Parallel U-Net: Improving Image Colorization Using Bounding Boxes with a Modified U-Net Architecture

Shrey Verma
Stanford University
shreyv@stanford.edu

Abstract

This paper proposes a novel approach to image colorization by enhancing the traditional U-Net architecture by providing context-aware information about the subjects in the image in terms of their bounding boxes. This modified U-Net is used as the generator for a GAN baseline, which is used to test the context-specific capabilities of this modified approach. I trained the modified generator model on 5000 images from the COCO dataset, using 3500 images for training and 1500 for validation, and compared its performance against a baseline U-Net without bounding box information. This approach highlights the potential of combining object detection with image colorization techniques to achieve superior results.

1. Introduction

Image colorization is the problem of inferring colors from greyscale images. This problem has diverse applications, including enhancing old photographs and films, contributing to art and animation, and aiding in historical archiving [3]. Image colorization not only serves as a practical tool for these applications but also functions as a self-supervised task that can possibly benefit various other computer vision tasks, such as image embedding and feature learning.

Traditional approaches have utilized convolutional neural networks (CNNs) and, more recently, generative adversarial networks (GANs) to produce realistic color images from their greyscale counterparts [8]. One of the most effective architectures for image colorization has been the U-Net, a type of CNN that excels in tasks requiring precise localization and segmentation due to its unique encoder-decoder structure with skip connections [8].

Despite the success of U-Net in image colorization, there are inherent limitations in its ability to understand and incorporate the context of the subjects within an image. This often results in less realistic colorizations, especially in complex scenes with multiple objects. To address this issue, I propose a modified U-Net architecture that leverages context-aware information through object detection using YOLOv8 [1]. By incorporating bounding boxes that specify the locations of different objects in the image, the modified U-Net, referred to as Parallel U-Net, can better understand the context and produce more accurate colorizations.

In this work, I integrate the modified U-Net as the generator in a GAN framework to test its context-specific capabilities. The generator receives the greyscale image and the corresponding bounding boxes of predefined classes, which provide crucial contextual information about the image content. This method allows the model to focus on different parts of the image, guided by the object locations, to improve the colorization process.

I trained this model on 5000 images from the COCO dataset [2], using 3500 images for training and 1500 for validation, and compared its performance against a baseline U-Net that does not utilize bounding box information [4]. The results indicate that the inclusion of bounding boxes tends to enhance the colorization quality, demonstrating the potential of combining object detection with image colorization techniques to achieve superior results.

This approach not only improves the quality of colorized images but also opens new avenues for using context-aware models in other computer vision tasks, where understanding the context of different parts of the image can lead to more accurate and meaningful outputs.

2. Related Work

Existing colorization techniques largely rely on memorizing object-color relationships in the network parameters. This presents several problems: the network must have enough capacity and sufficient training data to memorize these relationships, the model may generalize poorly, and it may color the same object with different colors in slightly different contexts. These issues are evident in previous works that often result in unrealistic colorizations, especially for complex scenes with multiple objects [3] [7].

Reference colorization is a relaxation of the general colorization problem where a reference image is provided in addition to the target image. The reference image shows similar objects to the target image and provides grounding for colorization choices. Vondrick et al. [5] demonstrated that supplying a reference frame in video colorization led to substantially improved results over prior techniques. Their approach extracted a feature embedding for each pixel in low-resolution versions of the reference and target images and used a softmax attention mechanism to retrieve the color directly from the reference image. This method assumed the color retrieved from the reference image to be the color of the corresponding pixel in the target image. However, colorization was not the primary focus of their work; it was a self-supervised task to aid in object tracking.

Despite the advancements in reference colorization, there are inherent limitations. First, objects present in the target image but not in the reference image may not have any corresponding color information, leading to inaccurate color retrieval. Second, changes in object color across subsequent frames, such as due to illumination changes, are not accounted for, resulting in inconsistencies.

To address these limitations, I propose a novel approach that integrates context-aware information through object detection using YOLOv8 [1]. By incorporating bounding boxes that specify the locations of different objects in the image, the modified U-Net architecture, referred to as Parallel U-Net, can better understand the context and produce more accurate colorizations. This method builds on the strengths of U-Net while leveraging object detection to enhance the colorization process.

