Black and White Image Colorization with Deep Learning

Yunus Emre Bayraktar
2210765023

Ali Yiğit Şenyurt
2200765023

***Abstract*—Image colorization is a task that transforms grayscale images into colorized versions, enhancing their visual appeal and usefulness. This paper presents a comparative study of three U-Net-based architectures for automatic image colorization, evaluated using the MIRFLICKR-25K dataset. The models include a baseline architecture with minimal depth and no skip connections, an intermediate model with increased depth and skip connections, and an advanced model that integrates both skip connections and dilation layers. The first model showed limited colorization ability, failing to add significant color to images. The second model improved performance by adding basic colors, particularly in simpler areas like the sky, indicating some learning of semantic information. The third model demonstrated superior performance, producing vibrant and realistic colorizations, although it tended to favor red hues. These results highlight the crucial role of model depth and architectural enhancements in achieving higher quality image colorization, offering insights into the effectiveness of various U-Net configurations for this application.**

I Introduction

Ever since the start of black and white photography, people have tried to add color to old monochrome photographs, videos, and sketches. Traditionally, this task is done by the artists who had to color each pixel by hand. In recent years, however, deep learning and computer vision models have accelerated this process and even automated it, making image colorization more accessible and efficient.

The goal of image colorization is to add colors to a grayscale image, making the resulting image both perceptually meaningful and visually appealing. One of the primary challenges of this task is that it is under-constrained, meaning that a single gray pixel can potentially be colored in many different ways (for example, the sky could be blue, pink, or orange). Consequently, there is no single correct solution for colorizing an image, and human intervention often plays a crucial role in the process.

Colorization methods can generally be divided into two main categories: interactive and automatic. Interactive colorization techniques involve users manually adding color scribbles to the target image. These colors are then propagated smoothly across the image based on an optimization framework. However, this method demands significant effort from the user and the quality of the colorization heavily depends on the quality of the user-supplied scribbles. This can be a challenge, especially for users who may struggle to provide effective scribbles.

Automatic image colorization involves adding colors to a grayscale image without any user intervention. These colorization methods take a different approach by using a reference color image from which colors are transferred to the target image. This problem is challenging because it is impossible to determine the correct colors for a grayscale image without prior knowledge. Many objects can have a variety of colors; for instance, artificial objects made of plastic can come in any color, and natural objects like tree leaves can range in shades of green and turn brown in autumn without changing their shape. These methods often require careful adjustment of numerous parameters to achieve satisfactory results.

II Related Work

Recent advancements in image colorization have explored various methods to transform grayscale images into colorful images with minimal user intervention. An example-based method uses a semantically similar reference color image, extracting features at the superpixel level to colorize the target image. High confidence superpixel matches guide initial color assignments, corrected by an image domain voting framework for further consistency. This method outperformed existing techniques with a fixed parameter set [1].

Another approach developed an automatic colorization method, allowing user corrections if needed. It addresses local texture complexity by estimating possible color distributions for each pixel and using graph cut algorithms to maximize the colorized image's probability. Working in the L-a-b color space, it leverages machine learning to handle uncertainty, achieving speed and resistance to texture noise [3].

A third project tackled automatic colorization as a classification task, increasing color diversity using a Convolutional Neural Network (CNN) trained on over a million images. This method successfully fooled humans in 32% of "colorization Turing test" trials and served as a powerful pretext task for self-supervised feature learning [4].

These projects share goals of optimizing colorization through deep learning and neural networks, ensuring color consistency, and minimizing user intervention. They rely on extensive training datasets to enhance accuracy and performance. They show a significant progress in automated image colorization.

III Method

LAB Color Space

Although the Red-Green-Blue (RGB) color space is widely known for color representation, the Lightness-A-B (LAB) color space is often preferred in colorization research due to its perceptual benefits. The LAB space separates lightness from color, with the A channel representing the green-red axis and the B channel representing the blue-yellow axis. This distinction allows the lightness (L channel) of a grayscale image to be used as input to a model that predicts the color components (A and B channels). These predicted channels are then merged with the original lightness to produce a colorized image. This method leverages LAB's ability to handle color and luminance independently. This results in more accurate and visually appealing colorization by retrieving the image's structural and lightness details while adding realistic color.

