VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF INFORMATION TECHNOLOGY

FINAL PROJECT REPORT

CS331.P11.KHTN - The Advance Computer Vision

Image Corlorization via Deep-learning

Instructor: Dr. Mai Tien Dung

Students of Group:

Full name	Student ID
Tran Nhat Khoa	22520691
Lý Nguyen Thuy Linh	22520766

Ho Chi Minh City, January 2024

Acknowledgments

We would like to express our deep gratitude to **Dr. Mai Tien Dung** for his dedicated teaching and valuable guidance in the advance Computer Vision course. Your support and feedback have been instrumental in shaping our project. Despite our dedicated efforts to complete this project, we recognize the possibility of overlooked errors. We genuinely appreciate your constructive feedback to help us refine and improve further.

Name	Work Contribution	Percent
Tran Nhat Khoa	Project and baseline idea Slide preparation Report writing	50%
Ly Nguyen Thuy Linh	Baseline refinement Slide preparation Report writing	50%

1 Introduction

1.1 Motivation

Image colorization is a fascinating computer vision task within the domain of **Image-to-Image Translation**. Its applications extend beyond just technological curiosity, offering significant value in various areas of our lives:

1. **Historical Restoration (Figure 1a):** Imagine holding a faded photograph of your grandparents or a cherished old memory. By colorizing these black-and-white images, we breathe life back into history, creating vivid visuals that preserve and honor precious moments. These restored images become treasures that connect us to the past and make memories feel alive again.
2. **Creative Inspiration for Children (Figure 1b):** Colorization isn't limited to nostalgia—it's also a creative tool! For example, you can transform black-and-white sketch images into colorful artwork, inspiring children to engage their imagination and experiment with colors as they learn and draw.
3. **Data Augmentation for Machine Learning (Figure 1c):** From a technical perspective, colorization offers an innovative way to **augment training datasets**. By generating multiple colorful variations of an image, we can diversify datasets, improving model performance and robustness in downstream tasks. This is especially valuable in applications like object detection, segmentation, and more.

a) Before and after applying image colorization on historical images.

b) Before and after applying image colorization on children drawing.

c) Data argument.

Figure 1: Motivation of Image colorization

1.2 Problem Statement

1.2.1 Inference Level (Figure 2)

- **Input:** a gray-scaled image (sketch image without colors)
- **Output:** a colorized version of an input image.

1.2.2 Training Level

- **Input:** dataset including gray-scaled images(be transformed from colorized image) and original colorized images.
- **Output:** Trained Machine learning model.

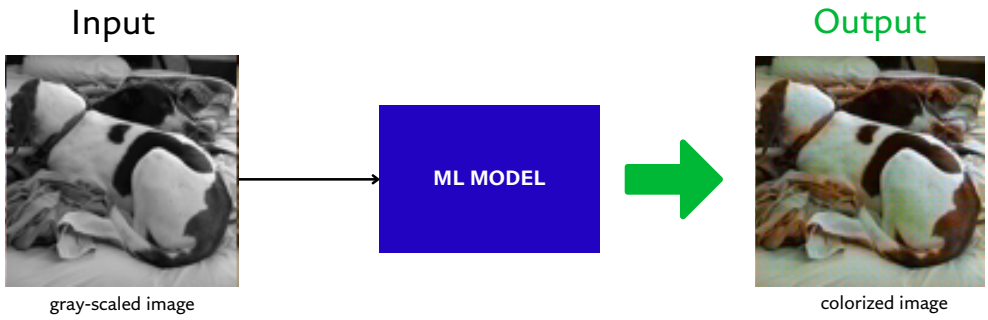


Figure 2: Input and output of Image Colorization(inference level)

2 Method

2.1 Reconstruction Approach

2.1.1 Reconstruction with RGB color channel

The first basic approach for tackling the image colorization problem is to consider it as a reconstruction problem. The goal is to train a machine learning model f to reconstruct a grayscale image x into a colorized image $\hat{y}$ where y represents the original color image used as the ground truth label. Figure 3 illustrate the pipeline of this approach.

Mathematically, this can be formulated as $f : R^{H \times W \times 1} \rightarrow R^{H \times W \times 3}$.

