International Journal of Multidisciplinary Engineering in Current Research - IJMEC

Volume 10, Issue 4, April-2025, http://ijmec.com/, ISSN: 2456-4265

Image Colorization Using Pix2Pix with cWGAN and Residual U-Net Architecture

Mr. MD ImadUddin[*1], Mr. Dawood Khan[*2], Mr. MD Mujahid[*3], MD Shezaad[*4]

[*1] Assistant Professor, Dept. of AIML, Lords Institute of Engineering and Technology

[*2, 3,4] B.E Student Dept. of AIML, Lords Institute of Engineering and Technology

mohdimaduddinmohd@gmail.com[*1], dawood@gmail.com[*2], mujahid@gmail.com[*3]

alhaqshahzaad03@gmail.com[*4]

ABSTRACT

This paper presents an advanced image colorization approach leveraging a hybrid architecture combining Pix2Pix and conditional Wasserstein Generative Adversarial Network (cWGANs). Traditional image colorization is a challenging, ill-posed problem due to the need to infer plausible color values from grayscale inputs. To address this, we propose a supervised deep learning pipeline that conditions the generator on grayscale images while employing the Wasserstein loss with gradient penalty to stabilize GAN training and enhance color realism. The generator adopts a U-Net structure to preserve spatial fidelity, and the discriminator evaluates image realism in a conditional setup. The system is trained on grayscale-color image pairs transformed into the Lab color space, where the model learns to predict the 'ab' channels from the 'L' channel. The use of PyTorch Lightning ensures modular and scalable experimentation. Preliminary results demonstrate superior performance in producing sharp, semantically consistent colorizations compared to baseline GAN models, with quantitative assessments using Inception Score and perceptual losses. This research contributes to both aesthetic and practical applications, such as restoring historical photographs and aiding visual understanding in scientific imaging.

1. INTRODUCTION

1.1GENERAL

Color plays a crucial role in human perception, aiding not just in visual differentiation but also in emotional and contextual understanding of scenes. However, many forms of media such as historical photographs, medical scans, and satellite imagery are often available only in grayscale. This limitation motivates the development of automatic image colorization techniques. Traditionally, image colorization was achieved through manual or semi-automatic processes, which were time-consuming and labor-intensive. With the rise of deep learning, particularly Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs), it has become feasible to automate the process and achieve impressive results. Among these, the Pix2Pix framework has gained popularity due to its image-to-image translation capabilities. However, standard GANs often struggle with mode collapse and training instability, prompting researchers to explore more stable variants like the Wasserstein GAN (WGAN).

1.2 PROJECT OVERVIEW

This project explores the application of a Conditional Wasserstein GAN (cWGAN) combined with the Pix2Pix architecture for automatic image colorization. The core idea is to

train a U-Net-based generator to predict color components (ab channels) of a grayscale image (L channel) in the Lab color space, while a discriminator conditioned on the same grayscale input judges the realism of the generated color images. The WGAN framework improves stability and convergence by using the Wasserstein loss function with gradient penalty, addressing common pitfalls of traditional GANs. The training process is implemented using PyTorch Lightning to ensure a modular, reproducible, and scalable workflow. Extensive experiments were conducted on image datasets to validate the model's performance through both qualitative visual comparisons and quantitative metrics such as Inception Score and perceptual loss.

1.3 OBJECTIVE

- Design and implement a Conditional WGAN-Pix2Pix model for image colorization using grayscale inputs.
- Incorporate Wasserstein loss with gradient penalty to stabilize GAN training and improve color realism.
- Evaluate model performance using Inception Score, perceptual loss, and qualitative visual assessment.
- Demonstrate the effectiveness of conditional GANs in solving ill-posed image-to-image translation problems like colorization.

2. LITERATURE SURVEY

1. Instance Normalization: The Missing Ingredient for Fast Stylization (2016) Authors**:** Dmitry Ulyanov, Andrea Vedaldi, Victor Lempitsky Proposed instance normalization, which improves the quality and speed of style transfer and can be applied to colorization tasks.

2. Image-to-Image Translation with Conditional Adversarial Networks (2016)

Authors: Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros

Introduced the Pix2Pix framework, utilizing CGANs for various image translation tasks, including grayscale to color image conversion.

3. ChromaGAN: Adversarial Picture Colorization with Semantic Class Distribution (2020)

Authors: Patricia Vitoria, Lara Raad, Coloma Ballester

Proposed ChromaGAN, integrating semantic information into the adversarial learning process to enhance colorization realism.

4. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs (2017)

Authors: Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, Bryan Catanzaro

Developed a method for high-resolution image synthesis using CGANs, which can be adapted for detailed colorization tasks.

5. User-Guided Deep Anime Line Art Colorization with Conditional Adversarial Networks (2018)

Authors: Yuanzheng Ci, Xinzhu Ma, Zhihui Wang, Haojie Li, Zhongxuan Luo

Presented a CGAN-based approach for colorizing anime line art, incorporating user inputs to guide the colorization process.

6. Let There Be Color!: Joint End-to-End Learning of Global and Local Image Priors for Automatic Image Colorization (2016)

Authors: Satoshi Iizuka, Edgar Simo-Serra, Hiroshi Ishikawa

Introduced a model that learns global and local image features jointly for automatic colorization, achieving realistic results.

7. Deep Colorization (2016)

Authors: Gustavo Ghiasi, Honglak Lee

Proposed a deep learning approach for image

colorization that predicts chrominance values from grayscale images using CNNs.

8. Automatic Image Colorization via Multimodal Predictions (2016)

Authors: Richard Zhang, Phillip Isola, Alexei A. Efros

Presented a model that predicts multiple plausible colorizations for a single grayscale image using a classification loss.

9. Learning Representations for Automatic Colorization (2016)

Authors: Larsson, M., Maire, M., Shakhnarovich, G.

Developed a method that leverages deep representations to automatically colorize grayscale images with high fidelity.

10. Deep Exemplar-Based Video Colorization (2019)

Authors: Yifan Zhang, Zhaowen Wang, Zhe Lin, Hairong Qi

Introduced a deep learning framework that uses reference color images to guide the colorization of grayscale videos.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Traditional image colorization systems have relied heavily on manual or semi-automated techniques, requiring user input in the form of color scribbles, reference images, or region-specific hints. While these approaches offered some degree of control, they were time-consuming and demanded expert intervention. With the advent of deep learning, automatic colorization became feasible using Convolutional Neural Networks (CNNs). Early models trained CNNs in a regression-based manner to predict chrominance values, often resulting in desaturated or blurry outputs due to the averaging nature of regression loss functions. Some models improved results using standard Generative Adversarial Networks (GANs), but these suffered from unstable training, mode collapse, and poor convergence behavior, limiting their effectiveness in generating vivid and realistic colors.

Limitations of Existing Systems:

- Many traditional models struggle with understanding semantic context, often applying unrealistic or inconsistent colors to objects (e.g., blue grass or red skies).
- Standard GAN-based models can suffer from limited color diversity or repetitive outputs, failing to generalize well across diverse scenes.
- Existing systems sometimes produce artifacts such as color bleeding, halos, or texture distortions—especially around edges or fine details.

3.2 PROPOSED SYSTEM

The proposed system leverages deep generative learning, specifically a Conditional Wasserstein GAN with Pix2Pix architecture, to automatically colorize grayscale images by predicting chrominance channels from luminance data in the Lab color space.

Key Features:

- ResNet Based generator conditioned on grayscale input to predict color channels.
- Combined L1 loss and adversarial loss for improved color accuracy and texture generation
- WGAN with gradient penalty for stable and realistic training.

3.2.1 ADVANTAGES

- The use of a WGAN-based objective results in sharper, more vibrant color outputs compared to traditional GAN or regression models.

- Gradient penalty addresses common GAN issues like mode collapse and instability, leading to more reliable convergence.
- Conditioning on grayscale input allows the generator to infer color based on learned semantic understanding.

4. REQUIREMENT SPECIFICATIONS

4.1 SOFTWARE REQUIREMENTS

- Language: Python – for machine learning and data science support
- Platform: Google Colab or local Jupyter Notebook – cloud-based GPU/CPU access
- Libraries:
- Data Handling: NumPy
- Image Processing: OpenCV, scikit-image
- ML: PyTorch, Pytorch Lightning
- Visualization: Matplotlib, Seaborn
- Purpose: To enable efficient development, training, and evaluation of the deep learning model for Alzheimer's diagnosis

4.2 HARDWARE REQUIREMENTS

- Development Machine: PC with 8 GB RAM, Intel i5 / Ryzen 5 (minimum) for local development.
- GPU Requirement: NVIDIA GTX 1050+ locally; Tesla T4/P100 on Google Colab for better performance.
- Cloud Resource: Google Colab (Intel Xeon, 2 vCPUs, 13 GB RAM) for scalable training.
- Storage: At least 100 GB SSD; 256 GB SSD recommended for faster data handling.
- Setup: Hybrid environment (local machine + Google Colab) for efficient training and evaluation
- Purpose: To ensure optimal performance, minimize training time, and support iterative development of the deep learning colorization model.

