Image Recoloring with conditional GANs

Mahmoud Z. A. Wahba†, Emilija Dokanovic†, Naoures Atidel Hamrouni †

***Abstract*—Image recolorization is a challenging and relevant task in image translation. In this paper, we propose and compare three models based on the Pix2Pix framework for this task. The first model adopts the original Pix2Pix method, while the second model improves the generator by adding the perceptual loss (LPIPS) to the loss function. The third model modifies the Pix2Pix structure and employs the WGAN loss. Our contribution is to advance image recolorization task and to evaluate the impact of different loss functions on the quality and diversity of the generated images. We conduct extensive experiments and use various metrics such as FID, IS, PSNR, MAE, SSIM, and human evaluation to assess the performance of our models. Our results show that the model with the perceptual loss produces images with high quality and diversity, as reflected by the IS and FID scores. However, Pix2Pix achieves slightly better pixel-level accuracy and similarity to the ground truth images. The WGAN model does not generate realistic images as other two models. Our findings provide useful guidance for researchers and practitioners in image recolorization and image translation. Future work can further explore other loss functions, architectural variations, and dataset compositions to enhance image generation tasks.**

***Index Terms*—Unsupervised Learning, Generative Adversarial Network, Convolutional Neural Networks, PatchGAN, Wasserstein GAN, and U-Net**

I. Introduction

The process of image recolorization entails transforming grayscale images, which lack color information, into RGB images that exhibit vibrant and natural colors. The automatic generation of accurate and visually pleasing colorizations has diverse practical applications, including restoring old photographs, enhancing visual content, creating artistic renderings, enabling accessibility for the visually impaired, facilitating product design and prototyping, and supporting educational and historical documentation. Image recolorization can be seen as a specific type of image translation domain. Image translation encompasses a broader range of tasks, including style transfer, domain adaptation, and other forms of visual transformation. By understanding image recolorization within the context of image translation, techniques, and methodologies developed for this matter can be done to enhance the performance and efficiency of image recolorization models. Moreover, convolutional neural networks (CNNs) have emerged as a powerful tool in computer vision, revolutionizing various image-related tasks such as image classification, object detection, image segmentation, and image generation. CNNs excel at capturing and understanding spatial patterns and features within images, making them an ideal choice for tasks such as image recolorization. One of the advantages of CNNs is the ability to preserve the inherent structure of pictures. Unlike fully connected networks that may disrupt this structure and loss important features and local coherence between neighboring pixels, the convolutional neural network is designed to retain these crucial aspects. By leveraging the unique features of CNNs, meaningful patterns, and representations can be extracted from images, enabling more accurate and efficient analysis and interpretation of visual data [1]. Therefore, the main goal of supervised learning is to train a model to make predictions based on labeled training data. While the training dataset consists of input features (X) and corresponding target labels (Y). So, the objective is to minimize the discrepancy between the predicted outputs and the true labels [2]. Common loss functions used in supervised learning include Mean Squared Error (MSE) and Binary Cross-Entropy (BCE). In addition, in unsupervised learning, explicit target labels are not available for the training data. We can say that the objective is to extract meaningful patterns, relationships, or structures from the unlabeled data. Loss functions used in unsupervised learning differ in their purpose and characteristics. Some common examples include Autoencoder Loss [3], Variational Autoencoder (VAE) Loss [4], and Adversarial Loss.

Adversarial learning, as seen in Generative Adversarial Networks (GANs) [5], consists of generator and a discriminator network: the generator aims to produce realistic outputs, while the discriminator distinguishes between real and generated data [6] as refferring to figure 1. Here, the loss function in GANs combines the generator's loss (to deceive the discriminator) and the discriminator's loss (to classify real and fake data in a correct way). The discriminator's loss function incorporates the loss function of the generator. The output classification of the discriminator is also considered in the loss functions [7].