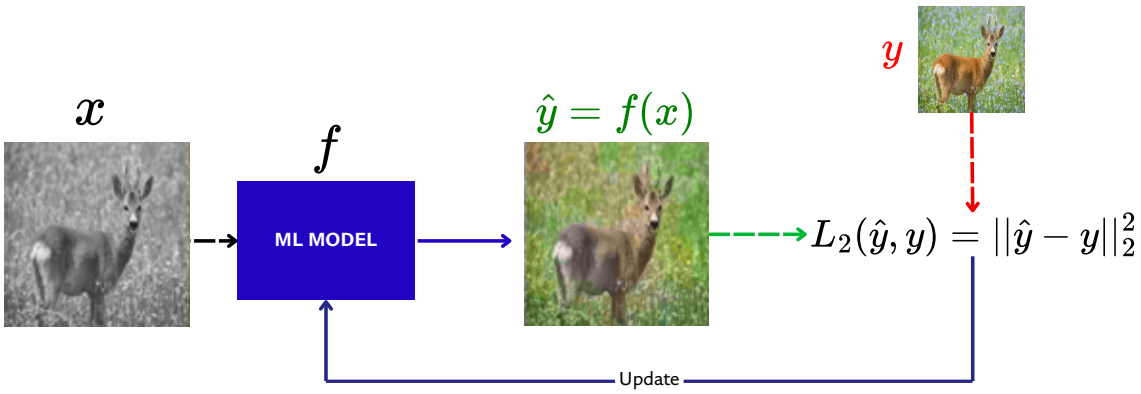


Figure 3: The reconstruction approach pipeline with input gray scale image x is passed to machine learning model f to create $\hat{y}$ and compare it with ground truth colorized image y by using L_2 loss function.

2.1.2 Limitation of using RGB color channel

However, this approach has some limitations: The output image must **reconstruct the entire image**, including edges, structure, and other intricate details, which can be challenging. This method requires **regression from one value to three values**, as the input consists of a single grayscale channel, but the model must predict two additional channels to generate the colorized output.

Figure 4 illustrates the results after training for 200 epochs using this baseline approach. It is evident that the model struggles to reconstruct the entire image accurately.

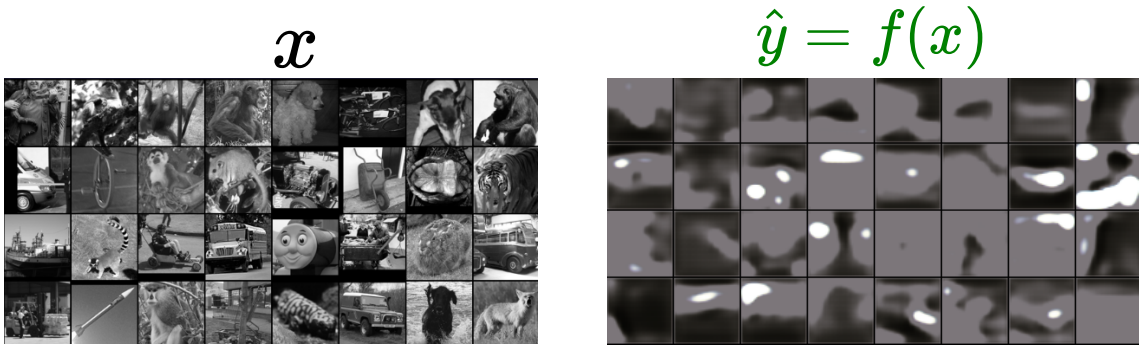


Figure 4: Results after 200 epochs using the reconstruction baseline. The model faces difficulties in accurately reconstructing the entire image.

2.1.3 Reconstruction with CIE Lab color channel

To address the limitations of using RGB channels, we adopt the CIE-Lab color space. This color space is particularly advantageous because it separates the lightness channel ($L \in [0, 100]$), which directly corresponds to the grayscale input, from the chromatic channels ($a, b \in [-110, 118]$). Therefore, we clip the values to this range for consistency.

This approach enables us to leverage the grayscale image as the luminance component, while focusing the model's learning on predicting the chromatic information in the a and b channels. The problem can thus be formulated as $f : R^{H \times W \times 1} \rightarrow R^{H \times W \times 2}$. Figure 10 illustrates the pipeline of this approach. First, the ground truth color image y is converted to the CIE-Lab space by using a black-box sklearn function, and the a and b channels are used as ground truth. Additionally, the grayscale input x is normalized to $[0, 100]$ to match the luminance channel constraint.