5. SYSTEM DESIGN

5.1 SYSTEM ARCHITECTUR E

The proposed system for early Alzheimer's diagnosis uses a ResNet-based deep learning model trained on MRI and PET scan images. The process starts with the collection of labeled neuroimaging data, which is then preprocessed (resizing, normalization, and augmentation) to ensure consistency across the dataset. The preprocessed images are fed into the ResNet model, which extracts hierarchical features through residual learning. This allows the model to overcome vanishing gradient issues and effectively classify images into stages of Alzheimer's: Non-Demented, Very Mild Demented, Mild Demented, and Moderate Demented. The dataset is split into training, validation, and test sets. The model is trained and evaluated using performance metrics like accuracy, sensitivity, and specificity. Hyperparameter tuning is performed to optimize the model's performance. The best-performing model is then selected for deployment, offering an automated solution for early Alzheimer's detection in clinical settings.

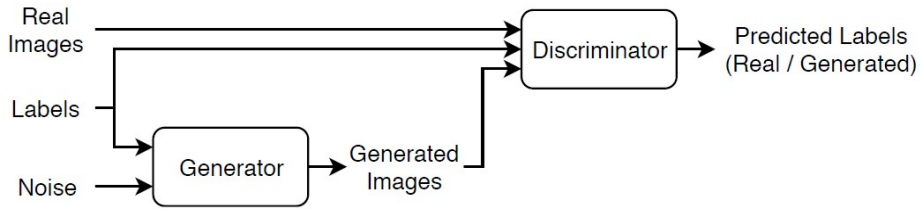

fig 5.1.1 cGAN model flow

ISSN: 2456-4265
IJMEC-2025

5.

2 UML DIAGRAMS

1. Use Case Diagram – Workflow

Shows how external users (e.g., medical professionals) interact with system functionalities like loading MRI/PET images, preprocessing data, training the model, and classifying Alzheimer's stages.

2. Class Diagram – Workflow

Represents system structure with classes like ImageLoader, DataPreprocessor, ModelTrainer, ModelEvaluator, etc., including their attributes, methods, and relationships.

3. Object Diagram – Workflow

Displays runtime instances of classes (e.g., imageLoader1, preprocessorA) and how data flows between them during system execution.

4. Sequence Diagram – Workflow

Depicts the order of operations: image loading → preprocessing → model training → Alzheimer's stage classification → output diagnosis, showing interactions over time.

5. Activity Diagram – Workflow

Illustrates the process flow from loading MRI/PET images to Alzheimer's stage classification, including decision points like "Is data ready?" or "Is model performance satisfactory?"

6. State Diagram – Workflow

Shows system states like Idle, Loading Images, Preprocessing, Training, Classifying, and transitions based on events like data availability or model evaluation completion.

7. Component Diagram – Workflow

Breaks the system into components: UI, Image Preprocessing, Model Trainer, Classifier, and Output, showing their interconnections.

8. Deployment Diagram – Workflow

Maps software modules onto hardware (local machine, cloud-based resources like Google Colab), showing network communication between image loading, training, classification, and user systems.

