

Application and Analysis of Black and White Image Coloring Based on Generative Adversarial Networks (GANs)

Ming Him Foun^a

School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland

Keywords: PatchGAN, U-Net, VGG, Image Colorization, ResNet.


Abstract: This paper demonstrates a comprehensive study on the application of Generative Adversarial Networks (GANs) for colorizing black-and-white images, employing the extensive COCO dataset for training and evaluating various deep learning frameworks. By integrating U-Net architecture with Residual Network (ResNet) 18 and Visual Geometry Group (VGG) 16 backbones within a PatchGAN framework, the study proposes a sophisticated method for adding color to grayscale images, aiming to create visually compelling and aesthetically pleasing results. The research adopts a systematic approach, beginning with image resizing and conversion to the Commission Internationale Eclairage lab (CIELAB) color space, followed by generator pretraining and subsequent PatchGAN training to finalize the colorization process. Through extensive experimentation, the study assesses the performance of the proposed models, revealing that the U-Net generator enhanced with a ResNet18 backbone significantly outperforms the VGG16 counterpart across multiple metrics, including Mean Squared Error (MSE), with a score of 1446.38961, Color Structural Similarity Index Measure (Color SSIM) of 0.87444, and 3.28116 for CIEDE2000. Despite building upon existing codes and frameworks, this study significantly advances the discourse in deep learning-based image colorization, emphasizing the comparative performance of different architectural choices and paving the way for future enhancements in the field.

1 INTRODUCTION

Colorizing historical black-and-white images enhances their artistic appeal and provides viewers with a more immersive experience, bridging the gap between the past and present. However, accurately determining the original colors of early photographs is challenging due to limited insights into historical color schemes. Nonetheless, the objective of colorization is not to achieve perfect accuracy but to create convincing illusions of authenticity, deceiving viewers into believing in the realism of the colored images. This technique is applied in various fields such as historical image restoration, movie colorization, and the enhancement of astronomy photographs.

Colorization is a complex process that entails assigning RGB color values to grayscale pixels in a visually plausible manner, enhancing the visual appeal and usability of images across various fields such as image recognition and object detection. The

primary challenge in colorization stems from the fact that grayscale images lack innate color details, making it difficult to accurately restore or assign colors in a way that authentically represents the original scene or object. The challenging nature of colorization has generated ongoing interest in the research community, driving continued innovation in the field. In recent years, image colorization has experienced a profound evolution fueled by groundbreaking advancements in deep learning methods. Utilizing the capabilities of machine learning algorithms, particularly convolutional neural networks (CNNs) (Dabas, 2020) (Varga, 2016) (Dias, 2020) and generative adversarial networks (GANs) (Xiaodong, 2020) (Wengling, 2018) (Cao, 2017), researchers have offered novel approaches for addressing the intricate challenges associated with colorization tasks. These advanced deep learning models are proficient at capturing subtle patterns and revealing concealed features within large datasets, thus offering compelling solutions to the complexities

^a <https://orcid.org/0009-0006-0184-4282>

of image colorization. The evolution of colorization methods has resulted in the emergence of fully automatic (Zhuge, 2018) (Larsson, 2016) and semi-automatic (Cheng, 2019) approaches. Fully automatic approaches offer end-to-end colorization but often lack control and may provide simplistic or unrealistic outcomes. In contrast, semi-automatic methods enable user guidance for more precise control, although they can be challenging for inexperienced users. Addressing the trade-offs between automation and controllability remains a key area of exploration in colorization research. Despite significant progress in colorization techniques, several challenges persist, including the need for large-scale datasets, issues with color consistency and diversity, and the presence of artifacts and loss of detail in colorized images. Evaluating the quality of colorization results and identifying areas for improvement remain active areas of research.

This study explores the utilization of GANs for colorizing grayscale images, utilizing the COCO dataset to evaluate and compare the efficacy of various deep learning models. It specifically focuses on a method that combines the U-Net architecture with Residual Network (ResNet) 18 and Visual Geometry Group (VGG)16 backbones within a PatchGAN framework, proposing an advanced technique for infusing color into grayscale images to produce visually compelling results. Extensive experiments were conducted to evaluate model performance, revealing that U-Net with a ResNet18 backbone outperforms the VGG16 model across multiple metrics. This research significantly contributes to the discourse on deep learning-based image colorization by comparing different architectural approaches and suggesting avenues for future research enhancements.