Models

Our models are mostly inspired from the U-Net architecture. The U-Net architecture is a widely used deep learning model designed for image segmentation tasks. It is characterized by its distinctive U-shaped structure. It consists of two main parts: an encoder and a decoder. The encoder part follows a typical convolutional network design where it involves repeated application of convolutional layers with ReLU activation and max-pooling operations. This progressively down samples the image to capture context and high-level features. At the bottom of the U, the bottleneck layer connects the encoder to the decoder, capturing the most abstract features. The decoder then mirrors the encoder but uses transposed convolutions to up sample the feature maps. Skip connections link corresponding layers of the encoder and decoder, allowing the network to retain fine-grained details by combining low-level and high-level features. The final layer of U-Net employs a 1x1 convolution to produce the segmentation map, where each pixel is classified. The exact structure of the U-Net can be seen in Figure 1.

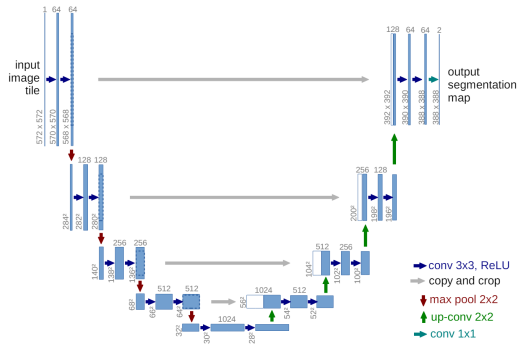


Fig. 1: Original U-Net Architecture

We developed three progressively complex models based on this U-Net architecture to explore their effectiveness in image colorization, each incrementally adding complexity and sophisticated techniques. The first model used as a baseline with a straightforward architecture, comprising two convolutional layers and two transpose convolutional layers, and taking the Lightness (L) channel as input. It intentionally omits skip connections to see their impact on performance. The final layer of this model is a 2x2 convolution that produces the A and B color channels. These channels are then combined with the input L channel to generate the final colorized image. This basic model establishes a reference point, helping us understand the foundational capabilities of a U-Net without additional complexities.

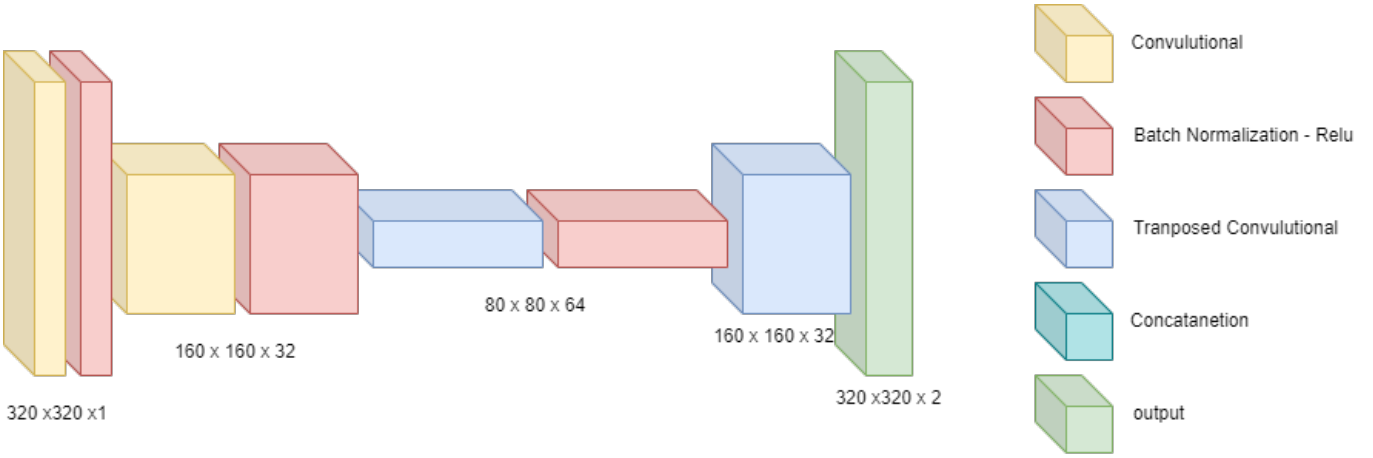
The second model enhances the baseline by introducing increased depth and integrating skip connections, thus aligning more closely with the traditional U-Net structure. It includes three convolutional layers and three transpose convolutional layers, maintaining the input as the Lightness (L) channel. The addition of skip connections between corresponding encoder and decoder layers allows the network to retain critical spatial features from earlier layers. This effectively uses detailed information and improves the model's ability to reconstruct colorized images. This architectural enhancement aims to demonstrate the benefits of feature reuse and increased depth in capturing and preserving image details.