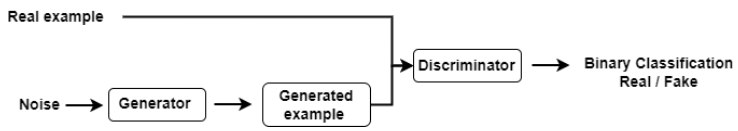

Fig. 1: Generative Adversarial Network (GAN)

In this project, we employ a conditional GAN (cGAN) [8], which is a variant of the GAN that integrates additional conditional information during both the training and generation processes. The cGAN architecture includes both a generator and a discriminator network, and they receive extra conditioning input in addition to the random noise input. This conditioning input can take various forms, such as labels,

†Department of Information Engineering, University of Padova, email: {mahmoudza.wahba}@unipd.it

classes, or even other images, and it serves as a guiding factor for the generation process. By conditioning the model on specific information, the cGAN can generate outputs that are tailored and conditioned on the given information, enabling more controlled and targeted generation.
During training, both the generator and discriminator networks are fed with the conditioning input, resulting in a two-player minimax game similar to a standard GAN. The generator aims to produce outputs that match the conditioning information, while the discriminator strives to distinguish between real and generated data. This adversarial training framework allows the cGAN to learn and generate colorized images that are visually plausible and realistic. The utilization of conditional information in cGANs makes them well-suited for a wide range of tasks, including image-to-image translation, text-to-image synthesis, and style transfer. By conditioning the model on specific attributes or labels, cGANs can generate outputs that align with the desired conditions, facilitating the automatic addition of color information to images. The primary objective of this project is to leverage the power of cGANs to automatically infuse color information into images, resulting in visually convincing and realistic colorizations.

II. Related Work

Numerous studies have been conducted in the field of image recolorization showcasing significant contributions. Some notable works include: In the pioneering work of "Colorization using optimization," an algorithm was introduced that makes coloration to grayscale images automatically. In this study, researchers proposed an approach that combines user-provided color hints with image statistics to generate plausible colorizations [9].

The "Colorful Image Colorization" paper presented a deep learning approach for automatic image colorization [10]. By training a Convolutional Neural Network (CNN) on a large dataset of colored images, the authors enabled the prediction of chrominance channels (a and b) based on the luminance channel (L) of an image, effectively establishing a mapping between grayscale and color images [10].

Moreover, to enhance the deep learning-based image colorization approach, the work on "Deep Polarization: Image Colorization using CNNs and Inception-ResNet-v2" combined CNNs with the Inception-ResNet-v2 architecture, was introduced. This concept generates high-quality and realistic colorizations. This approach was achieved through an encoder-decoder model that integrated a deep CNN trained from scratch with high-level features extracted from the pre-trained Inception-ResNet-v2 model [11].

While not only focused on image recolorization, the "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks" introduced the CycleGAN framework, which was applied to tasks such as recolorization. By training on unpaired datasets of grayscale and color images, CycleGAN learned mappings between the two domains without the need for paired training examples. Consequently, the model demonstrated the capability to perform image recolorization as a form of image-to-image translation [4].

Therefore, in the field of image translation using conditional generative adversarial networks (GANs), the "Image-to-Image Translation with Conditional Adversarial Networks," also known as the Pix2Pix paper, proposed a framework for learning a mapping between input and output images using paired training data. Through a conditional GAN architecture, where a generator network and a discriminator network were trained simultaneously. The generator aimed to produce a corresponding output image and the discriminator assessed its authenticity by distinguishing it from real output images. The two networks engaged in an adversarial training process, with the generator striving to deceive the discriminator and the discriminator aiming to accurately classify real and generated images [12]. We say that these works have made significant contributions to the field of image colorization and image translation, demonstrating the effectiveness of various approaches, such as optimization, deep learning, and GANs, in achieving realistic and visually appealing results.

Image-to-Image Translation with cGANs has a wide range of applications, including style transfer, colorization, super-resolution, and domain adaptation. They offer the ability to generate high-quality images with desired modifications, making them a valuable tool in areas such as computer vision and graphics.
In the context of various works conducted in the field of image recolorization and image translation tasks, in this project, we implement our work in 3 scenarios, the first scenario is to implement an image recoloring task using pix2pix architecture by using the losses that they used: binary cross-entropy, L1 loss with lambda as a regularization term. The second scenario uses perceptual loss with other losses to enhance model behavior. The third scenario uses Wasserstein Loss implementation. all scenarios using the same architecture for the generator and discriminator; U-Net structure for the generator and Patch-GAN for the discriminator with small differences in some parts we will mention later in this report.

III. Processing Pipeline

The processing pipeline for a Conditional Generative Adversarial Network (CGAN) for image recoloring is A multi-steps process, as shown in figure 2. The process includes data collection, data preprocessing, learning framework, and model evaluation. Deep learning models require big data sets in order to build more reliable data sets. Collecting this massive data is an exhausting task, therefore, we used an existing data set Mini-ImageNet that is subset of ImageNet 1000 which contains over 1 000 000 images that belong to 1000 classes. Before starting the training phase, collected data need to be pre-processed. This step includes converting images to greyscale, resizing and normalization. Also, this step includes dividing data set into training, validation and test sets. After proper prepocessing, next step includes building a framework for learning and training data. Learning framework for CGANs contain two components, generator and discriminator

with an objective to generate images as similar as possible to real ones. Last step includes the evaluation of the results by using metrics defined in the beginning of project.