2 METHODOLOGIES

The network's workflow in Figure 1. commences by resizing the image to a 256×256 resolution, followed by a conversion into CIELAB color channels. and then converting the image into CIELAB channels. Subsequently, the generator undergoes pretraining before training alongside PatchGAN for the ultimate colorization output.

2.1 Dataset Description and Preprocessing

For this experiment, the COCO dataset was utilized, featuring more than 330,000 images, with annotations provided for 220,000 of these (Lin, 2014). This extensive dataset contains 1.5 million objects belonging to 80 object categories (e.g., person, car, elephant) and 91 stuff categories (e.g., grass, wall, sky). Within the COCO dataset, 10,000 images were chosen at random, of which 8,000 were designated for training and 2,000 for testing. The selected images were resized into 256×256 pixels. The selected images were converted from RGB color spaces into CIELAB color spaces which expresses color with three values: L^* for perceptual lightness, a^* for red and green, and b^* for blue and yellow which are the four unique colors perceived by human vision. The L^* perceptual lightness was used as the input layer identical to the black and white image to train and predict the a^* and b^* color values.

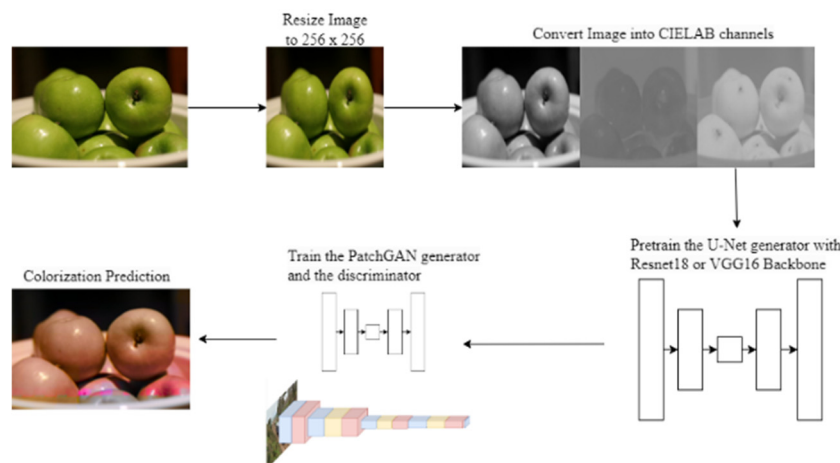


Figure 1: The pipeline of the model (Photo/Picture credit: Original).

2.2 Proposed Approach

The network pipeline integrates U-Net with ResNet18 and VGG16 backbones as the generator's architecture for image colorization tasks, with a focus on efficiently translating grayscale images into colored outputs. The U-Net structure, known for its encoder-decoder configuration with skip connections, is pretrained on the training dataset comprising pairs of grayscale and colored images. Pretraining the generator for grayscale image colorization ensures initial sample diversity and a smooth transition towards covering the entire target color distribution, resulting in more gradual image evolution (Grigoryev, 2022). The PatchGAN discriminator evaluates the authenticity of the generated images on a localized patch basis, refining the generator's output through adversarial training. The training procedure involves a cycle of updating the discriminator using both real and synthesized images, followed by refining the generator to create images that are progressively more difficult to distinguish from real ones. This comprehensive approach, illustrated in Figure 1, combines advanced architectures and a detailed training strategy to successfully colorize grayscale images with high quality.

2.2.1 U-Net

The utilization of U-Net as the foundational architecture for the network's generator proves advantageous for tasks centered around image-to-image translation (Isola, 2017). In multiple image translation scenarios, a significant amount of fundamental information is commonly exchanged between the input and output and becomes advantageous to directly transfer this information across the network. In order to provide the generator with an effective mechanism to overcome information bottlenecks, skip connections are introduced, inspired by the architecture of a "U-Net" (Ronneberger, 2015). These connections are strategically placed between every layer i and its corresponding layer $n - i$, where n represents the total number of layers. Each skip connection functions by concatenating all channels at layer i with those at layer $n - i$.

