DL4CV - Assignment Report
CMS4CV - Historical Image Recoloring and Restoration

Giovanni Carboniero, Pierluigi Mancinelli, Fabio Sacerdote,

December 6, 2024

1 Introduction

Historical black-and-white photographs are highly valued for their creative and artistic significance. However, their lack of color limits the vividness of the original scenes. The process of colorizing black-and-white photographs profoundly transforms the viewer's perspective, effectively bridging the temporal divide between past and present and enhancing the images' relevance. Nevertheless, the accurate reconstruction of original colors remains a significant challenge, as reliable references to the authentic hues of early photographs are often unavailable. This paper proposes a solution to this issue, with the aim of providing an automatic way of *re-coloring* historical images and *reconstructing* their visual quality using **deep learning** and **image reconstruction** techniques. To reach such objectives, we compare different Computer Vision architectures and strategies: a **vanilla U-Net** model, a **Fusion Model** combining a U-Net with the pre-trained segmentation model DeepLabV3+[3], and a **GAN architecture** implementing the U-Net as generator.

1.1 Goals

To be more specific, our goal is to compare the performance of these models in addressing **two simultaneous tasks**: *super-resolution* of the images to **twice their original size**, and conversion from a single-channel gray-scale image to a 3-channel **RGB** *recoloring*.

1.2 Approach

Our approach consists in artificially creating a dataset of gray-scale, lower-resolution, and noisier images, starting from a dataset of colored images. We then train the models to **reconstruct the original colored image** starting from the gray-scale one. Finally, we use the models to make **zero-shot inference** on an additional dataset of historical gray-scale images.

2 Problem Analysis

2.1 Dataset Overview and Preprocessing

To address the task, we start by carefully crafting a training dataset. Given that the subjects portrayed in historical images are people, buildings, and natural landscapes, we gathered two datasets: the *Flickr30k Dataset*, containing roughly 31k images of various subjects, and *Landscape Images*, containing 5589 pictures. Finally, we leverage the *Historical Image Similarity Dataset* for zero-shot inference on actual historical images. More about these can be found in the Appendix: Dataset Overview.
After a thorough data exploration, we design the following **preprocessing steps** to generate, for each image, a corresponding input image in gray-scale and lower quality. Our aim is to match the appearance of historical images as closely as possible by applying several transformations: we drop images with an aspect ratio (width/height) outside 0.5–1.5 to avoid distortion; we reduce image quality through **graining** (used in one experiment), **Gaussian blurring** and **resizing** to 128x128 for efficient training while maintaining reasonable sizes. Finally, we convert to **gray-scale** and we apply **CLAHE**: histogram equalization to enhance contrast without over-concentration of pixel values. Ground truth images, instead, are resized to 256x256, that is twice the input images.

2.2 Challenges

The problem of simultaneously *re-coloring* and applying *super-resolution* to gray-scale images presents several challenges. The first, is adapting model architecture to solve both tasks. A second issue to consider, is the inherently-complex prob-

lem of mapping gray-scales to a higher-dimensional colored space of an image that is twice as large: since to each shade of gray can correspond many colors, inferring local semantic information is essential to properly reconstruct the original image. To this purpose, the **U-Net** architecture is particularly suited. A final challenge is the choice of the loss function, as we detail in the following paragraph.

2.3 Literature Review

Referring to existent literature, the choice of appropriate loss function and evaluation metrics in this task is critical[9]. The existing evaluation metrics differ in terms of color ranges and formats[2][5]. However, given our case employs RGB and grayscale images, we adopt Peak-Noise-Signal-Ratio (PNSR)[6] and Structural Similarity Index Measure (SSIM)[8]. These quantify the image quality in two ways: the former compares the similarity between original and processed images using Mean Squared Error (MSE), with higher values indicating better quality. In contrast, the latter analyses structural, brightness, and contrast similarities to closely resemble human visual perception. However, neither of the two assesses colors. Some previous work proposes CIEDE2000, an advanced method for quantifying color differences that incorporates corrections for perceptual non-uniformities in lightness, chroma, and hue in the LAB color space [2]. Indeed, implementing metrics that account for color differences may result in better performances but, after some testing, we decided not to translate our predictions in LAB spaces, as the International Commission on Illumination (CIE) would suggest[2][1].
In addition to metrics and losses, architectures take inspiration from related works. Specifically, the Fusion Model architecture arises from a combination of ideas proposed by Žeger et al.[9]; whereas the idea of using segmented images as inputs comes from Isola et al.[7]. Similarly, the GAN was previously proposed by both Isola et al.[7] and Goodfellow et al.[4]. Coherently, we use different losses for each model. For the **Vanilla and Fusion U-Net** we use a linear combination of Mean Absolute Error (MAE) and SSIM loss (1 - SSIM value) as the latter is less sensitive to image degradation[5]. In the case of **GAN**, instead, we adopt an adversarial loss that includes the Mean Absolute Error (MAE) to make a reference to the input image during the generative reconstruction. However, our work slightly departs from existing literature, since it adds a super-resolution step to the more common colorization task.