Related work has also explored the use of attention mechanisms in colorization tasks. Yoo et al. [6] developed a model that employs an attention mechanism to improve colorization, but their model, data, and results are proprietary. This approach contributes an open implementation for further study and comparison, providing a comprehensive evaluation of the impact of incorporating object detection on colorization quality.

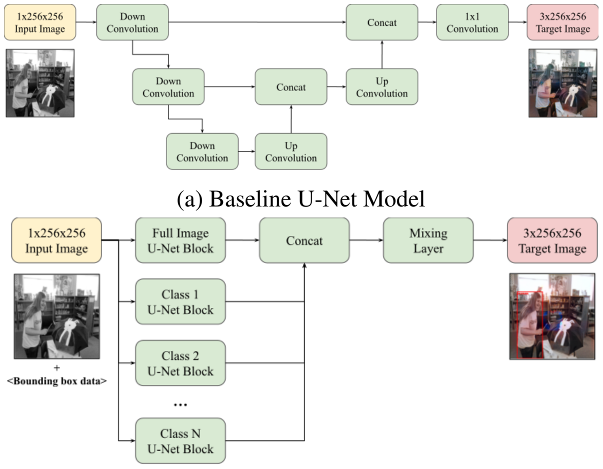
3. Methods

In this section, I will highlight the modifications made to the traditional U-Net architecture to incorporate context-aware information through bounding boxes using YOLOv8. I will also describe how this *Parallel U-Net* is integrated into a GAN framework to improve image colorization. The key components of this method are the modified U-Net architecture, the bounding box incorporation mechanism, and the mixing layer for combining outputs.

3.1. Modified U-Net Architecture

The traditional U-Net architecture consists of an encoder-decoder structure with skip connections. The encoder gradually reduces the spatial dimensions while increasing the number of feature maps, and the decoder progressively restores the spatial dimensions to the original size, leveraging the high-resolution features from the encoder through skip connections. This architecture is effective for tasks requiring precise localization, such as segmentation and colorization [4] [8].

To enhance the U-Net with context-aware information, I modified the architecture to process multiple parallel inputs corresponding to the bounding boxes of different object classes in the image. Each object class has a corresponding U-Net block that processes the region within the bounding box. Additionally, a separate U-Net block processes the entire image to capture the global context. The outputs of these parallel blocks are then combined using a mixing layer to produce the final colorized image.



(a) Baseline U-Net Model

(b) Modified U-Net Model with Bounding Boxes

Figure 1: Comparison of Baseline and Modified U-Net Models

3.2. Incorporating Bounding Boxes

Bounding boxes provide crucial contextual information about the objects present in the image. I utilize YOLOv8

[1] to detect objects and obtain their bounding boxes. For each image, I extract bounding boxes for up to five predefined classes (*person, bicycle, car, airplane, boat*). These bounding boxes are then used to create cropped regions of the image, which are resized to match the input dimensions of the U-Net blocks.

In my implementation, each bounding box is processed by a separate U-Net block. If a bounding box is not present for a particular class, I use a zero tensor as input to the corresponding U-Net block to maintain consistency. This ensures that the network structure remains the same regardless of the number of detected objects, and the missing information is effectively masked.

3.3. Mixing Layer

The mixing layer is designed to combine the outputs of the parallel U-Net blocks. After processing the cropped regions and the entire image, the outputs are concatenated along the channel dimension. This results in a tensor with dimensions $[batch\_size, (num\_classes + 1) * output\_c, height, width]$.

The concatenated output is then passed through a series of convolutional layers to mix the information from the different U-Net blocks. The first convolutional layer reduces the number of channels to 6, followed by a ReLU activation function. The second convolutional layer reduces the number of channels to $output\_c$, followed by a Tanh activation function to produce the final colorized image. This process ensures that the combined output incorporates both local and global contextual information from the image.