This framework revolves around integrating the U-Net structures as the central generator as illustrated in Figure 2. while incorporating diverse backbone architectures such as ResNet18 and VGG16 to bolster feature extraction and representation. The U-Net generators, augmented with ResNet18 and VGG16

backbones, undergo pretraining on a curated training dataset comprising pairs of grayscale and colored images. Employing the L1 loss function during pretraining and optimizing with the Adam optimizer contributes to refining the generators' capacity to generate realistic color predictions from grayscale inputs.

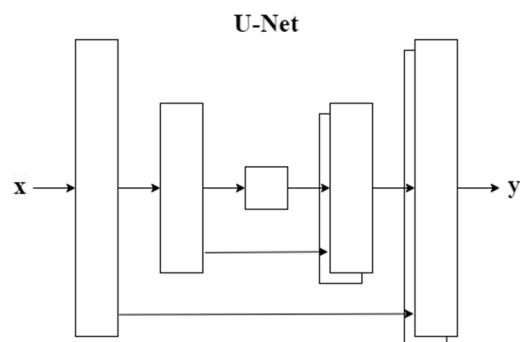


Figure 2: The structure of U-Net (Photo/Picture credit: Original).

The "U-Net" in Figure 2. configuration is an encoder-decoder structure distinguished by skip connections, where 'X' is the greyscale input and 'Y' is the resultant colorized image. This design facilitates the direct flow of information across the network, allowing the model to preserve details from the input for a precise colorization output.

2.2.2 PatchGAN

The discriminator utilizes a convolutional PatchGAN which is a model comprised of stacked blocks of Convolutional layer, Batch Normalization layer, and Leaky ReLU layer, as illustrated in Figure 3, to decide whether the input image is fake or real. The first and last blocks do not use normalization and the last block has no activation function. The PatchGAN solely penalizes structural inconsistencies within patches of an image, operating at a localized scale rather than , rather than evaluating the image as a whole. By applying convolution across the entire image, the discriminator combines all responses to produce its final output.

Initially, the discriminator undergoes training as fake images generated by the generator are inputted into the discriminator. Subsequently, a batch of real images from the training set is fed into the discriminator and labeled as real. The losses incurred from both fake and real images are summed up, averaged, and subjected to the backward operation to update the discriminator. Then, the generator is trained by feeding fake images into the discriminator

with the intention of deceiving it into categorizing them as real. The adversarial loss is computed accordingly. Additionally, L1 loss is calculated by measuring the discrepancy between the predicted and target channels, then multiplied by the coefficient ($\lambda=100$) to balance both losses. This resultant loss is added to the adversarial loss, and the backward method is invoked to update the generator's parameters. Network optimization involves alternating between conducting a single gradient descent step on the discriminator and another step on the generator.

Diagram of a PatchGAN discriminator architecture, processing an input of size $3 \times 256 \times 256$ to produce an output matrix of size $1 \times 30 \times 30$, highlighting the transformation of high-resolution color images through convolutional layers to evaluate the authenticity of generated images on a patch-wise basis.

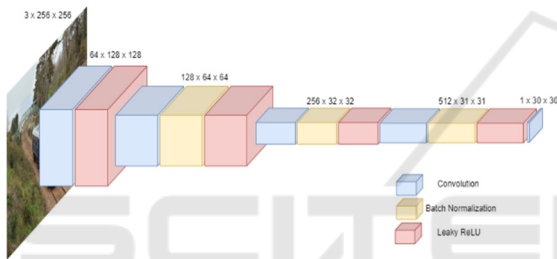


Figure 3: The architecture of PatchGAN (Photo/Picture credit: Original).

2.2.3 Loss Function

The selection of an appropriate loss function significantly impacts the training process of deep learning models. In the context of image colorization, the L1 loss function proves to be ideal for accurately quantifying the difference between the predicted and target channels. Compared to L2 loss, L1 loss tends to produce less blurry images, because L1 loss is less sensitive to outliers and does not penalize large errors as heavily as L2 loss.

$$\mathcal{L}_{L1}(G) = E_{x,y,z}[\|y - G(x,z)\|_1] \quad (1)$$

Let x represent the L^* grayscale image, y stand for the a^* and b^* two color channels of the real image, and z denotes the input noise vector for the generator.

Equation (1) calculates the expected value of the L1 norm of the difference which corresponds to the sum of the absolute differences between the real target channels y and the generated image $G(x,z)$.