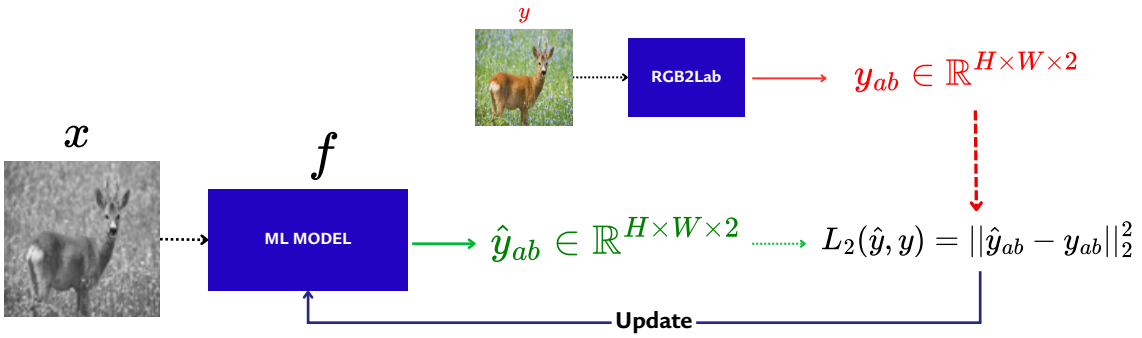


Figure 5: The reconstruction approach pipeline using the CIE-Lab color space. The input grayscale image x is passed through the machine learning model f to produce $\hat{y}_ab$, which is compared with the ground truth y_ab using the L_2 loss function.

During the inference phase, the grayscale image x is input into the trained model to generate $\hat{y}_{ab}$. Output $\hat{y}_{ab}$ is then combined with x to reconstruct the colorized RGB

image. This conversion is achieved using a black-box function from the sklearn library. Figure 6 illustrates the inference process for this approach.

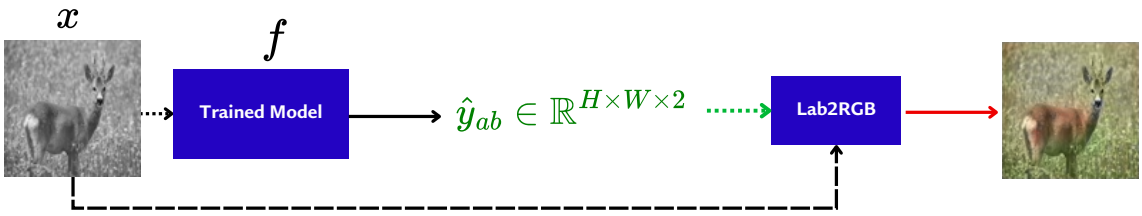


Figure 6: The inference pipeline using the CIE-Lab color space. The grayscale input x is processed by the trained model to produce $\hat{y}_{ab}$, which is then combined with x and converted back to the RGB color space using a black-box function.

2.2 Classification Approach

With the reconstruction approach, we can only return a single option (a, b). However, our goal is to determine the **likelihood of a specific pixel being colorized with a particular color**. To achieve this, we propose a classification model that takes a single-channel grayscale image as input and outputs the probability distribution over all possible colors. This approach requires **converting the continuous color space into a categorical space**. A straightforward method is to divide the RGB color space into Q classes, where each class represents a fixed-size color bin.

However, the limitation of the RGB channel arises again: the number of color bins (equivalently, classes) is $Q = \frac{256^3}{\text{binsize}^3}$, which is computationally prohibitive for practical purposes. To address this, we once again leverage the CIE-Lab color space to implement the classification approach, reducing the complexity of the problem while maintaining compatibility with the grayscale input.

With the valid range of values for CIE Lab mentioned earlier, we set the binsize to 12, resulting in $Q = 361$. Figure 7 illustrates the a, b space with $L = 25$ and binsize $= 12$. Each color is represented by the centroid value (a, b), which is converted to the RGB color format.