3.4. GAN Loss

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))]$$

The GAN loss function $\mathcal{L}_{GAN}$ is used to optimize the generator G and the discriminator D. Here, x represents the input greyscale image, y represents the real color image, and z is a noise vector. The generator aims to minimize $\log(1 - D(x, G(x, z)))$ while the discriminator aims to maximize $\log D(x, y) + \log(1 - D(x, G(x, z)))$.

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[||y - G(x, z)||_1]$$

The L1 loss function $\mathcal{L}_{L1}$ ensures that the generated image $G(x, z)$ closely matches the real image y in terms of pixel values. This helps to maintain the color distribution and structural integrity of the original image.

4. Dataset and Features

In this section, I will describe the dataset used for training and validation, along with the features extracted to aid in the colorization process.

4.1. Dataset

I used the COCO dataset for the experiments, which is a large-scale object detection, segmentation, and captioning dataset [2]. The dataset contains over 200,000 images with a wide variety of objects in diverse contexts. For this study, I selected 5000 images, with 3500 images allocated for training and 1500 for validation.

Each image in the dataset is accompanied by annotations that include object bounding boxes and class labels. These annotations are crucial for this method as they provide the context-aware information necessary for the modified U-Net architecture. The images in the dataset vary in resolution, but for consistency, I resized all images to 256x256 pixels.

4.2. Preprocessing

Before feeding the images into the model, several preprocessing steps were applied:

1. **Resizing**: All images were resized to a fixed resolution of 256x256 pixels to ensure uniform input dimensions for the U-Net blocks.

2. **Normalization**: The pixel values of the greyscale images were normalized to the range [-1, 1].

3. **Bounding Box Extraction**: Using YOLOv8 [1], I extracted bounding boxes for five predefined classes (*person, bicycle, car, airplane, boat*) in each image. For each class, only the bounding box with the highest score corresponding to that class was used. These bounding boxes were used to create cropped regions that were resized to match the input dimensions of the U-Net blocks.

4.3. Features

The key features used in this model are the L*a*b color space representations of the images and the bounding boxes for object detection:

1. **L*a*b Color Space**: I converted the input greyscale images to the L*a*b color space, where the L channel represents lightness, and the a and b channels represent color information. The L channel of the input image was fed into the U-Net blocks, while the a and b channels were used as ground truth for the colorization task.

2. **Bounding Boxes**: The bounding boxes provided the spatial locations of objects in the image. For each class, I created a cropped region within the bounding box and processed it through a corresponding U-Net block. This allowed the model to focus on specific objects and their context within the image.

The combination of these features enabled the modified U-Net to incorporate both local and global contextual information, leading to more accurate and realistic colorizations. The use of bounding boxes ensured that the model could attend to important regions in the image, improving the overall quality of the generated color images.

5. Experiments

To evaluate the performance of the proposed *Parallel U-Net*, I conducted experiments using the COCO dataset. The dataset was split into 3500 images for training and 1500 images for validation. I compared the performance of the modified U-Net against a baseline U-Net model without bounding box information. Both models were trained using a GAN framework, where the U-Net acted as the generator, and a PatchGAN was used as the discriminator.

I used the Adam optimizer with a learning rate of 2e-4 for both the generator and discriminator. The training process was conducted over 25 epochs.

I addition to the GAN Loss, which was used for training, I also included an L2 validation loss term to evaluate the models. The L2 loss function is defined as:

$$\mathcal{L}_{L2} = \left(\frac{1}{n} \sum_{i=1}^{n} \left((y_i - \hat{y}_i)^2 \right)^{0.5} \right)$$

where y_i is the ground truth color value and $\hat{y}_i$ is the predicted color value.

5.1. Results

The final losses for both models are summarized in Table 1.

Loss Type	**Modified U-Net**	**Baseline U-Net**
Discriminator Loss	0.5315	0.5480
Generator Loss	9.7835	10.2879
Validation Loss	37.9919	39.8897

Table 1: Final Loss Values for Modified and Baseline U-Net Models

5.2. Discussion

The results indicate that the modified U-Net model with bounding box information performs better than the baseline U-Net model. The final discriminator and generator losses for the modified U-Net are lower than those of the baseline model, suggesting improved adversarial training. Moreover, the modified U-Net achieved a lower validation loss compared to the baseline, indicating better generalization to unseen data.