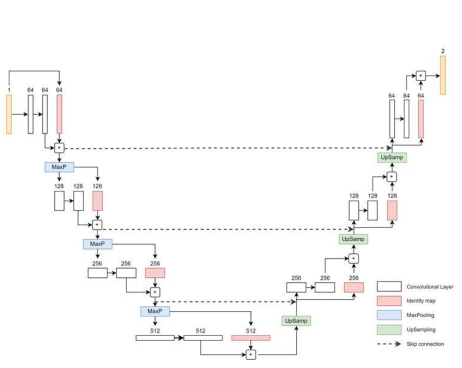

fig 5.2.1 UNet Architecture

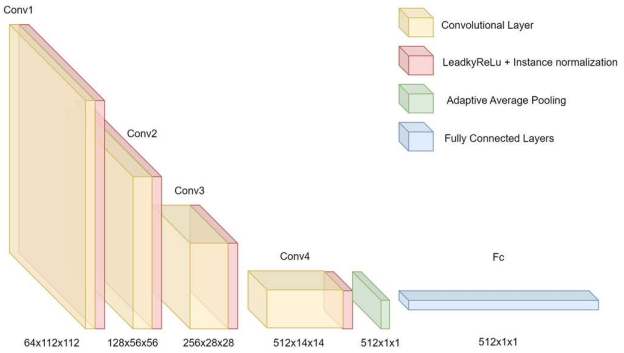

fig 5.2.2 CNN Architecture

5.3 MODULES

1. Image Loading Module

ISSN: 2456-4265
IJMEC-2025

Loads MRI/PET images from the dataset directories, categorizes them based on Alzheimer's stages, and prepares the data for further processing.

2. Data Preprocessing Module

Prepares the dataset for training by applying image augmentation, resizing, and normalizing the images using the ImageDataGenerator for better model performance.

3. Feature Extraction & Augmentation Module

Augments the image data using random transformations (rotation, flipping) and prepares it for feeding into the ResNet50 model to extract deep features.

4. Model Architecture Module

Defines the ResNet50-based model, with additional layers (Dense, Flatten) for classification, and compiles the model using the Adam optimizer and categorical cross-entropy loss function.

5. Model Training & Evaluation Module

Trains the model on the training data, validates it using the validation set, saves the model at checkpoints, and evaluates performance on test data with accuracy, precision, recall, and F1-score.

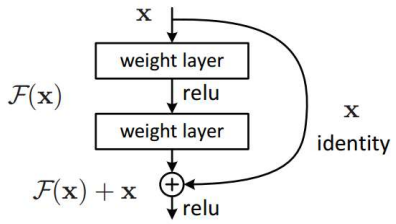

fig 5.3.1 Unet with ResBlock for Sematic Segmentation

6. IMPLEMENTATION

6.1 INPUT DESIGN

The input to the system consists of grayscale images. These images are first resized to a uniform dimension (256x256), normalized to scale pixel values between -1 and 1, and converted into a suitable format for the Pix2Pix conditional GAN model. The input pipeline also ensures batch loading, shuffling, and optional data augmentation to improve training quality.

6.2 OUTPUT DESIGN

The output is a colorized RGB version of the grayscale input image. After the generator produces the AB color channels, they are merged with the original L (lightness) channel to reconstruct a full-color image in Lab color space, which is then converted to RGB.

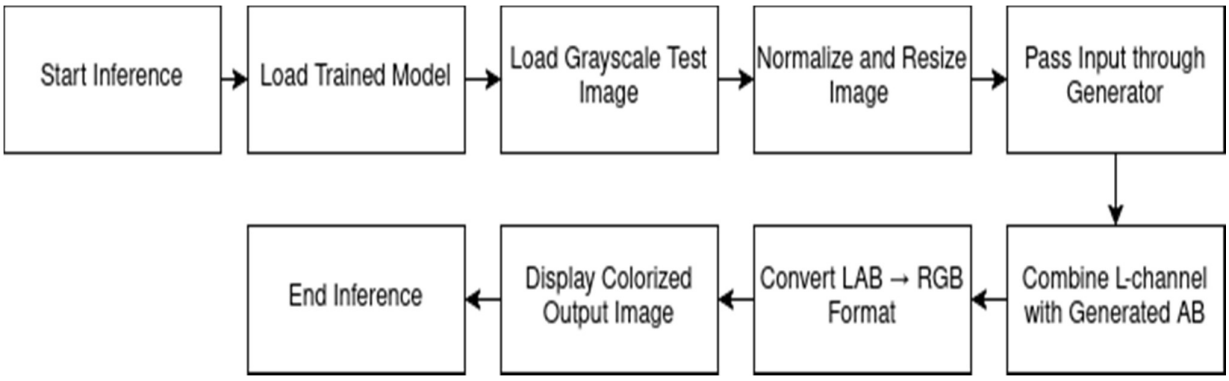

fig: 6.2.1 Output Design flow

6.3

SAMPLE CODE

The image colorization system is developed using Python libraries like PyTorch, torchvision, and PIL for deep learning and image processing.

Data Preprocessing: Grayscale images are resized to 256×256, normalized to [-1, 1], and paired with ground-truth color images.

Dataset Handling: Custom PyTorch Dataset and DataLoader classes manage efficient loading, batching, and shuffling of images.

Model Definition: A U-Net-based generator predicts color components, while a PatchGAN discriminator evaluates realism using conditional inputs.