3 Method

We frame the problem of *super-resolution* and *recoloring* as a **regression problem** with dense predictions. In particular, our output is found in a final convolutional block with three channels, which we interpret as Red, Green, and Blue (RGB).
For this purpose, the backbone architecture of all the models we test is a slightly modified **U-Net** that ends with an additional UpSampling and Convolutional block, thus achieving the desired output dimension (i.e. twice the input size). Finally, the output is processed through a Sigmoid, and is then compared to the original full-resolution color image (rescaled to the 0-1 interval) using appropriate loss functions defined before.
As a second approach, under the assumption that the segmentation patches could improve the coloring performances, we also build a **Fusion Model** on top of the U-Net backbone. This consists in using a pre-trained segmentation model, DeepLabV3+ (built on MobileNetV3), to enrich the input of the previous architecture with segmented images.
To conclude, we employ a **GAN model**, whose generator ($\mathcal{G}$) is the previously mentioned U-Net. In this approach, we build a discriminator ($\mathcal{D}$) with 4 convolutional blocks and a single Dense layer to distinguish generated images between "real" and "fake". Even though GANs conceptualize as mappings random noise z to an output y [4], our model starts from a grayscale image x rather than random noise. We train the model employing an **adversarial loss**: the discriminator aims at minimizing the *BinaryCrossEntropy* loss, whereas the generator tries to "fool" the discriminator through a combined loss. Specifically, the U-Net aims at maximizing the discriminator's loss while minimizing MAE, as exemplified in Pix2pix[7]. We privilege MAE over MSE as it is more consistent with respect to coloring and blurring. Specifically, we choose to weight the MAE in the GAN loss with a λ in $(100, +\infty)$ to give more importance to color and input image fidelity, all else equal.
More details about models' blocks, activations and parameters can be found in the Appendix, along with mathematical derivation of the losses.

4 Experiments

We experiment with three different models. In each case we test different losses, activations and architectures but we train them all on the same dataset: *Landscape Images*. For the first two approaches, we test the following losses: Mean Squared Error (MSE); Mean Absolute Error (MAE); linear combinations of MAE and SSIM[5]. Moreover, we adopt PSNR and SSIM as evaluation metrics for comparison, according to literature suggestions[9][5].
Furthermore, we experiment with the GAN model, trying to strike a balance between the generator and discriminator's loss, but also with the λ parameter which weights the MAE. Specifically, we try λ equal to 100, 150 and 200, finally finding a best option in the middle one. In this case we also tweak the learning rate (lr) and we end up choosing a very small 1e-5 to guarante a slow but consistent and gradual training.
In conclusion, given that historical images portray many subjects, we also experiment with a larger, more diverse, dataset of 10000 images taken both from the *Flickr30k Dataset* and the *Landscape Images*.

5 Results

Here we showcase the results achieved by our models and we compare them according to MAE, PSNR and SSIM. Despite better-than-expected results, we considered the Vanilla U-Net as the baseline model, at least from a structural perspective. The other models build on top of the baseline aiming to improve the performance.

Model	MAE	PSNR	SSIM	Epochs
U-Net	0.067	21.20	**0.78**	30
Fusion Model	0.07	20.98	0.77	30
GAN	**0.064**	**21.56**	0.77	50

Table 1: Performance out-of-sample on 20% data.

Contrarily to our initial expectation, the distance in performance between the alternatives is modest. Indeed, we believe that these metrics could not perfectly grasp the coloring differences, due to the highly non-uniform RGB space. In fact, the GAN produce a visually better colorization not evaluated by these metrics. A visual comparison between predicted images is found in the Appendix.
To conclude, GAN's performance on the larger dataset was unsatisfactory: despite the longer training, 150 epochs, it did not manage to color complex and heterogeneous subjects. Nevertheless, training the model for even more epochs may achieve this.