The third model represents the most advanced iteration, significantly deepening the network with six convolutional layers and six transpose convolutional layers. It not only employs skip connections but also integrates dilation layers (or 'a trous' layers) into the convolutional stages. These dilation layers expand the receptive field, enabling the network to capture broader contextual information and finer image details without a substantial increase in parameters. Research by Liang et al. [5] has shown that dilation layers can offer considerable improvements in tasks similar to ours by allowing the model to incorporate a wider range of contextual features and spatial information. As with the previous models, the output A and B channels are combined with the input L channel to form the final colorized image. This model aims to leverage the deeper architecture and advanced convolutional techniques to enhance feature extraction and improve the overall quality of image colorization.

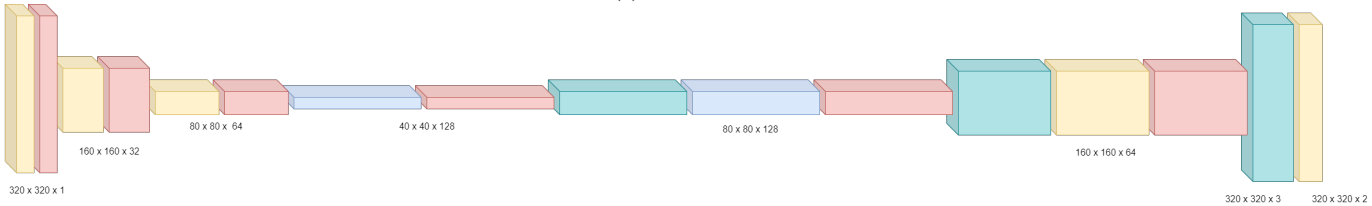
Each model is visually presented in the Figure 2, providing a clear representation of their respective architectures and how they handle information flow during the image colorization process. By comparing the performance of these three models, we aim to gain insights into the effects of depth, skip connections, and advanced convolutional methods on the colorization task, thereby refining our understanding of optimal U-Net configurations for this application.

IV Dataset

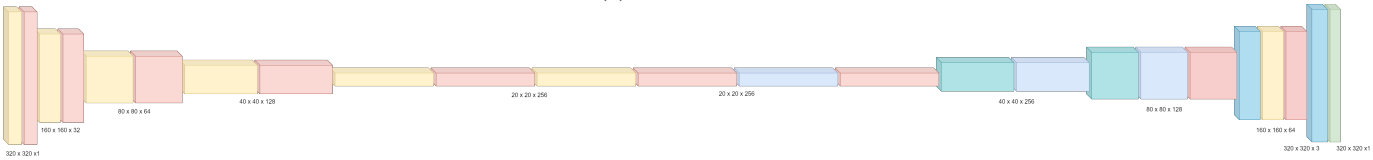
The MIRFLICKR-25K dataset [XX] is a well-known benchmark in the field of image processing and computer vision. It contains 25,000 images collected from photo-sharing website Flickr. These images cover a wide variety of subjects, including natural landscapes, urban scenes, people, and abstract compositions. They were selected based on their "interestingness" scores on Flickr. This ensures a diverse and



