename) }}" alt="BW Image">

            </div>

            <div>

                <h3>Colorized Image</h3>

                <img src="{{ url\_for('static', filename='outputs/' + colorized\_filename) }}" alt="Colorized Image">

            </div>

        </div>

        <div class="image-info">

            <p><strong>Resolution:</strong> {{ resolution }}</p>

            <p><strong>Original LAB Mean:</strong> {{ lab\_info\_original }}</p>

            <p><strong>Colorized LAB Mean:</strong> {{ lab\_info\_colorized }}</p>

            <p><strong>Accuracy (SSIM):</strong> {{ accuracy }}</p>

        </div>

    {% endif %}

</div>

</body>

</html>

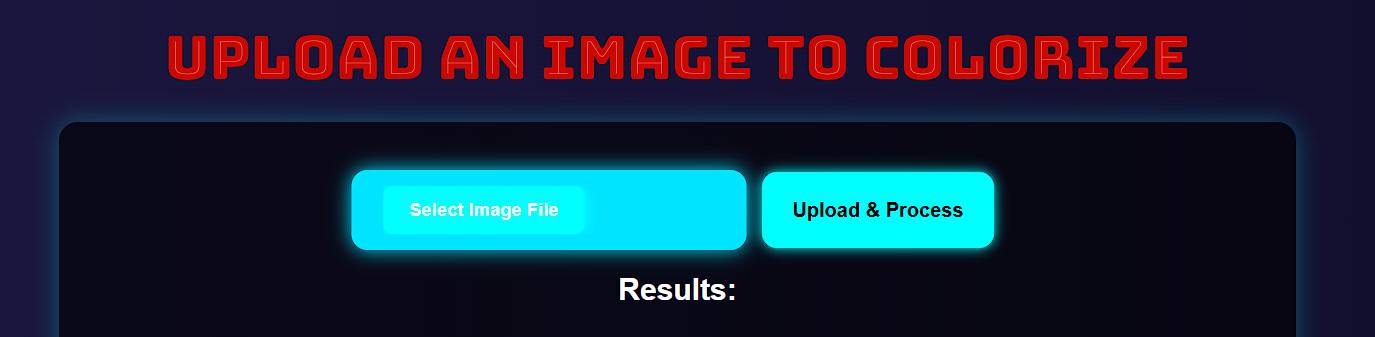
1. Screenshots  
     
   

Figure10.1: Start Window

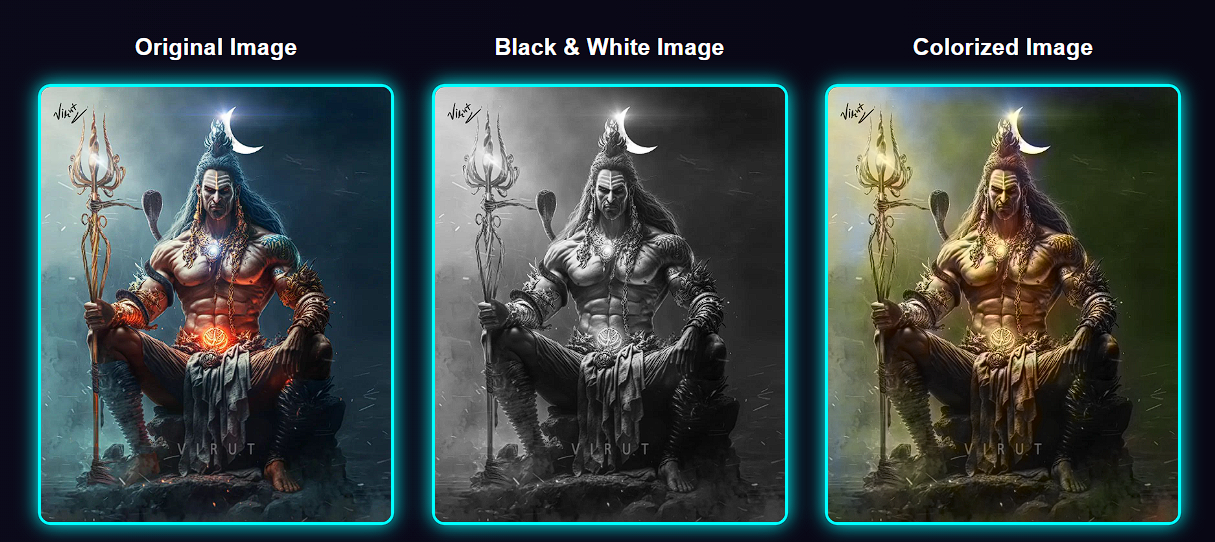


Figure10.2: Result Window

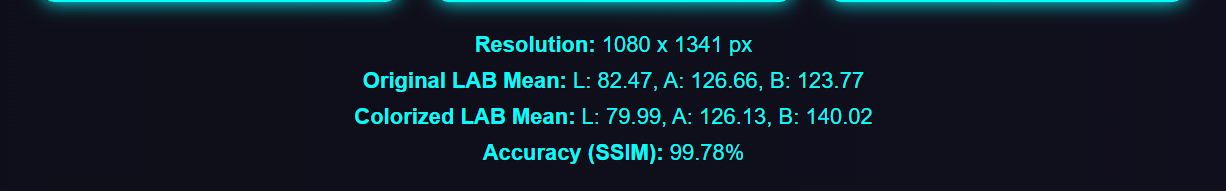


Figure10.3: List of unitary comparison