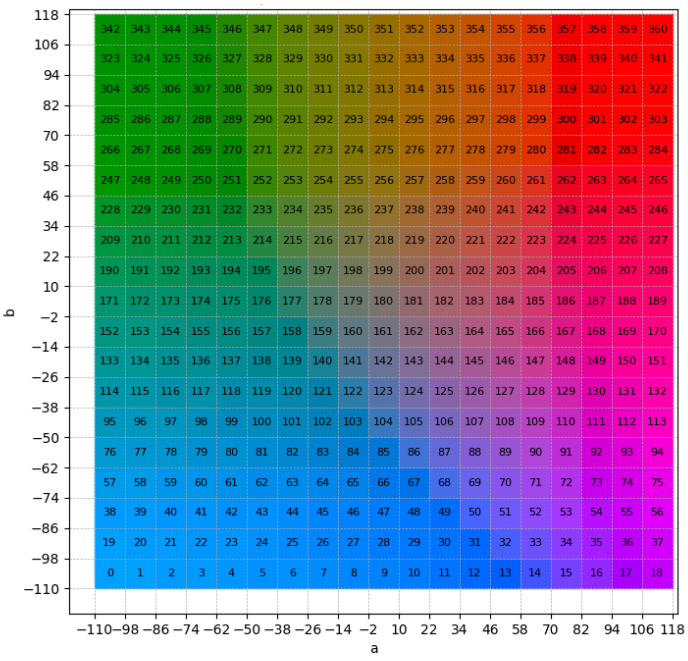


Figure 7: Visualization of the a, b color space at $L = 25$ with a binsize of 12. Each point represents a color in the RGB format derived from the centroid values.

The training pipeline for this approach is mostly the same as the reconstruction approach, but the output $\hat{y}_{\text{logit}}$ is the logits probability of pixels that likely belong to the Q bins. For the ground truth, we perform the same process by taking y_{ab} from the ground truth image and encoding it by finding the bin that has the nearest distance to it.

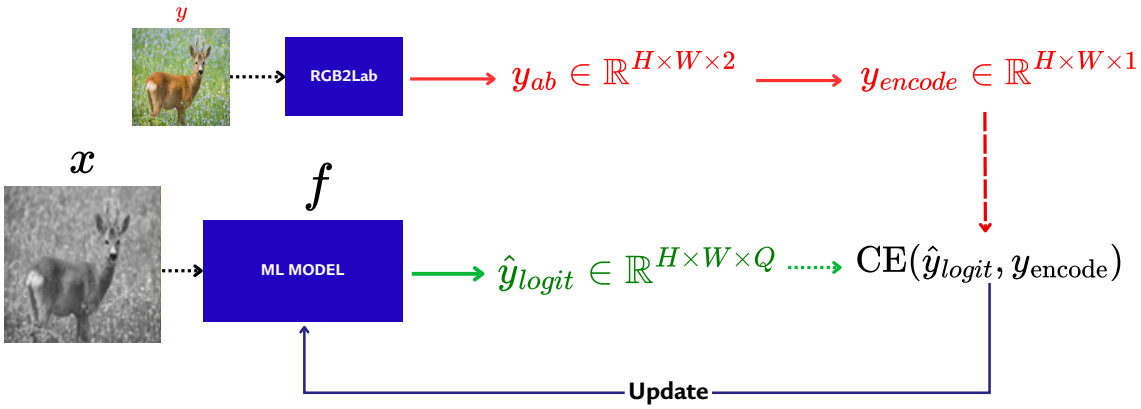


Figure 8: The classification approach pipeline using the CIE-Lab color space. The input grayscale image x is passed through the machine learning model f to produce $\hat{y}_{\text{logit}}$, which is compared with the ground truth y_{encode} using the cross-entropy loss function.

During inference, after generating the output $\hat{y}_{\text{logits}}$, we can select the bin with the highest logits value and assign the centroid color of this bin to the corresponding pixel. Another approach is to choose the top k highest logits values and merge them using the softmax function to enhance the color variety of that pixel.

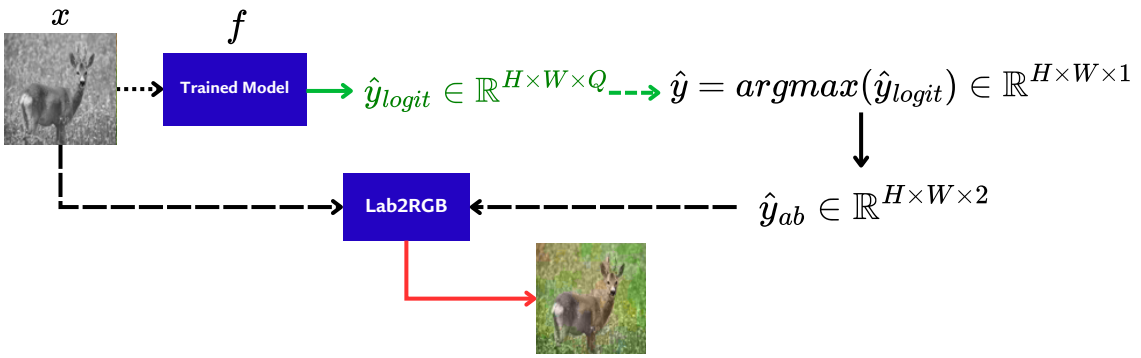


Figure 9: The inference pipeline using the CIE-Lab color space with centroid assigned. The input grayscale image x is passed through the machine learning model f to produce $\hat{y}_{\text{logit}}$, and using $argmax$ to make the final prediction.