The adversarial (GAN) loss was also utilized during the training of PatchGAN:

$$\mathcal{L}_{cGAN}(G,D) = E_{x,y}[\log D(x,y)] + E_{x,z}[\log(1 - D(x,G(x,z)))] \quad (2)$$

Breaking down equation (2), $E_{x,y}[\log D(x,y)]$ represents the average log likelihood that the discriminator assigns to the real data pairs (x,y) . The discriminator tries to maximize this term, meaning it aims to correctly identify real pairs as real. $E_{x,y}[\log(1 - D(x,G(x,z)))]$ represents the generator G creates an image from the grayscale image x and the noise vector z , and then the discriminator D evaluates this generated image paired with its input x . The generator tries to minimize this value by getting better at generating images that the discriminator will classify as real. This means that G aims to fool D into thinking the generated images are real.

Combining equation (1) and equation (2) together:

$$G^* = \underset{G}{\operatorname{argminmax}}_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L1}(G) \quad (3)$$

where the λ coefficient serves to balance the contribution of the two losses towards the final loss.

2.3 Implementation Details

The study used Python 3.10 and the Pytorch library with the built in Resnet18 and VGG16 model. The proposed network was trained on an A100 GPU. The model's initialization involved utilizing values drawn from a Kaiming normal distribution. During the initial pretraining of the generator, the learning rate of the Adam optimizer was set at 0.0001. Subsequently, for the training of the PatchGAN, the Adam optimizer was used with a learning rate of 0.0002 with momentum parameters $\beta_1=0.5$, $\beta_2=0.999$. The λ used in equation (3) is 100.

3 RESULTS AND DISCUSSION

3.1 Pretrained Generator Loss Curve

Figure 4. shows the training loss curves for a U-Net generator with two different backbone architectures, ResNet18 and VGG16, during the pretraining phase for image colorization.

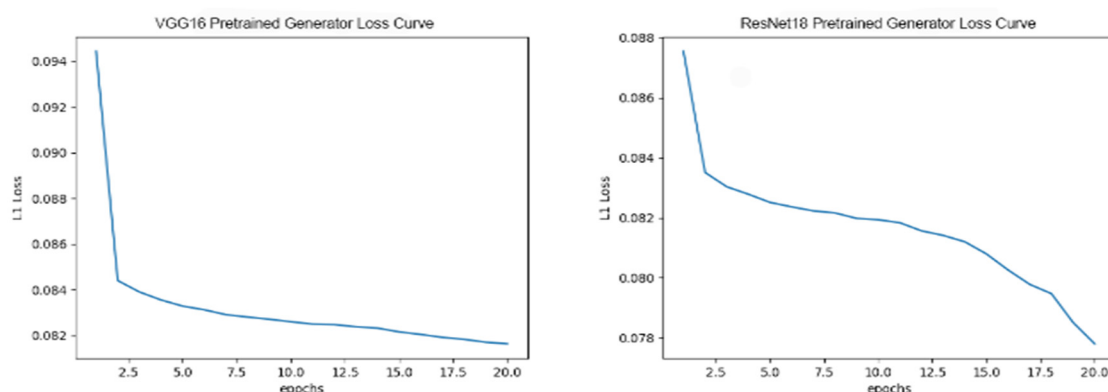


Figure 4: Loss curves of pretrained generators using VGG16 and ResNet18 backbones (Photo/Picture credit: Original).

Both architectures show a common trend of sharp improvement at the beginning, which is typical as the optimizer corrects high initial errors quickly. ResNet18's generator loss decreases to a lower value than VGG16's by the end of 20 epochs, which might suggest that for this specific task of image colorization, the ResNet18 backbone is more effective or efficient. The loss with the ResNet18 backbone appears to be reducing at a more consistent rate compared to the VGG16 backbone, which shows a slightly more pronounced plateau. This could indicate that the ResNet18 architecture is learning more steadily or that it is a better fit for the nuances of the colorization task.

The choice of architecture has a significant impact on the training process. ResNet architectures are known for their residual connections which effectively address the issue of vanishing gradients by allowing for the flow of information through shortcut paths, thereby facilitating more efficient training of deep neural networks. This could be why the ResNet18 backbone shows a more consistent learning rate without as pronounced a plateau as VGG16. The VGG16 architecture is simpler and more straightforward but lacks these residual connections, which might lead to less efficient training in deeper layers as the network learns.