(a) First Model



(b) Second Model



(c) Third Model

Fig. 2: Model Architectures

engaging collection. Each image in the dataset comes with rich metadata, including user-provided tags, titles, and descriptions. The MIRFLICKR-25K dataset is widely used for research purposes due to its real-world relevance and wide range of scenarios, making it ideal for testing and developing robust image processing algorithms.

Pre-processing

To prepare the MIRFLICKR-25K dataset for our image colorization task, we performed several preprocessing steps. First, we removed any grayscale images already present in the dataset, as we require pairs of grayscale images and their corresponding colorized versions for training our model. After filtering out the grayscale images, we randomly split the remaining dataset into training and testing sets, allocating 8,000 images to the training set and 1,000 images to the testing set. Each image was then split into the Lightness (L) and color (a and b) channels. For consistency and to manage memory and computational constraints, all images were resized to 320 × 320 pixels. These preprocessed grayscale images and their color counterparts are shown in the Figure 3.

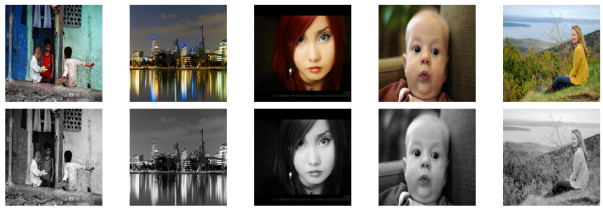
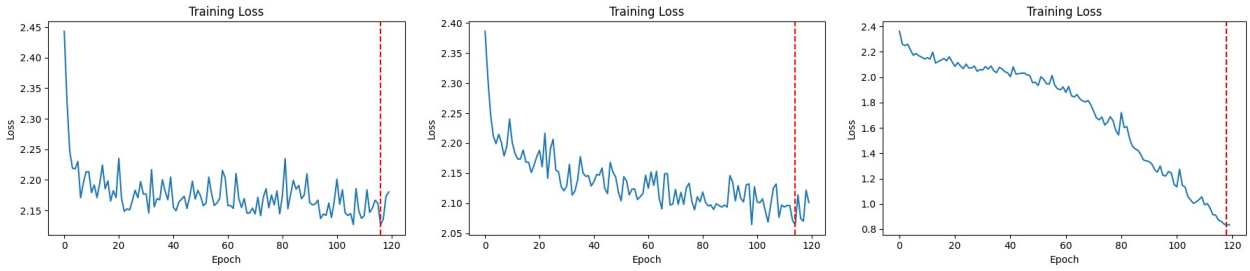


Fig. 3: Obtained grayscale images and original versions

V Experiments

In our experiments, we evaluated different U-Net-based models for image colorization using the Adam optimizer with a learning rate of 0.001 and L2 loss as the metric for both training and evaluation. We trained each model with a batch size of 128 for 120 epochs on Google Colab, with an NVIDIA L4 Tensor Core GPU. We did not use early stopping or a learning rate scheduler, instead we maintained a constant learning rate throughout the training process. L2 loss is as follows:

$$L2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$



(a) First Model (b) Second Model (c) Third Model

Fig. 4: Training losses of the models.

The best model was selected based on the lowest training L2 loss, and we saved this model and its weights for further analysis. Hyperparameter tuning, including adjustments to the learning rate and network depth, was performed manually to determine the optimal settings for our final experiments. To ensure reproducibility, a random seed of 7 was used for all experiments, providing consistency across different runs. The learning curves, which show the progression of training loss over epochs, are illustrated in Figure 4. This methodical approach allowed us to observe the effects of model depth and evaluate the inherent capabilities of each architecture.