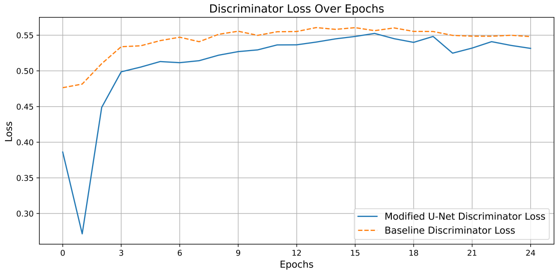


Figure 2: Discriminator Loss Over Epochs

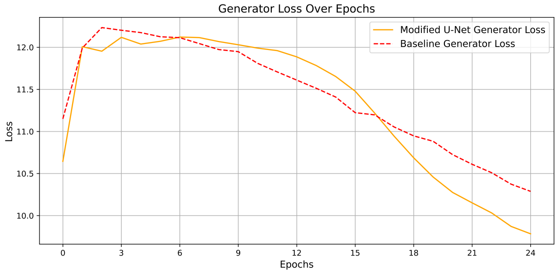


Figure 3: Generator Loss Over Epochs

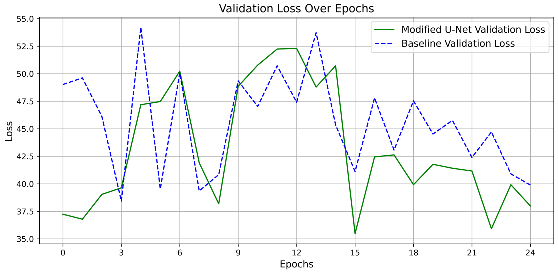


Figure 4: Validation Loss Over Epochs

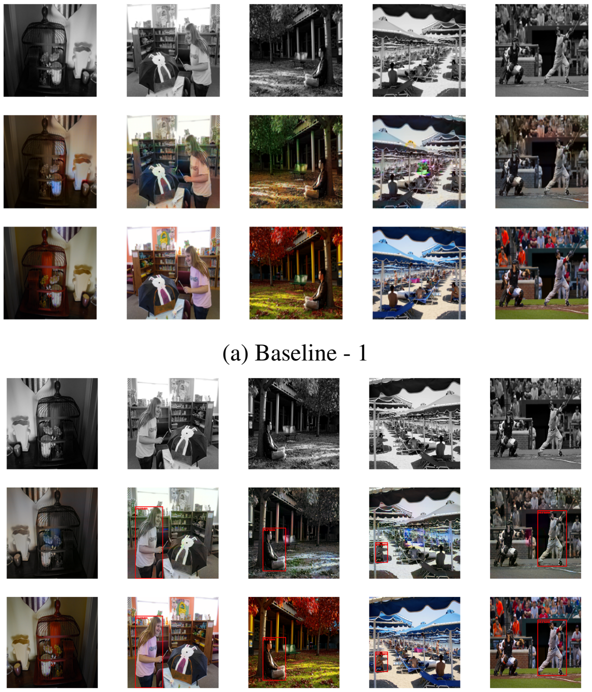
The inclusion of bounding box information allowed the modified U-Net to focus on specific regions of interest in the image, leading to more accurate colorization. This context-aware approach helped the model to better understand the spatial relationships and object structures within the image, resulting in more consistent colorizations, especially for the subject.

The loss curves (2, 3, 4) over the epochs show that the modified U-Net consistently outperforms the baseline

U-Net in terms of both discriminator and generator losses. The validation loss also demonstrates a more stable and lower loss for the modified U-Net, further validating the effectiveness of the proposed modifications.

Overall, the experiments suggest that integrating object detection with image colorization techniques is a promising tool to enhance the performance of colorization models. The modified U-Net architecture, with its ability to incorporate context-aware information, provides a promising direction for future research in image colorization and other related tasks.