The continued optimization of the generator is crucial for the quality of colorization. Reduced loss indicates that the generated images are increasingly aligning with the actual distribution of colored images, which is the primary goal of colorization tasks. The effectiveness of the U-Net generator with either ResNet18 or VGG16 backbones in learning the colorization mapping can directly impact the visual quality of the colorized images.

3.2 PatchGAN Generator Loss Curve

For the ResNet18 backbone shown in Figure 5, the generator loss curve demonstrates a consistent and smooth decrease in both adversarial (GAN) loss and L1 loss over 100 epochs. The adversarial loss begins around a value of 12 and steadily declines to approximately 2, while the L1 loss experiences a minor decrement, indicating a relative stability in the pixel-wise accuracy of the generated images. The overall generator loss mirrors the trend of the adversarial loss, reflecting the generator's improvement in producing images that progressively align more closely with the target distribution.

Conversely, the VGG16 backbone displays a slightly more erratic pattern in its loss curves. The GAN loss commences at a lower value compared to ResNet18 but exhibits a transient increase before continuing a downward trajectory, eventually converging to a value near 6. The L1 loss remains mostly constant throughout the training epochs, suggesting a consistent level of pixel accuracy from early in the training process.

The discriminator loss curves for both architectures commence at a loss value indicative of uncertainty when distinguishing between real and fake images. Across the training epochs, both curves for fake and real images converge to a loss value marginally above 0.5, an ideal scenario indicating the discriminator's inability to differentiate effectively between the two types of images, thus signifying a well-trained generative model.

The pretraining of the generators appears to be beneficial, as evidenced by the downward trend in loss, implying an effective learning process. However, the fluctuations, especially in the VGG16 curve, suggest an adjustment phase within the learning process, potentially due to the generator exploring

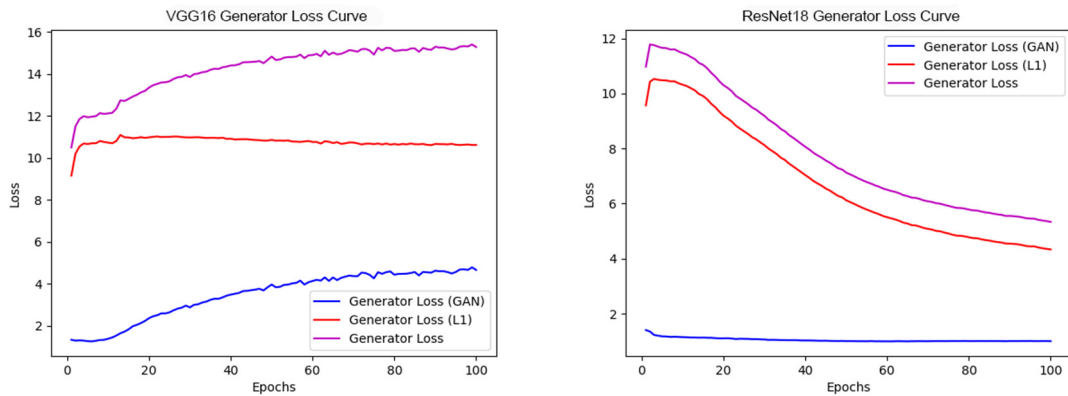


Figure 5: Loss curves of patchGAN generators using VGG16 and ResNet18 backbones (Photo/Picture credit : Original).

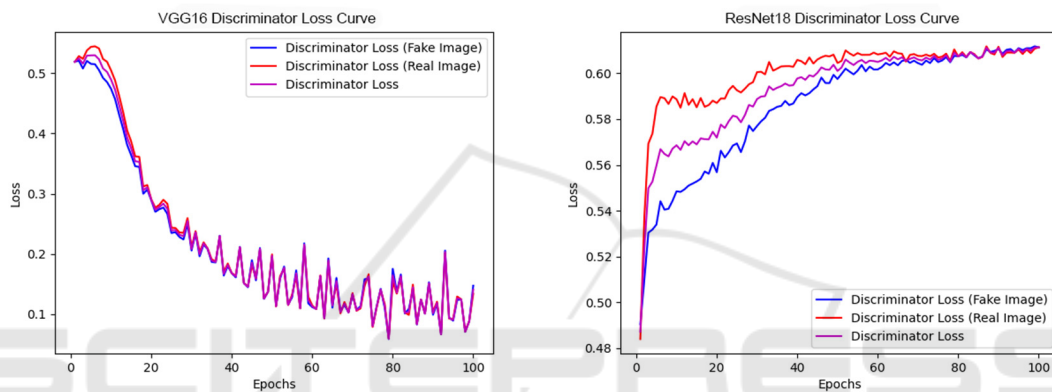


Figure 6: Discriminator loss curves for patchGAN with VGG16 and ResNet18 generators (Photo/Picture credit : Original).

various strategies to enhance image quality before converging to a more optimal solution.