VI Results

As we stated in the introduction section, it is hard to evaluate the performance of this task with quantitative metrics. Because this task is under-constrained, meaning that a single gray pixel can potentially be colored in many different ways (for example, the sky could be blue, pink, or orange). Consequently, there is no single correct solution for colorizing an image, and human intervention often plays a crucial role in the process. Because of this reason we evaluated the results qualitatively. The results of the models 1,2 and 3 can be seen from the Figures 5 6 and 7 respectively.

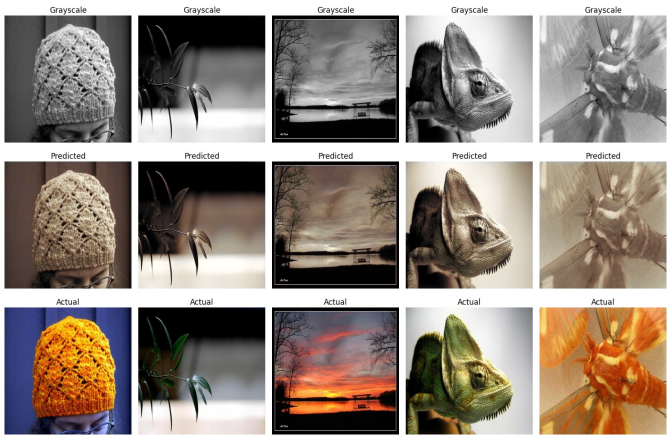


Fig. 5: Colorization results of the first model

The first model performed poorly in colorizing the images. In many cases, it failed to add any meaningful color, resulting in outputs that were almost entirely grayscale. This indicates a significant limitation in the model's learning capability, as it struggled to learn even from the training data. This failure to generalize or even memorize training instances is a critical red flag, suggesting that the shallow architecture without skip connections lacks the capacity to capture and apply color information effectively.

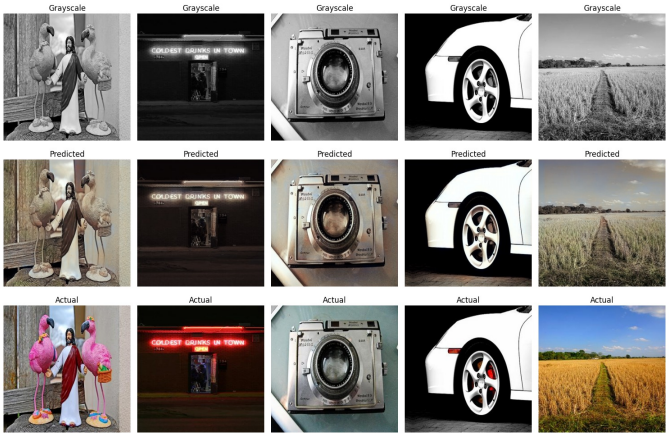


Fig. 6: Colorization results of the second model

The second model showed noticeable improvements over the baseline. It succeeded in adding some color to the images, particularly in simpler regions such as the sky. This suggests that the model has started to learn semantic information and can differentiate between background elements and other parts of the image. However, its colorization was often limited to easier, more uniform areas, indicating that while the addition of skip connections improved learning, the model still struggled with more complex details and varied textures within the images.

The third model outperformed the previous ones, producing the most vibrant and accurate colorizations. This model demonstrated a clear ability to separate objects within the images and color them appropriately. It effectively utilized the additional depth and dilation layers to capture more contextual information and apply it to the colorization task, resulting in richer and more realistic outputs. However, we observed a tendency towards red hues in certain images, indicating a potential bias in the model's color predictions. This bias could