Here are some of the images from the final run:



(a) Baseline - 1

(b) Modified U-Net - 1

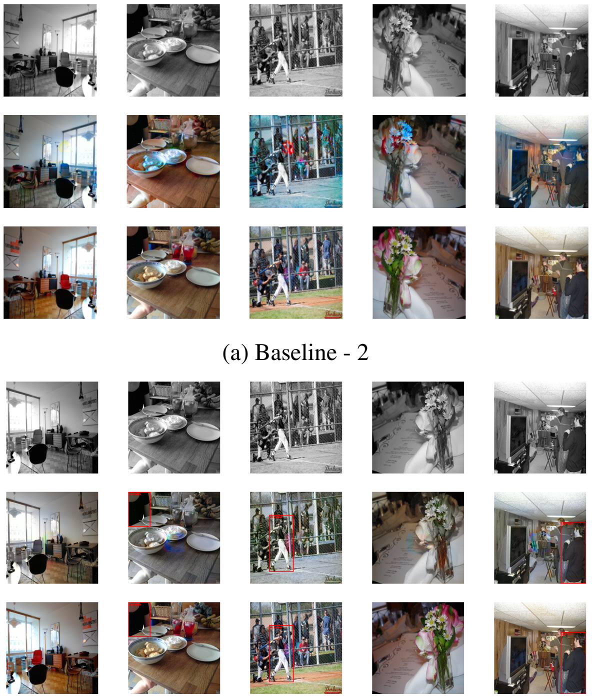
Figure 5: Comparison of Final Colorized Output - 1

6. Conclusion and Future Work

The final output image results show that the modified U-Net-based GAN provides more consistent colors for the subjects in the image.

However, training the baseline and the modified U-Net model for 100 epochs and 10,000 COCO Images can better show these results. Because of a time crunch, I could not train the models on the bigger dataset.

Apart from changes in the dataset, there is a huge potential for tuning the mixing layers to incorporate better the flow of information between the parallel U-Net blocks.



(a) Baseline - 2

(b) Modified U-Net - 2

Figure 6: Comparison of Final Colorized Output - 2

Another approach for tuning involves characterizing the features in terms of their super-classes; for example, a car, truck, and boat can all be classified as the *vehicle* class. Suppose the modified U-Net is trained this way. In that case, more information about the subjects can be passed into the individual U-Net blocks, and the generator can generalize better over the classes. Utilizing the super-classes will also help reduce the number of U-Net blocks significantly.

7. Contributions and Acknowledgements

7.1. Contributions

This project was a solo effort by me. The key contributions are as follows:

- **Model Development**: Design and implementation of the modified U-Net architecture, incorporating context-aware information through object detection and bounding boxes using YOLOv8.

- **Experimentation**: Conducting extensive experiments using the COCO dataset, training both the modified U-Net and the baseline U-Net models, and comparing their performance.

- **Analysis and Evaluation**: Analyzing the results, creating visualizations for loss curves and final colorized outputs, and evaluated the effectiveness of the proposed modifications.
- **Report Writing**: Writing the report, detailing the methodology, experiments, results, and discussions.

7.2. Acknowledgements

I would like to acknowledge the following:

- **COCO Consortium**: For providing the COCO dataset, which was instrumental in training and evaluating the models.
- **Ultralytics**: For developing YOLOv8, which was used for object detection and bounding box extraction.
- **The Course Instructors and TAs**: For their guidance and support throughout the course, which helped in the successful completion of this project.

References

[1] G. Jocher, A. Chaurasia, and J. Qiu. Ultralytics yolov8, 2023.

[2] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Doll'a r, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[3] . Saeed Anwar, Muhammad Tahir. Image colorization: A survey and datase. https://arxiv.org/pdf/2008.10774.

[4] M. Shariatnia. Baseline code. https://github.com/moein-shariatnia/Deep-Learning/tree/main/Image

[5] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy. Tracking emerges by colorizing videos, 2018.

[6] S. Yoo, H. Bahng, S. Chung, J. Lee, J. Chang, and J. Choo. Coloring with limited data: Few-shot colorization via memory-augmented networks, 2019.

[7] . Yuanzheng Ci, Xinzhu Ma. User-guided deep anime line art colorization with conditional adversarial networks. https://arxiv.org/pdf/1808.03240.

[8] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*, 2016.