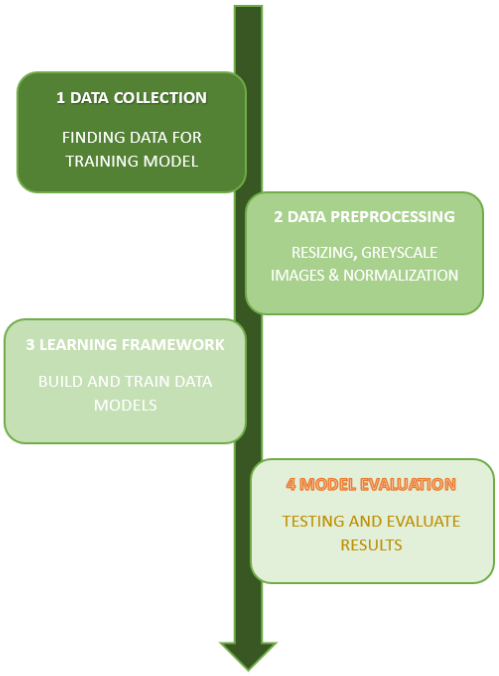

Fig. 2: Processing pipeline

IV. Signals and Features

For project purposes, we used a subset of well-known ImageNet 1000 data sets which is a large dataset of annotated photographs intended for computer vision research. This dataset spans 1000 object classes and contains over 1,200,000 training images, 50,000 validation images and 100,000 test images. Choosen subset is called Mini-ImageNet data set which contains 34745 training and 3923 test images. The reason for chosing Mini-ImageNet dataset is due to its extensive variety of pictures, diverse scenes, objects, and images with varying levels of brightness. This dataset offers a wide range of visual content, enabling us to enhance the diversity and comprehensiveness of our project. Before proceeding with the training phase, the images underwent a pre-processing stage. The initial step involved converting original images to greyscale since the objective was to develop framework to recolor them. All images were resized to the 256x256 and converted to tensors. Additionally, normalization was applied with parameters of 0.5 and 0.5. These parameters are often used to transform the pixel values of an image to a standardized range, making it more suitable for subsequent stages of the project.

As mentioned, above Mini-ImageNet has images with plenty of features and different brightness, therefore we choose to use it in order to maximize the reliability of our model. We worked with two subsets of Mini-ImageNet, first one cointains around 7000 images, from which 5000 for training, 900 for validation and 1000 for testing; and second one has 27 000 images, from which 20 000 for training, 3500 for validation and 3500 for testing. By carefully choosing this subset, we aimed to ensure an effective training process while achieving a robust evaluation framework.

V. Framework

Recoloring images presents a challenge because multiple colored images can generate the same greyscale image. The objective of this project is to develop a generative framework capable of producing a colorized image with a high probability. For making a good framework we started with the pioneering work of Isoala et al. [12] and other popular research papers in the field. These works served as crucial starting points and provided valuable insights for the development of our project which contains three proposed models.

A. Netwrok Architecture

One common characteristic shared by all three models is that they represent cGAN models, which consists of both a *Discriminator* and a *Generator* component. Hence, the initial focus of this project will be on presenting a general framework that is shared by all three models. In the subsequent subsections, we will delve into the specific details and distinctions of each model, highlighting their unique characteristics and contributions.

1) ***U-Net Generator*** *:* The generator network has a U-Net structure [13], which is composed of two primary components: a downsampling path and an upsampling path, working together to achieve pixel-wise segmentation of the input image. The downsampling path utilizes convolutional and max-pooling layers to decrease the spatial dimensions of the image. On the other hand, the upsampling path employs transposed convolutional layers to restore the spatial dimensions back to their original size as shown in figure 3. These two paths are connected by skip connections, which aid in preserving fine details and enhancing the segmentation results [12].