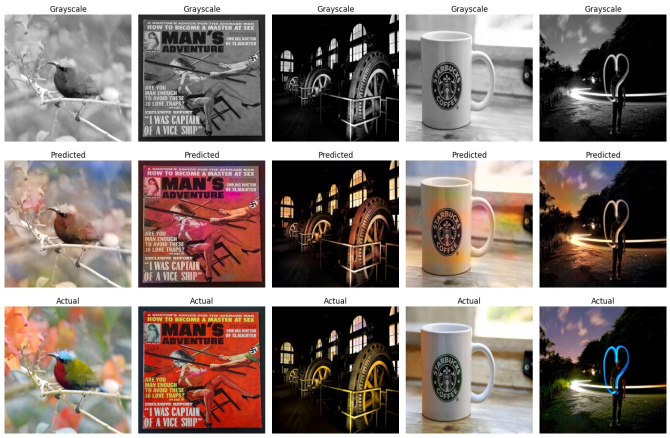


Fig. 7: Colorization results of the third model

be attributed to the model's over-reliance on certain features or training data characteristics that favored red tones.

Overall, the progression from Model 1 to Model 3 highlights the importance of model depth and architectural enhancements such as skip connections and dilation layers in improving the quality of image colorization. Model 1's failure to colorize effectively underscores the limitations of a shallow network without sophisticated information flow. Model 2's success in simple regions demonstrates the benefits of skip connections in capturing basic semantic information. Model 3's superior performance showcases the advantages of a deeper network and advanced convolutional techniques in generating more detailed and realistic colorizations, though it also reveals areas for further refinement, such as addressing color biases. A general comparison of these three models can be seen in Figure 8.

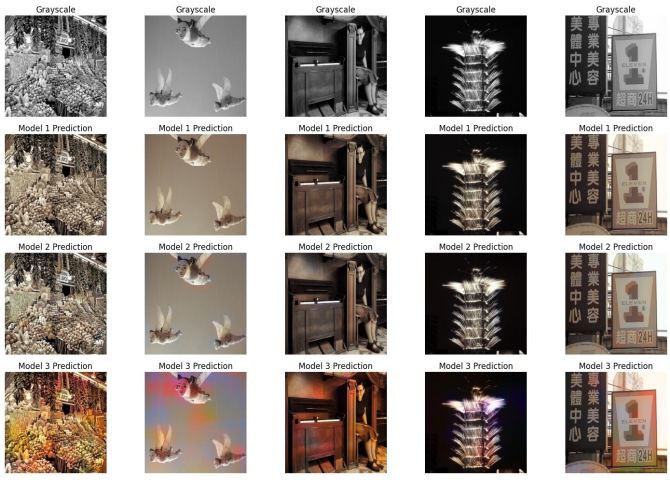


Fig. 8: Comparison of the three models

VII Conclusion

This paper presented a study of three U-Net-based models for automatic image colorization using the MIRFLICKR-25K dataset. Our findings highlight the crucial role of model complexity and architectural enhancements in achieving effective colorization. Model 1 lacked depth and failed to add meaningful color, demonstrating the limitations of shallow architectures. Model 2 improved with the addition of skip connections, successfully colorizing simpler regions but struggling with complex details. Model 3 excelled, producing the most vibrant and realistic results thanks to its deeper architecture and dilation layers. But still it showed a slight bias toward red hues.

Overall, the study shows that deeper networks with advanced features significantly enhance colorization quality. Future work should focus on mitigating color biases and exploring further architectural improvements or training strategies. These findings provide valuable insights into optimizing U-Net configurations for better image colorization.

References

[1] Gupta, Raj Kumar, et al. "Image colorization using similar images." Proceedings of the 20th ACM international conference on Multimedia. 2012.

[2] Cheng, Zezhou, Qingxiong Yang, and Bin Sheng. "Deep colorization." Proceedings of the IEEE international conference on computer vision. 2015.

[3] Charpiat, Guillaume, Matthias Hofmann, and Bernhard Schölkopf. "Automatic image colorization via multimodal predictions." Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part III 10. Springer Berlin Heidelberg, 2008.

[4] Zhang, Richard, Phillip Isola, and Alexei A. Efros. "Colorful image colorization." Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. Springer International Publishing, 2016.

[5] Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." IEEE transactions on pattern analysis and machine intelligence 40.4 (2017).