6 Discussion

All the models proved to be able to *recolor* landscapes and produce very plausible results, while struggling with people and other subjects. Failed recoloring examples can be found in the Appendix. Nevertheless, all models proved to be extremely powerful in the *super-resolution* task. In addition, since the U-Net architecture is fully convolutional, it allows to manipulate images of different sizes.
In the light of the results obtained in Table 1, we deem that those metrics cannot express the true ability of the models to reconstruct landscapes with high fidelity and plausibility, and do not reflect the significant difference in quality perceived by human eye. In the hope of improving model's recoloring capabilities, we also train a GAN on a larger sample that includes heterogeneous subjects. As a matter of fact, results after 150 epochs are unsatisfactory. Still, those results show hints of partial recoloring suggesting that continuing the training for a larger number of epochs, or having the computing power to further enlarge the training dataset, may achieve the desired result.

7 Conclusions

This study tackled super-resolution and re-coloring of historical images, comparing a Vanilla U-Net, a Fusion Model with DeepLabV3+, and a GAN. While the latter showed better performance, evaluating color fidelity remains challenging with existing metrics. Future work should focus on improved color-assessing metrics, training on larger datasets, and advanced architectures to enhance generalization and accuracy. Indeed, we believe including Transformers may be a valuable path to follow to find a conclusive solution to this problem.

References

[1] Colorimetry, part 6: Ciede2000 colour-difference formula.

[2] M. P. L. Alexander Toet. *A new universal colour image fidelity metric*, volume Volume 24. 2003.

[3] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.

[5] A. Horé and D. Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010.

[6] Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.

[7] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.

[8] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[9] I. Žeger, S. Grgic, J. Vuković, and G. Šišul. Grayscale image colorization methods: Overview and evaluation. *IEEE Access*, 9:113326–113346, 2021.

8 Appendix

8.1 Dataset Overview

Below, we attach samples of images contained in the three dataset involved in the study: the *Flickr30k Dataset* (37,142 images of various subjects), the *Landscape Images* dataset (5,589 images), and the *Historical Image Similarity Dataset* used to perform inference.

Figure 1: Sample from the *Flickr30k Dataset*

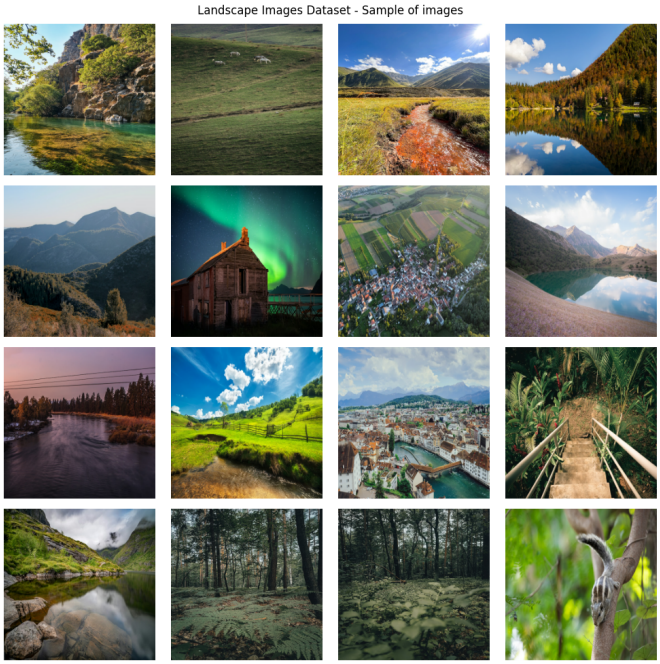

Figure 2: Sample from the *Landscape Images Dataset*

Figure 3: Sample from the *Historical Image Similarity Dataset*

8.2 Image Preprocessing

Below, we attach a sample of images and the resulting gray-scale image obtained from our preprocessing steps.