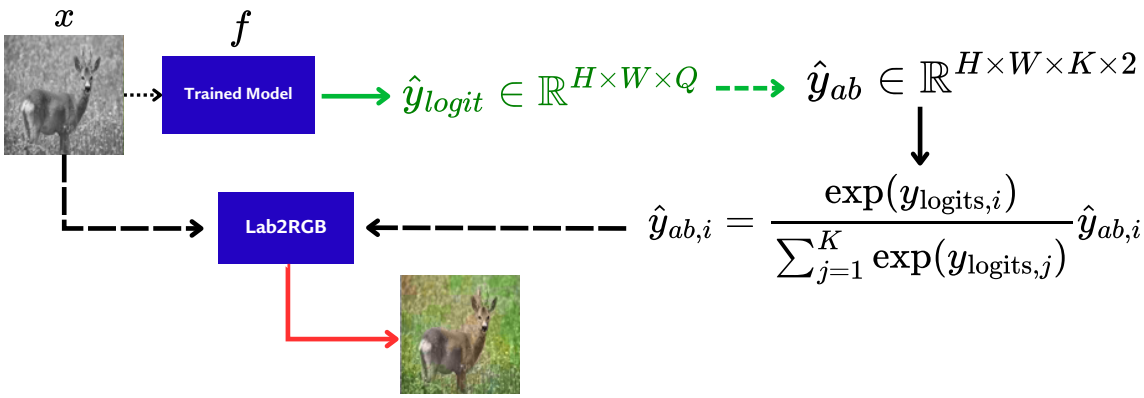


Figure 10: The inference pipeline using the CIE-Lab color space with centroid assigned. The input grayscale image x is passed through the machine learning model f to produce $\hat{y}_{\text{logit}}$ and then take the k highest item from that, then merge them using **cross entropy** function to make the final prediction.

2.3 GAN approach

The main objective of colorization is to apply colors reasonably and potentially creatively. Colorization can be approached using Generative Adversarial Networks (GANs), where the generator creates colorized images. To deceive the discriminator, the generator aims to color the images as naturally as possible. To encourage creativity and prevent mode collapse and vanishing gradients, we choose the Wasserstein GAN with Gradient Penalty (WGAN-GP) as our GAN approach. (Figure **??**) The pipeline is similar to the reconstruction approach but with the addition of a discriminator that distinguishes between real and fake images.

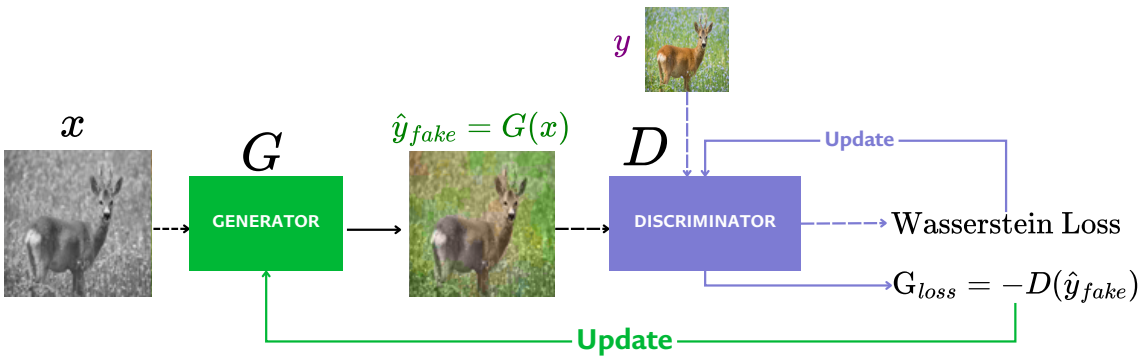


Figure 11: The training pipeline of the GAN approach involves passing the grayscale input x to the generator (similar to the reconstruction approach). The output from the generator is then passed to the discriminator D. Both the generator and discriminator are optimized in parallel, utilizing the Wasserstein loss for the reconstruction task.