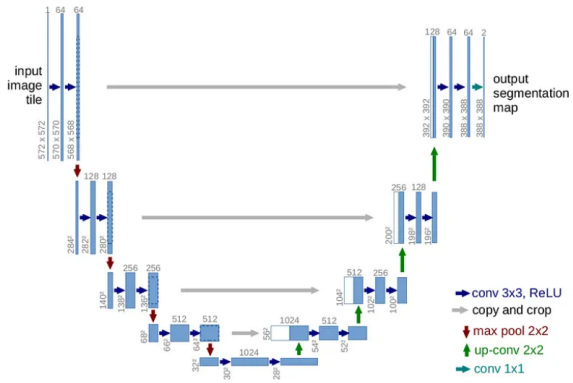

Fig. 3: U-Net architecture

The generator network is represented as an encoder-decoder structure, with a series of convolutional layers and skip connections. The encoder part of the network includes a series of residual blocks. These blocks consist of convolutional layers with Batch normalization and ReLU activation, which help the network learn the residual mapping between the input and output. By combining the encoder and decoder with residual blocks and skip connections, the network can effectively learn the mapping between the input and output, capturing both low-level and high-level features. This structure improves the network's performance and facilitates the optimization process during training.

The encoder of the network incorporates downsampling layers that reduce the size of the feature maps while increasing the number of filters. This enables the network to learn higher-level representations of the input image. Downsampling is achieved using convolutional layers with a stride of 2, which decreases spatial dimensions and increases the number of filters. Dropout layers are added after each encoding layer and the bridge layer to prevent overfitting by randomly dropping a percentage of neurons during training.

The decoder of the network consists of upsampling layers that increase the spatial dimensions and reduce the number of filters in the feature maps. These upsampling layers are accompanied by skip connections that concatenate the encoder's feature maps with the decoder's feature maps. This fusion of low-level and high-level features is crucial for image colorization. Upsampling is achieved using bilinear upsampling followed by concatenation with the corresponding encoder feature maps.

2) ***PatchGAN Discriminator*** *:* The goal of the PatchGAN Discriminator is similar to other GAN discriminators and it is to distinguish between real samples from the dataset and fake samples generated by the generator. However, its architecture differs slightly, particularly in terms of how the output layer handles regression. The term "PatchGAN" comes from the matrix of values that the discriminator generates for a given input. Unlike traditional GAN models that employ a CNN with a single output to classify entire images, the model employs a carefully designed PatchGAN to classify patches of an input image as real or fake, rather than considering the entire image as a whole [12].

The discriminator network follows the standard Convolution-BatchNormalization-ReLU blocks commonly used in deep convolutional neural networks. Three different types of convolutional blocks are defined, first one creates a convolutional layer with a LeakyReLU activation function, without Batch Normalization; second creates a convolutional layer with Batch Normalization and a LeakyReLU activation function and last one creates a convolutional layer without Batch Normalization as the last layer. The network outputs a single feature map that provides predictions of real or fake, which can be averaged to obtain a single score (loss value). The output is passed through a sigmoid activation function to output values between 0 and 1, representing the discriminator's confidence in the input being real or fake/generated.

The PatchGAN architecture offers several advantages by providing feedback on each local region or patch of the image:

- By classifying each patch individually, the PatchGAN discriminator can more effectively distinguish between real and fake images enabling better discrimination and enhances the overall classification accuracy.
- The PatchGAN's ability to evaluate local patches allows the generator to generate more realistic images and "fools" it. By receiving feedback on specific regions of the image, the generator can focus on improving the quality and details of these patches, leading to visually convincing outputs.

3) ***Optimizers*** *:* In the context of machine learning and deep learning, optimizers are algorithms or methods used to adjust the parameters of a model to minimize or maximize a certain objective function. The objective function is typically a measure of how well the model performs on a task, such as minimizing the error between the predicted output and the actual output. Optimizers play a crucial role in training neural networks, as they determine how the model's parameters are updated during the learning process. Many popular optimizers are used in deep learning for example Stochastic Gradient Descent (SGD), RMSprop, and Adam [1] [14].

In our project, we utilized two optimizers: one for the generator and another for the discriminator. In the first and second scenarios (pix2pix implementation and perceptual loss implementation), we used Adam optimizers for both the generator and discriminator. The learning rate was set to 2e-4, and the beta coefficients were (0.5, 0.999) respectively, as suggested in the pix2pix paper [12]. In the third scenario, we used RMSprop with a learning rate of 5e-5.

4) ***Objective Function*** *:* The objective of cGAN can be expressed as this equation 1:

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

The objective is formulated as a minimax game, where G seeks to minimize the loss against an adversarial D that aims to maximize it.

B. First Model - Pix2Pix

In the first implementation of the general framework, we implemented a discriminator and generator as authors [12] did. The goal of the generator is to generate as good an image as possible and "trick" the discriminator, while the objective of the discriminator is to be able to distinguish a fake image from the real one. Therefore, we could say that we have two optimization problems, *minimization* for the generator and *maximization* for the discriminator.