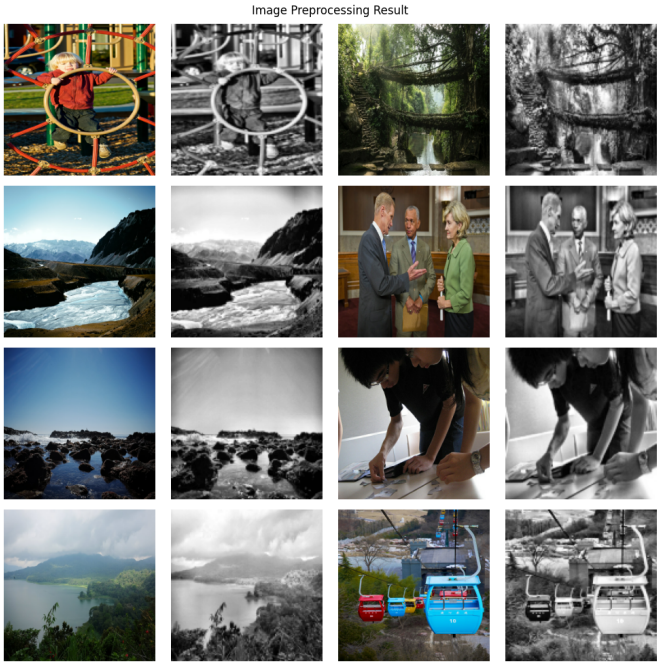

Figure 4: Sample taken from both datasets

8.3 Model Architectures and Experiments

The final architecture of our U-Net model is:

- Input Layer: 128x128 (Normalized Grayscale Image)
- 5 downsampling blocks (4 in the fusion model): Conv2D-BatchNorm-Leaky ReLU-Conv2D-BatchNorm-Leaky ReLU-MaxPooling2D
- Bootleneck with 1024 channels
- 6 upsampling blocks (5 in the fusion model) to achieve super-resolution: UpSampling2D-Conv2D-BatchNorm-LeakyReLU
- Skip connections between downsampling and upsampling blocks
- Output layer: 256x256 (RGB Image scaled between 0 and 1 through Sigmoid)

For the sake of clarity, in the following illustrations we have only drawn a single convolutional block between MaxPooling layers, despite having tried stacking between 1 and 3 of them.

8.3.1 Vanilla U-Net

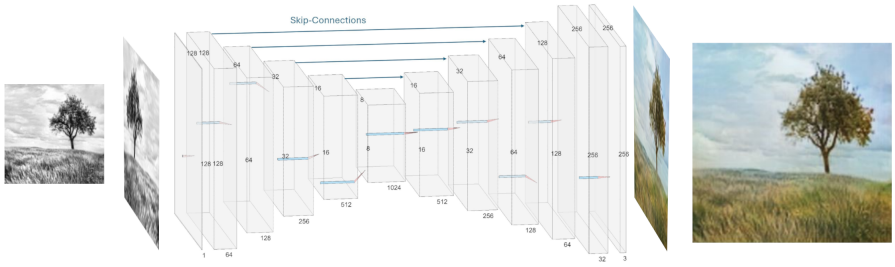

Figure 5: The architecture and functioning of our U-Net model

Tested loss functions: MSE, MAE, **combination of MAE and SSIM Loss**, combination of MAE, SSIM Loss and PSNR Loss.
Tested activation functions: ReLU, **LeakyRelu**.
Tested output activation functions: linear + clipping between 0 and 255, **Sigmoid**.
Parameters in **bold** highlight our final choices.

8.3.2 Fusion Model

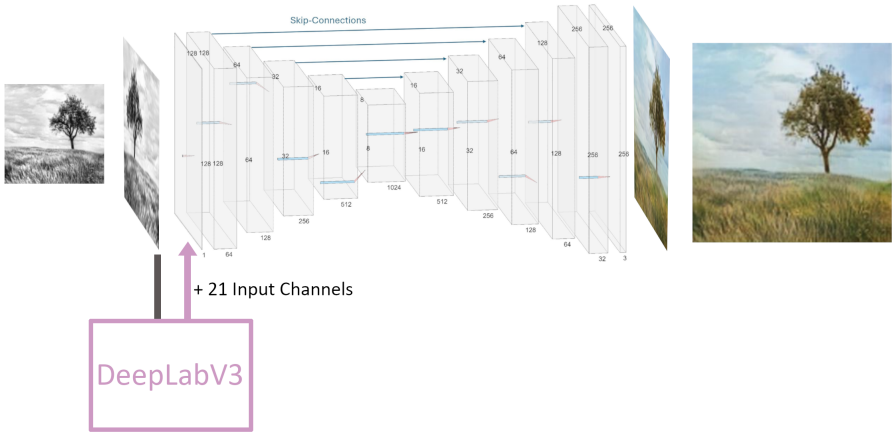

Figure 6: The architecture and functioning of our fusion model

Tested loss functions: MSE, MAE, **combination of MAE and SSIM Loss**, combination of MAE, SSIM Loss and PSNR Loss.
Tested activation functions: ReLU, **LeakyRelu**.
Tested output activation functions: **Sigmoid**.