2.4 Deep learning Architecture

We utilize a simple CNN Autoencoder architecture, modifying the final layer based on the specific approach. Figure 12 illustrates the architecture of the CNN autoencoder in detail. For the reconstruction and generation (GAN) tasks, the final layer outputs $W \times H \times 2$ using the Tanh activation function. In contrast, for the classification task, the final layer produces $W \times H \times Q$ with a ReLU activation function. Furthermore, with the GAN approach, we employ the CNN architecture depicted in the figure 13.

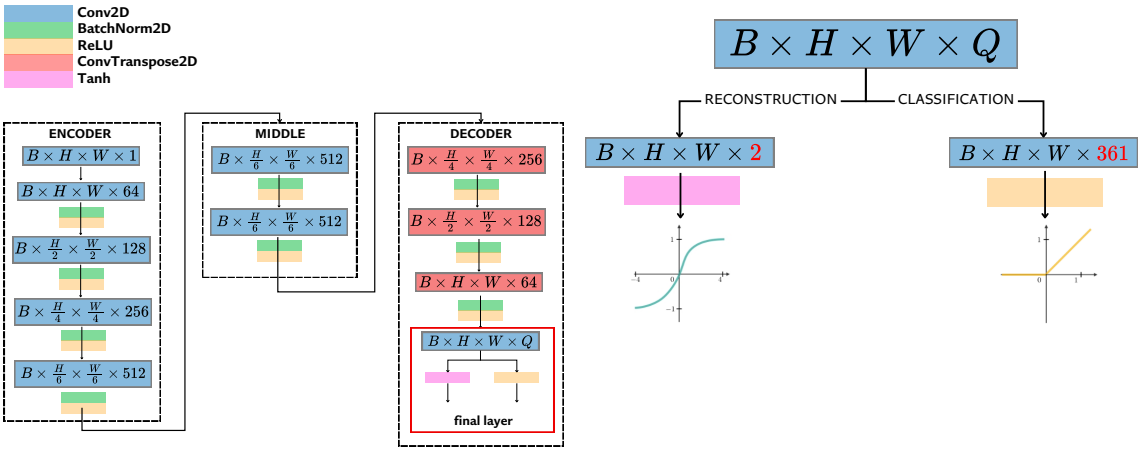


Figure 12: The CNN Autoencoder architecture serves dual purposes: it facilitates both the reconstruction of input data and the classification of labels.

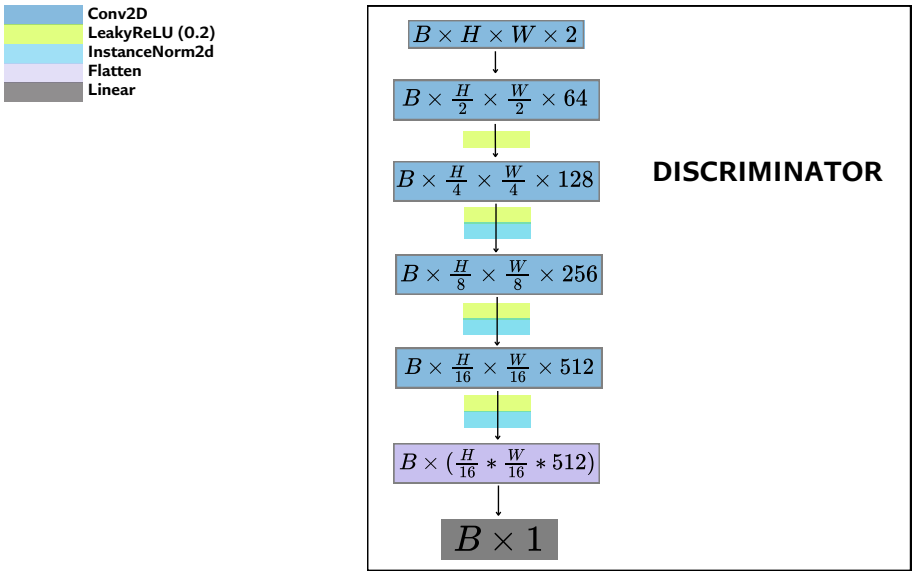


Figure 13: The CNN discriminator architecture.

3 Experiment

3.1 Setting

STL10 Dataset: This dataset contains object images for classification tasks, consisting of 96x96 images divided into 10 classes. The training set, which is unlabeled, contains 100,000 images, while the testing set includes 8,000 images. For each experiment, we conduct training for 200 epochs, except for the GAN approach, which is trained for 1,000 epochs. We apply a weight decay of 1×10^{-3} and utilize a GeForce RTX 4090 Ti for hardware acceleration.