1) ***Generator****:* In the first case scenario)Eq.3.), for the generator we used Binary Cross Entropy(BCE) and L1 loss with *lambda* as regularization term. *BCE* measures the dissimilarity between predicted and target probability distributions. This loss function penalizes large differences between the

predicted and true labels, assigning higher loss values when predictions deviate from the true labels. *L1 loss* calculates the absolute difference between the predicted value and the true value (Eq.2.), and then takes the sum of all these differences. The purpose of L1 loss is to measure the dissimilarity between the predicted and true values. In this case, L1 loss represents the sum of all the absolute pixel differences between the generator output and ground-truth image multiple with the value of regularization term *lambda* equal 100.

$$L1_{\text{Loss}} = \sum_{i=1}^{n} |\text{Generated Output} - \text{Real Output}| \quad (2)$$

$$G_{\text{loss}} = \text{BCEloss}(\text{real labels}) + \lambda \cdot L1\text{loss} \quad (3)$$

*2) **Discriminator**:* Discriminator loss shown on Eq.4 is calculated as a sum of *BCE* for the discriminator's output for real images and *BCE* for the discriminator's output for generated images. In order to minimize to rate at which the discriminator learns relative to the generator, the authors [12] divided the loss by 2 at the time of the optimization of the discriminator.

$$D_{\text{loss}} = \frac{\text{BCE}_{\text{loss}}(\text{real images}) + \text{BCE}_{\text{loss}}(\text{real images})}{2} \quad (4)$$

C. Second model - Perceptual

In the second case, we used Perceptual loss with other losses to enhance the models'behavior. Perceptual loss is a type of loss function used in image processing tasks. Unlike pixel-wise loss functions such as Mean Squared Error (MSE) or Mean Absolute Error (MAE), Perceptual loss focuses on capturing the higher-level semantics and structure of the images. The perceptual loss encourages the generated image to match the target image in terms of high-level features, such as texture, shape, and object composition. By optimizing the perceptual loss, the model learns to produce images that resemble the target images in terms of their overall visual appearance, rather than just focusing on pixel-level accuracy. In order to make a better model, we used Learned Perceptual Image Patch Similarity *LPIPS* represented in Eq.5.

$$G_{\text{loss}} = \text{LPIPS}_{\text{loss}}(\text{Generated Output}, \text{Real Output}) + \text{BCE}_{\text{loss}}(\text{real labels}) + \lambda \cdot L1_{\text{loss}} \quad (5)$$

D. Third model - WGAN

The third scenario uses Wasserstein Loss *WL* implementation, known as WGAN [15]. In this model, the role of the discriminator is replaced to the critic model which gives a score on how real or fake is given image, instead of prediction the probability of generated images as being real or fake. WGAN uses Wasserstein distance, also known as Earth-Mover's distance (EMD), a mathematical measure of the dissimilarity between two probability distributions. Wasserstein distance is often used as a metric to evaluate the quality and convergence of the generated samples. Eq.6 shows critic loss which is calculated as the difference between the average score for real and the average score for fake images multiplied with *lambda* regularization parameter equals 10 and *gradient penalty*. Also, generator loss is slightly different in this scenario. It is represented in this equation (Eq.7.) *lambda reconstruction* regularization parameter equal to 100. The WGAN training process involves alternating updates between the generator and the critic. The critic aims to approximate the Wasserstein distance between the real and generated data distributions, providing gradient signals to guide the generator's learning. The more iterations the critic undergoes, the better it becomes at distinguishing between real and generated samples. In our project we chose 3 iterations for critic training as a hyperparameter.

$$C_{\text{loss}} = (\mu_{\text{score real images}} - \mu_{\text{score fake images}}) \cdot \lambda_{\text{gp}} \cdot \nabla_{\text{penalty}} \quad (6)$$

$$G_{\text{loss}} = -\mu_{\text{score fake images}} + \lambda_{\text{reconstruction}} \cdot R_{\text{loss}} \quad (7)$$

$$R_{\text{loss}} = \frac{1}{n} \sum_{i=1}^{n} |\text{Generated Output} - \text{Real Output}| \quad (8)$$

VI. Results

A. Metrics for cGAN evaluation

Evaluating the quality of synthesized images represents a challenging task, and traditional metrics like per-pixel mean-squared error are often insufficient. Image quality evaluation is quite subjective and depends on human perception, assessing the effectiveness of structured losses requires more sophisticated evaluation methods. Therefore, we considered the following metrics:

- **Inception Score(IS)**
  The Inception Score is a commonly used metric for evaluating the quality of generated images by GANs. IS measures the diversity and quality of the generated images based on the output of an Inception model .In this project was calculated IS value for real and generated images. Similar IS values for these groups indicates preferred results.
- **Fretchet Inception Distance (FID)**
  FID is a more principled and comprehensive metric and has been shown to be more consistent with human evaluation in assessing the realism and variation of the generated samples.Lower FID means better quality and diversity because the distributions of the feature representations of the generated images and real images are closer. FID compares the statistics of feature representations extracted from the generated images and real images using an Inception model. In comparism with IS, FID is more robust metric.
- **Peak Signal-to-Noise Ratio (PSNR)** PSNR measures the difference between the original and reconstructed images based on the mean squared error (MSE) between

corresponding pixels. The PSNR value is usually expressed in decibels (dB). A higher PSNR indicates better quality, as it implies a smaller difference between the original and generated images. However, PSNR primarily focuses on pixel-level differences and does not always correlate well with perceptual image quality

- **Structural Similarity Index (SSIM)** SSIM takes into account structural information and perceptual aspects of image quality.The SSIM index is calculated by comparing three key components: luminance, contrast, and structure. It measures the similarity between the original and distorted images based on these components. The SSIM index ranges between -1 and 1, where 1 indicates a perfect similarity.
- **Mean Absolute Error (MAE)** MAE is a metric commonly used to measure the average magnitude of errors between predicted and actual values. It provides a measure of the average absolute difference between the predicted and true values, regardless of the direction of the errors.

TABLE 1: Evaluating model

	7000 images			27000 images		
	Pix2Pix	Perceptual	WGAN	Pix2Pix	Perceptual	WGAN
FID	89.98	**85.92**	109.80	76.37	**74.02**	172.74
IS_real	12.05	12.05	11.81	14.47	14.47	14.47
IS_generated	10.88	**10.99**	10.32	13.15	**13.41**	11.62
PSNR	**16.47**	16.37	15.07	**16.44**	16.31	10.78
SSIM	**0.72**	0.70	0.66	**0.72**	0.71	0.43
MAE	**0.12**	**0.12**	0.14	**0.12**	**0.12**	0.22

B. Objective evaluation

From evaluation metrics obtained using test dataset, it can be inferred that the perceptual loss model performs better compared to other two, Pix2Pix and W-GAN. The perceptual loss model achieved the best scores for IS and FID which shows its ability to generate diverse and high-quality images compared to the other considered models.

However, it is worth mentioning that Pix2Pix showed slightly higher values for PSNR, SSIM, and MAE compared to the perceptual loss model. While the differences were small, this suggests that Pix2Pix performed marginally better in terms of pixel-level accuracy and similarity to the ground truth images.

On the other hand, the W-GAN model demonstrated inferior performance across all the evaluated metrics, indicating its limitations in generating high-quality and visually faithful images compared to the other models.

C. Subjective evaluation

As mentioned, evauliating quality of images it is not easy task, therefore, in evaulation network's performance we included qualitative comparisons into account, between input greyscale and predicted colored image.

Fig. 4,5,6 represents recolored image paired with geryscale. Comparing three models, we found out that again perceptual yielded the best results in terms of image quality. However,

Fig. 4: Pix2Pix

Fig. 5: Perceptual

Fig. 6: WGAN

the results obtained from the Pix2Pix model were also very good, although not as exceptional as the perceptual loss model. These findings were consistent for both dataset sizes. In comparison, the W-GAN model did not perform as well as the other two models in terms of generating visually appealing and realistic images for both datasets.

However, the proposed framework generates high-quality colorized images that closely resemble the real images, showing its potential for the image colorization task.

VII. Conclusions

In this study, we explored image recolorization using three models based on the Pix2Pix architecture: Pix2Pix, perceptual, and W-GAN. Through extensive evaluations and analysis, including FID, IS, PSNR, MAE, SSIM, and subjective evaluation, we obtained valuable insights and findings. The model incorporating perceptual loss shows the best results, achieving very good image quality and diversity compared to the other models. However, the Pix2Pix model demonstrated notable strengths in terms of pixel-level accuracy and similarity to ground truth images. On the other hand, the W-GAN model exhibited limitations in generating high-quality images.