Note: since DeepLabV3+ takes as input 3-channel RGB images, we triplicate the channels of the grayscale image before feeding it to the pretrained model, this way it is still grayscale and is accepted as input. We also visually inspect output predictions to ensure people and objects are properly segmented.

8.3.3 U-Net GAN

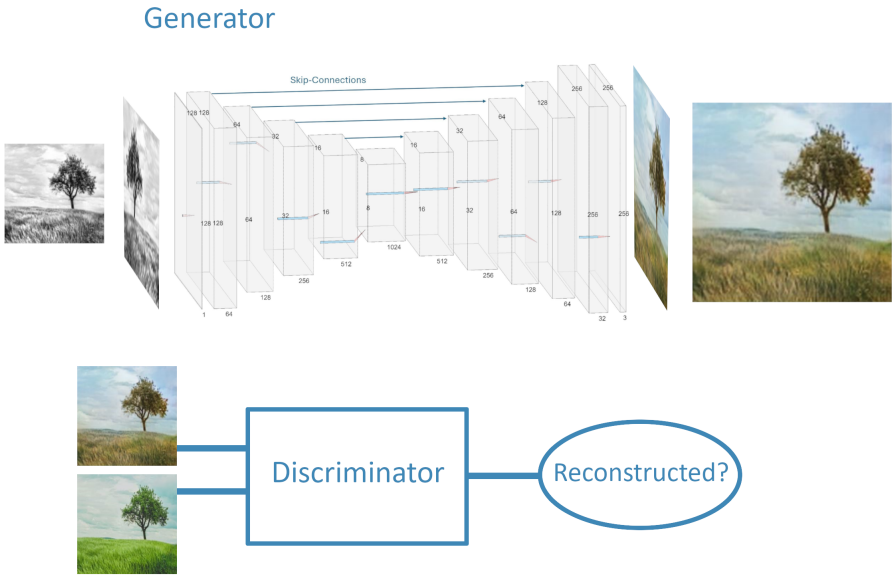

Figure 7: The architecture and functioning of our U-Net-GAN model

The loss functions we used are the Binary-Cross-Entropy-Loss for the Discriminator, and a combination of the Adversarial loss with Mean Absolute Error for the Generator:

$$\mathcal{L}_{\mathcal{D}} = \mathcal{L}_{fake} + \mathcal{L}_{real}$$

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

$$\mathcal{G}^* = \underset{G}{argmin}\ \underset{D}{max}\ \mathcal{L}_{GAN}(\mathcal{D}, \mathcal{G}) + \lambda * MAE$$

where x are input gray-scale images, y the colorized images and MAE is the $\mathcal{L}_1$ distance between x and y [7].
Tested activation functions: **LeakyRelu**.
Tested output activation functions: **Sigmoid**.

8.4 Predicted Validation Images examples

Figure 8: Comparison of, in order: Fusion Model, Vanilla, GAN, and Original Image (framed in red)

8.5 Zero-Shot Inference examples

Figure 9: Comparison of, in order: Fusion Model, Vanilla, GAN, and Original Image (framed in red)

8.6 Failed examples

Figure 10: Comparison of, in order: Fusion Model, Vanilla, GAN, and Original Image (framed in red)

8.7 Validation examples of the Large GAN (enlarged training set)

Here we employ the same architecture as

Figure 11: Large GAN predictions on the validation set

8.8 Examples of Large GAN on Zero-Shot Images

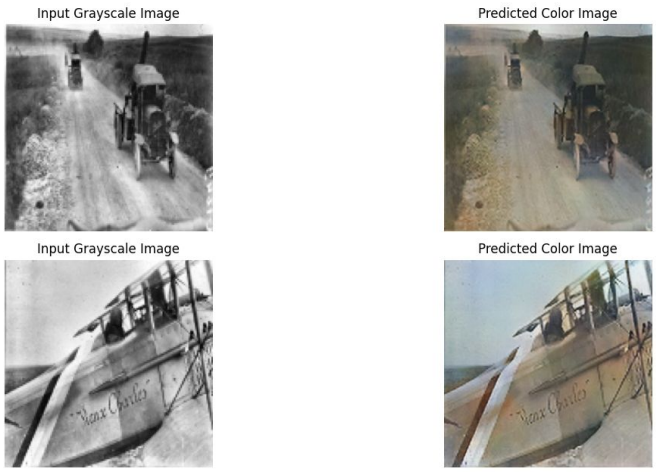

Figure 12: Large GAN predictions on zero-shot inference

8.9 Original GAN vs Large GAN

Figure 13: Comparison of large GAN (top Images) versus original GAN (bottom images)