Metric Evaluation (Figure 14): We use the Peak Signal-to-Noise Ratio (PSNR) score to evaluate the reconstruction ability and the Inception Score (IS) to assess the creativity of the colorized images compared to the original images.

Mean Square Error (MSE)

$$MSE = \frac{1}{MxN} \sum_{x=1}^{M} \sum_{y=1}^{N} [f(x,y) - g(x,y)]^2$$

where:
MxN: total number of pixels in the image
$f(x,y)$: input image
$g(x,y)$: enhanced (output) image

Peak Signal-to-Noise Ratio (PSNR)

$$PSNR = 10 \log \frac{f_{max}^2}{MSE}$$

where:
$f_{max} = 255$ is the maximum gray value

(a) PSNR score.

Inception Score (IS)

$$\text{IS}(G) = \exp\left(\mathbb{E}_{x \sim p_{\text{data}}} \left[D_{\text{KL}}\left(p(y|x) \parallel p(y)\right)\right]\right)$$

- $\text{IS}(G)$: Inception Score of the generative model G.
- p_{data}: The real data distribution.
- $p(y|x)$: The predicted class distribution from the Inception model on sample x.
- $p(y)$: The marginal distribution of the predicted class labels across all samples.
- D_{KL}: The Kullback-Leibler Divergence, which measures the difference between the predicted distribution $p(y|x)$ and the average class distribution $p(y)$.
- exp: The exponential function, representing the average of the logarithm.

(b) Inception score.

Figure 14: The metric to evaluate the image colorization task.

3.2 Results

Our experimental evaluation compared four different approaches to image colorization: reconstruction-based, classification with centroid assignment, classification with softmax assignment, and GAN-based methods. The results are summarized in Table 1 and visualized in Figure 15.

Overall, our approach **successfully colorized images** with results that were both technically accurate and visually appealing to human observers.

More details, our reconstruction-based approach achieved a peak PSNR of 30.44, demonstrating superior pixel-level accuracy when compared to the ground truth. Meanwhile, the classification-based approaches, particularly those using softmax assignment, yielded higher Inception Scores (18.7483), indicating better perceptual quality and greater diversity in the generated colors.

Figure 16 illustrates the relative strengths and limitations of each approach.

Approach	PSNR	IS
Reconstruction	**30.44**	12.1235
Classification - centroid assign	25.2813	18.6873
Classification - softmax assign	25.4372	**18.7483**
GAN	23.2057	17.1104

Table 1: Experimental Results Across Different Approaches

3.3 Limitations

Our study revealed several important limitations in the current approaches:

1. **Dataset Constraints:** The STL10 dataset used for training shows limited generalization capability, particularly for real-world applications.
2. **Background Handling:** All approaches struggle with noisy data, especially in handling complex backgrounds such as grass or foliage.
3. **Color Bias:** There is a consistent bias toward greenish tones in the generated images, suggesting potential dataset imbalance or model limitations.

These findings highlight the need for further research in developing more robust and versatile colorization methods that can better handle real-world scenarios while maintaining both accuracy and creative color generation.