The success of the perceptual loss model showcases the importance of considering perceptual factors in generating visually pleasing and diverse outputs. Our results can help researchers to improve their image recolorization approaches and enhance the user experience. Further research can be pursued to enhance the models' performance. Possible improvements could be gained by exploring alternative loss functions, architectural variations, and dataset compositions. Evaluating the models across different application domains and conducting user studies would provide comprehensive insights for practical deployment.

In summary, this study advances the field of image recolorization by demonstrating the effectiveness of perceptual loss and providing valuable insights for future research. By considering these findings, researchers can continue to explore image recolorization and improve the visual quality and usability of generated outputs.

VIII. Challenges through work

Throughout this project, we faced challenges related to time limitations and computational power. Although we achieved promising results with the selected dataset, we were also considering the possibility of evaluating our models on a different dataset to assess their generalizability. However, due to the time constraints and computational limitations, we were unable to undertake this exploration. In future work, it would be beneficial to explore the performance of our models on alternative datasets, including those with distinct visual characteristics or specific application domains. This would enable a comprehensive assessment of their capabilities and further validate their effectiveness in different scenarios.

By acknowledging these limitations and future possibilities, we highlight the potential for further investigation into the generalizability of our models and encourage researchers to explore additional datasets with more computational power and extended time resources.

References

[1] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[2] B. Liu and B. Liu, *Supervised learning*. Springer, 2011.

[3] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *arXiv preprint arXiv:2003.05991*, 2020.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[6] R. Sankar, A. Nair, P. Abhinav, S. K. P. Mothukuri, and S. G. Koolagudi, "Image colorization using gans and perceptual loss," in *2020 International Conference on Artificial Intelligence and Signal Processing (AISP)*, pp. 1–4, IEEE, 2020.

[7] M. Isaksson, "Exploring generative adversarial networks (gans)." Towards Data Science, 2020. [Online]. Available: https://towardsdatascience.com/exploring-generative-adversarial-networks-gans-488e1d901d4a.

[8] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[9] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM SIGGRAPH 2004 Papers*, pp. 689–694, 2004.

[10] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pp. 649–666, Springer, 2016.

[11] F. Baldassarre, D. G. Morín, and L. Rodés-Guirao, "Deep koalarization: Image colorization using cnns and inception-resnet-v2," *arXiv preprint arXiv:1712.03400*, 2017.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241, Springer, 2015.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[15] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, pp. 214–223, PMLR, 2017.