The comparative analysis of the ResNet18 and VGG16 backbones reveals that ResNet18 may offer a more stable and consistent training for the colorization task in this context, as indicated by the smoother loss curves and lower final loss values. These observations emphasize the importance of choosing a suitable architecture backbone, as it profoundly influences both the training process and the eventual performance of the generative model.

3.3 PatchGAN Generator Loss Curve

For the VGG16 Discriminator Loss Curve as displayed in Figure 6, there is a sharp decline in both the discriminator loss for fake and real images, indicating the discriminator rapidly learning and improving its ability to distinguish between the two. The discriminator starts with random weights and quickly adjusts to the data, the loss for both types of images continues to decrease as more epochs were trained. The continuous decrease in the discriminator loss indicates that the generator with the VGG16

backbone is becoming less effective at fooling the discriminator over time. This overly efficient discriminator is not desirable for a GAN network, as it can cause the generator to stagnate and stop improving. The ideal scenario is a balanced adversarial contest, where both the generator and discriminator progressively improve.

Conversely, for the ResNet18 Discriminator Loss Curve, there is a sharp ascent in the discriminator loss for both real and fake images and then plateau, oscillating around a value just above 0.6, suggesting that the discriminator is becoming equally uncertain about the authenticity of both real and generated images. This uncertainty is the desired outcome in adversarial training, indicating that the generator is improving and producing images that increasingly resemble the real images. The upward trend in the loss may suggest that the generator has reached a new level of capability, generating images that are more sophisticated and harder for the discriminator to classify correctly which might push the discriminator to learn more complex and abstract features, resulting in a more effective model for image colorization.

The graphs reveals that the ResNet18-based generator significantly influences discriminator performance by initially creating highly complex images that deceive the discriminator, before reaching a steady plateau, suggesting superior image complexity and realism. Conversely, the VGG16-based generator leads to a consistently decreasing loss for its discriminator, implying its generated images are easier to classify as fake, likely due to the VGG16's inferior learning or optimization within the adversarial setup, resulting in a less effective challenge to the discriminator and simpler image production.

Table 1: Performance metrics of the colorization models.

	VGG16	ResNet18
MSE	2091.98364	1446.38961
Color SSIM	0.85048	0.87444
CIEDE2000	4.234501	3.28116

Table 1 shows that the ResNet18 backbone has a lower MSE of 1446.38961 compared to the VGG16 backbone of 2091.98364, suggesting that ResNet18 is more accurate in reproducing the original image colors. However, MSE treats all errors equally where large errors in less important areas might be weighted the same as small errors in crucial areas. The Color SSIM value of 0.87444 for ResNet18 slightly surpasses the score of 0.85048 achieved by VGG16, indicating a marginally superior image quality. This comparison suggests that images colorized utilizing the ResNet18 backbone generator are perceived to bear a closer resemblance to the original images. Color SSIM adopts an approach where each color channel is analyzed independently, neglecting the interdependence among the channels that significantly influences color perception. The lower CIEDE2000 score for ResNet18 of 3.28116 compared to VGG16 which is 4.234501 indicates that the color differences between the original and the colorized images are less perceptible when using ResNet18, suggesting superior colorization quality. Table 1 has shown that ResNet18 appears to outperform VGG16 in the context of image colorization with U-Net as the generator architecture within PatchGAN. ResNet18 shows lower error in pixel values and higher similarity in structural color as perceived by human vision.



Figure 7: Original colored image (Photo/Picture credit: Original).

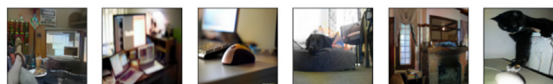


Figure 8: Colored image generated by VGG16 generator (Photo/Picture credit: Original).