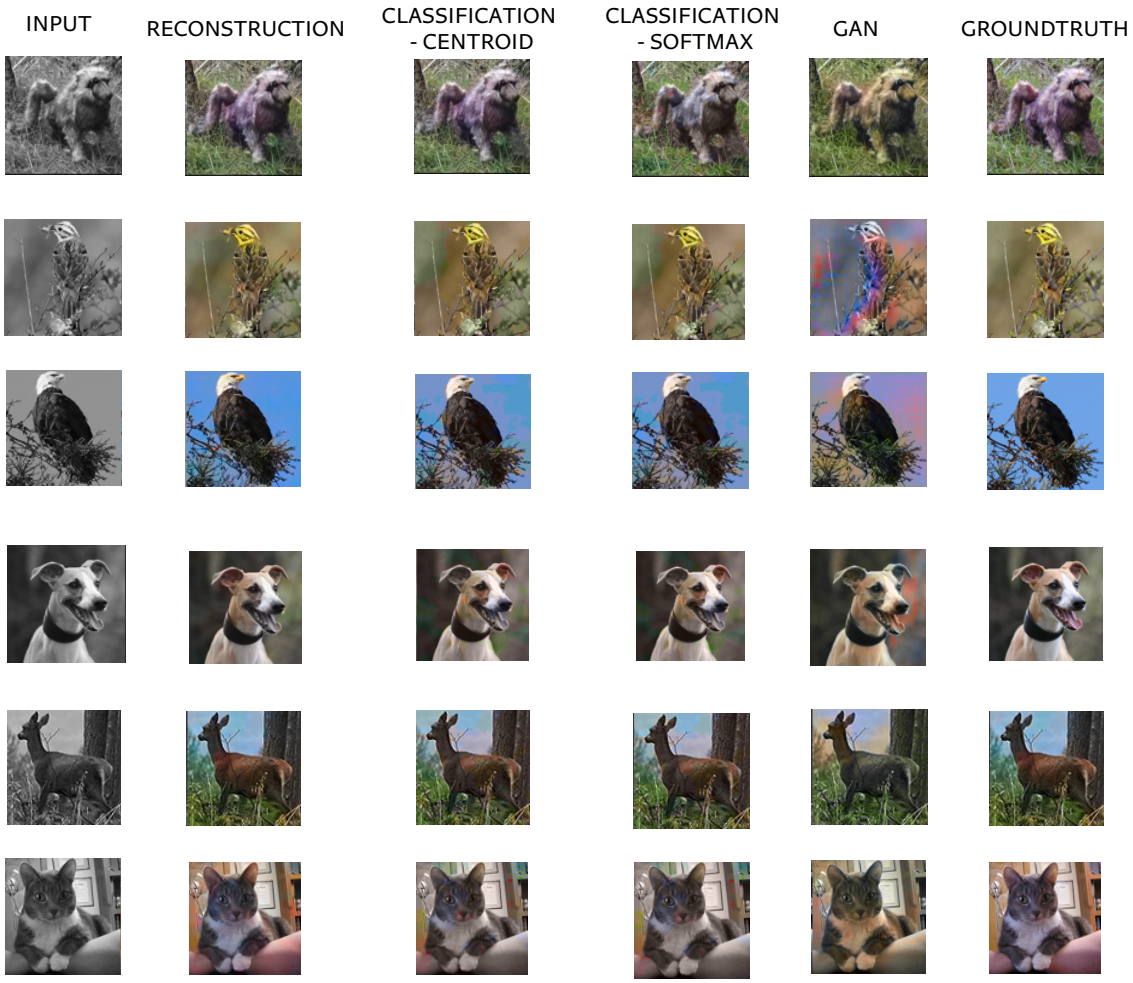


Figure 15: Comparison of colorization results across different approaches.

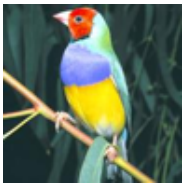
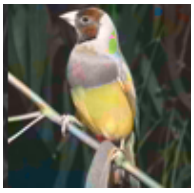
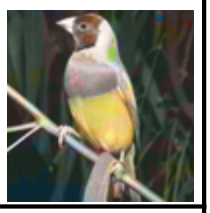
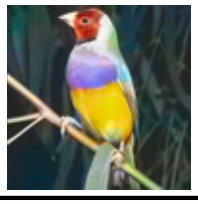
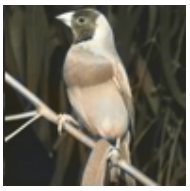
INPUT		**GROUND TRUTH** 
RECONSTRUCTION		• Pros: ◦ stable and easy training convergence ◦ easy implementation • Cons: lack of creativity
CLASSIFICATION - centroid		• Pros: ◦ more creative than reconstruction • Cons: ◦ larger architecture ◦ slower to convergence
CLASSIFICATION - softmax		
GAN		• Pros: ◦ most creative of all • Cons: ◦ hard to convergence due to mode collapse or other insufficient training stability

Figure 16: Visualization of limitations in current approaches.

References

[1] Adam Coates, Honglak Lee, and Andrew Y. Ng. An analysis of single layer networks in unsupervised feature learning. In *AISTATS*, 2011.

[2] M.R. Luo. Cielab. In R. Luo, editor, *Encyclopedia of Color Science and Technology*. Springer, Berlin, Heidelberg, 2015.

[3] Github Repository. Datumizer-wikipedia-illustrations. `https://github.com/mjhorvath/Datumizer-Wikipedia-Illustrations`, 2023.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[5] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *ECCV (3)*, pages 649–666, 2016.