Appendix

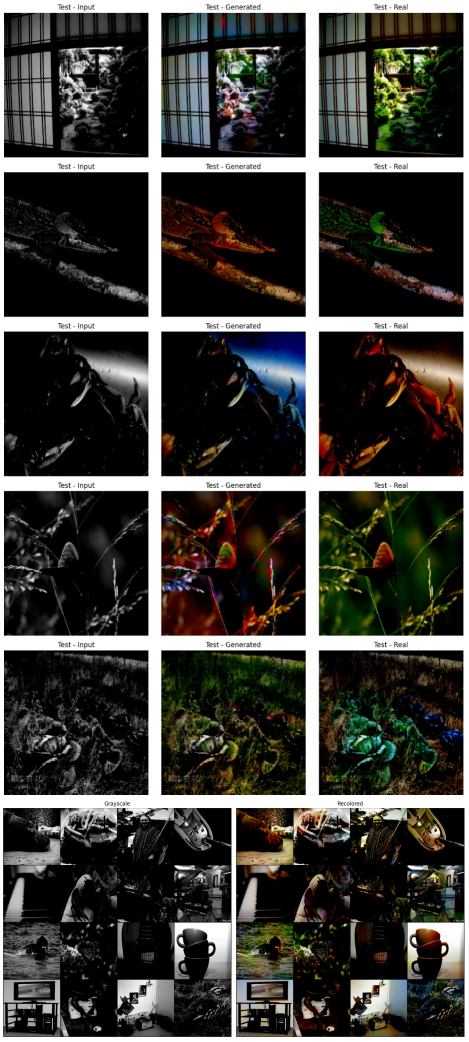

Fig. A.8: More results for the Perceptual model

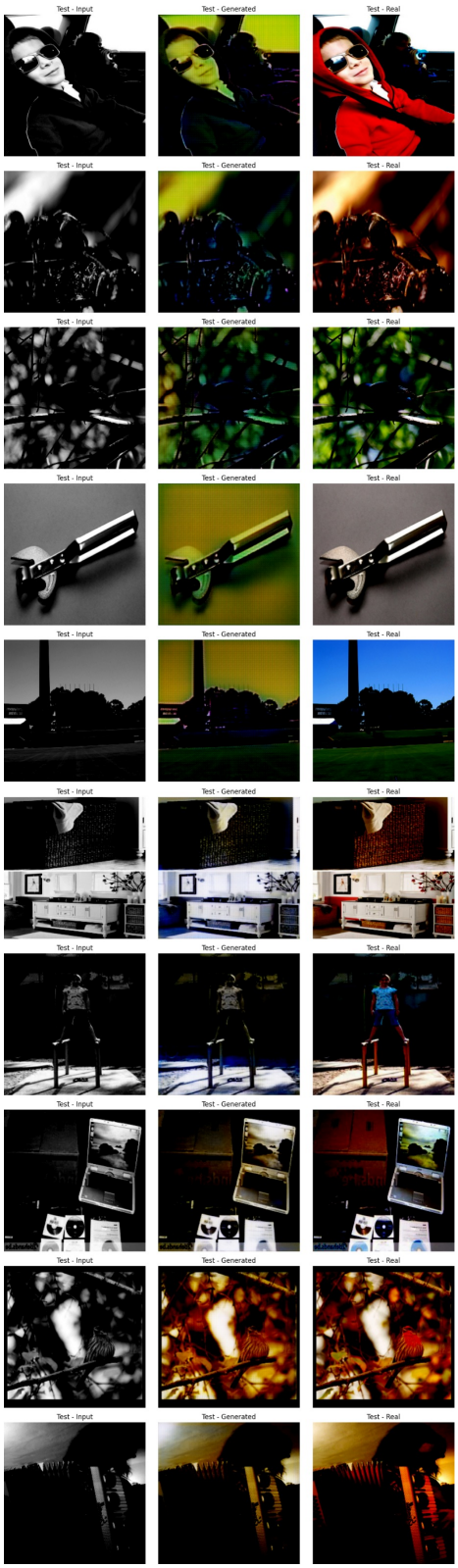

Fig. A.7: More results for the WGAN model

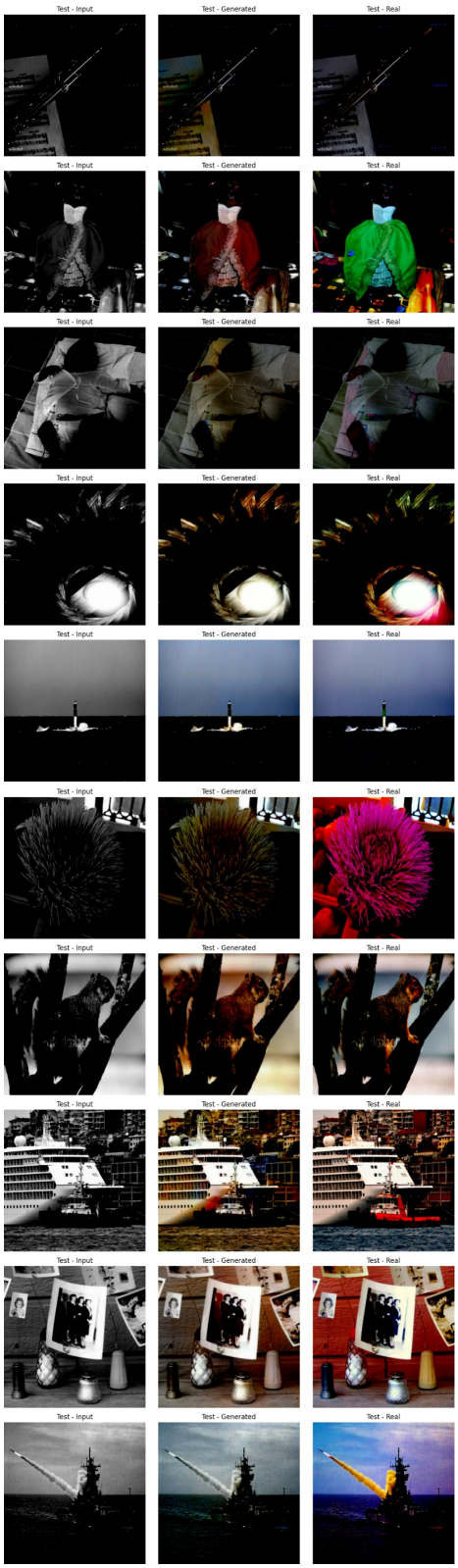

Fig. A.9: More results for pix2pix