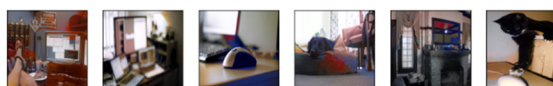


Figure 9: Colored Image generated by ResNet18 generator (Photo/Picture credit: Original).

Upon analyzing six samples from the test dataset, colored using different generators, notable differences emerged. As depicted in Figure 8, the generator employing the VGG16 architecture managed to color human parts accurately, despite the presence of several color artifacts. Conversely, the generator utilizing the ResNet18 architecture, as shown in Figure 9, precisely segmented human legs but failed to apply the correct colorization. Overall, the VGG16 generator succeeded in colorizing the images, though it introduced some noise, with color inaccuracies notably in pixels not corresponding to the primary image content, and an inclination to color objects with red and blue, specifically coloring shadows blue. On the other hand, the ResNet18 generator distinctly outlined the objects within the image, yet it inaccurately colored certain elements, such as human parts, rendering them a grey hue, which diverges from the original image presented in Figure 7. Overall, the ResNet18 images seem to display better color saturation, contrast, and accuracy, likely leading to a better representation of the original images. The VGG16 images, while still recognizable and maintaining the general color scheme, might suffer from a reduction in color vibrancy and contrast, affecting the overall fidelity of colorization.

Overall, the ResNet18 architecture demonstrated superior performance in object delineation and the preservation of structural coherence in the images, notwithstanding the presence of some colorization discrepancies.

4 CONCLUSIONS

In conclusion, this thesis has delved into the domain of image colorization using the COCO dataset, employing advanced deep learning techniques to tackle the complexities of the task. The proposed methodology exploits the innovative integration of U-Net architecture with embedded VGG16 and ResNet18 backbones, leveraging their robust feature extraction capabilities to enhance the colorization process. Throughout this research, a meticulous process was followed, involving pretraining the generators on grayscale images and employing PatchGAN discriminators to refine the generation of color images. Experimental results demonstrate that the U-Net architecture with a ResNet18 backbone outperforms its VGG16 counterpart in terms of Mean Squared Error, Color SSIM, and CIEDE2000 scores. Future work will focus on refining the model's precision in colorization to address the identified shortcomings. Specifically, efforts will be directed towards improving the model's proficiency in processing intricate textures and accurately coloring objects within their natural color ranges, crucial for achieving a higher degree of perceptual color accuracy in the colorized images.

REFERENCES

- Dabas., Chetna., et al. (2020). Implementation of image colorization with convolutional neural network. *International Journal of System Assurance Engineering and Management*, vol. 11(3), pp: 625-634.
- Varga., Domonkos., and Tamás, S., (2016). Fully automatic image colorization based on Convolutional Neural Network. *International Conference on Pattern Recognition*, ICPR.
- Dias., Maria., et al. (2020). Semantic segmentation and colorization of grayscale aerial imagery with W - Net models. *Expert systems*, vol. 37.6, p: 12622.
- Xiaodong, K., et al. (2020). Thermal infrared colorization via conditional generative adversarial network. *Infrared Physics & Technology*, vol. 107, p: 103338.
- Wengling, C., and James, H., (2018). Sketchygan: Towards diverse and realistic sketch to image synthesis. *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Cao, Y., et al. (2017). Unsupervised diverse colorization via generative adversarial networks. *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD*, p: 18–22.
- Zhuge., Jingjing., Jiajun L., and An, W., (2018). Automatic colorization using fully convolutional networks. *Journal of Electronic Imaging*, vol. 27(4), pp: 043025-043025.
- Larsson., Gustav., Michael, M., and Gregory, S., (2016). Learning representations for automatic colorization. *Computer Vision–ECCV*.
- Cheng., Zijuan., Meng, F., and Jingbo M., (2019). Semi-auto sketch colorization based on conditional generative adversarial networks. *International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI*.
- Tsung-Yi, L., et al. (2014). Microsoft coco: Common objects in context. *Computer Vision–ECCV*.
- Grigoryev., Timofey., Andrey, Voynov., and Artem, B., (2022). When, why, and which pretrained GANs are useful. arXiv:2202.08937.
- Isola., Phillip., et al. (2017). Image-to-image translation with conditional adversarial networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Ronneberger., Olaf., Philipp, F., and Thomas, B., (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical image computing and computer-assisted intervention–MICCAI*.