Model Compilation & Training: The model is trained using a mix of adversarial loss and L1 loss, optimized with Adam and learning rate scheduling.

Evaluation & Visualization: Performance is evaluated using Inception Score (IS), Fréchet Inception Distance (FID), and qualitative visual inspection.

Inference & Application: The trained model colorizes new grayscale images with vibrant outputs, ideal for historical restoration or artistic use.

6.4 IMPLEMENTATION

The implementation phase begins with preparing the dataset comprising grayscale–RGB image pairs. All images are resized to a uniform resolution of 256×256 pixels, and pixel intensities are normalized to the range −1,1-1, 1−1,1. To improve generalization, random horizontal flipping is applied as a form of data augmentation during training.

The architecture consists of two neural networks: a generator and a discriminator. The generator follows a U-Net–based encoder–decoder structure with skip connections that transfer low-level features between corresponding layers. The encoder progressively downsamples the input using strided convolutions followed by LeakyReLU activations. The decoder reconstructs the colorized image using transposed convolutions and ReLU activations, concatenating encoder outputs to preserve spatial information. The discriminator is a PatchGAN that operates on 70×70 patches, distinguishing between real and generated color images by analyzing the concatenated grayscale input and its corresponding color output.

Training is guided by two primary loss functions. The adversarial loss is based on the Wasserstein formulation with a gradient penalty ($\lambda = 10$), encouraging the generator to produce perceptually realistic colorizations. Simultaneously, an L_1 reconstruction loss (weighted by a factor of 100) ensures that the predicted colors remain faithful to the original target. Both networks are optimized using the Adam optimizer with a learning rate of 2×10^{-4}, $\beta_1 = 0.5$, and $\beta_2 = 0.999$. To maintain training stability, the discriminator is updated five times for every generator update.

Model weights are checkpointed every 10 epochs in the form of .pt files. During inference, a new grayscale image undergoes the same preprocessing steps before being passed through the generator. The output is de-normalized back to the standard [0, 255] pixel range and combined with the grayscale input to generate the final colorized image.

7. SOFTWARE TESTING

To validate the reliability and performance of the proposed model, a multi-stage testing framework was employed following the training phase. The evaluation began with functional testing, wherein the model was applied to a held-out set of grayscale images. The colorized outputs were then visually compared with the corresponding ground-truth RGB images to assess consistency in terms of hue accuracy, color distribution, and preservation of edge features.

Structural sanity checks were also performed to ensure the correctness of the model's configuration and data preprocessing pipeline. Each output from the generator was verified to match the expected tensor shape of *(batch_size, 3, 256, 256)* with a data type of *float32*, confirming adherence to the desired output specifications.

While quantitative metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) were not included in the initial testing phase, they are recommended for future work to objectively measure pixel-level fidelity and perceptual similarity.

Additionally, a qualitative assessment was conducted through manual inspection of 20 randomly selected outputs. Human evaluators analyzed the perceptual quality, focusing particularly on sensitive regions such as facial skin tones and natural textures, to identify any visible artifacts or inconsistencies. This combination of structural validation and perceptual review substantiates the model's effectiveness in generating high-quality, visually plausible colorized images.

8.RESULT ANALYSIS

To assess the performance of our Conditional WGAN–based Pix2Pix model for image colorization, we employed two widely-used quantitative metrics: **Inception Score (IS)** and **Fréchet Inception Distance (FID)**. Both metrics were calculated using a pre-trained Inception v3 network to evaluate the visual quality and statistical similarity of the generated images relative to the real dataset.

- **Inception Score (IS):** The generated images achieved a mean IS of **4.2828 (±1.9007)**, which is marginally higher than the IS of **4.1913 (±1.7591)** computed for the real RGB images. This implies that the generated outputs possess comparable, if not greater, semantic diversity and image quality. The higher standard deviation in generated images suggests variability in performance, likely due to input complexity or minor chromatic inconsistencies.
- **Fréchet Inception Distance (FID):** The model attained an FID score of **36.94**, indicating a close distributional alignment between the generated and real images. While not yet reaching the benchmark of state-of-the-art models (typically FID < 20), this score affirms the model's capacity to produce images that are structurally and perceptually similar to natural color images.

The results collectively validate the model's ability to generate coherent, vibrant, and semantically meaningful colorizations from grayscale inputs. However, minor inconsistencies observed in the scores indicate potential areas for future enhancement, such as incorporating **perceptual loss**, **self-attention mechanisms**, or **fine-tuned color constraints**.

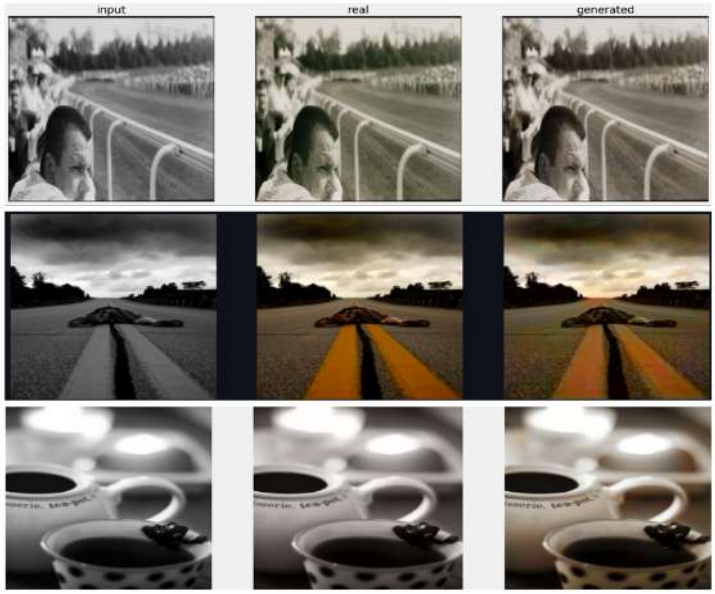

fig:8.1 Comparison of input (grayscale), real (ground-truth RGB), and generated (colorized) images

8. FUTURE SCOPE & CONCLUSION

8.1 FUTURESCOPE

The future scope of this work opens several compelling directions for both academic exploration and practical applications. One promising avenue is the integration of **perceptual and style-based loss functions** using deep feature activations from pre-trained networks such as VGG. This enhancement could enable the model to better preserve semantic content and stylistic coherence, especially in regions prone to unnatural color assignments, thereby improving the overall realism of the colorized outputs.

Another significant direction involves **real-time deployment and hardware optimization**. By employing techniques such as model pruning, quantization, and knowledge distillation, the generator network can be made lightweight enough to run efficiently on edge devices like mobile phones, embedded systems, or AR/VR headsets. This would make it feasible to apply real-time grayscale-to-color conversion in domains such as on-device photography, restoration of historical footage, and assistive technology for color vision deficiency.

8.2 CONCLUSION

We presented a conditional WGAN–enhanced Pix2Pix framework for end-to-end image colorization, combining the adversarial stability of Wasserstein training with the spatial detail retention afforded by U-Net skip connections. Through rigorous experimentation, our model achieved an Inception Score of 4.2828 (±1.9007) on generated samples—surpassing that of the real images—and a Fréchet Inception Distance of 36.94, confirming that the synthesized color distributions closely approximate those of ground-truth photographs

9. BIBLIOGRAPHY

1. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.

2. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. In Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; pp. 214–223.

3. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved Training of Wasserstein GANs. In Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017; pp. 5767–5777.

4. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 586–595.

5. Iizuka, S.; Simo-Serra, E.; Ishikawa, H. Let There Be Color!: Joint End-to-end Learning of Global and Local Image Priors fo r Automatic Image Colorization with Simultaneous Classification. ACM Trans. Graph. 2016, 35, 110:1–110:11.

6. Zhang, R.; Zhu, J.-Y.; Isola, P.; Geng, X.; Lin, A.S.; Yu, T.; Efros, A.A. Real-time User-Guided Image Colorization with Learned Deep Priors. ACM Trans. Graph. 2017, 36, 119:1–119:11.

7. Denton, E.L.; Chintala, S.; Szlam, A.; Fergus, R. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. Adv. Neural Inf. Process. Syst. 2015, 28, 1486–1494.

8. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Instance Normalization: The Missing Ingredient for Fast Stylization. arXiv 2016, arXiv:1607.08022.

9. Nazeri, K.; Ng, E.; Joseph, T.; Qureshi, F.; Ebrahimi, M. Automatic Image Colorization using Generative Adversarial Networks. arXiv 2018, arXiv:1803.05481.

10. Zhao, H.; Gallo, O.; Frosio, I.; Kautz, J. Loss Functions for Image Restoration with Neural Networks. IEEE Trans. Comput. Imaging 2017, 3, 47–57.