B&W Colorising Models
Final Technical Report
Team MMM

Ashera Dyussenova
a.dyusenova@innopolis.university

Mark Zakharov
ma.zakharov@innopolis.university

Nikolay Pavlenko
n.pavlenko@innopolis.university

November 28, 2023

1 Introduction

In the realm of artificial intelligence, one fascinating application that has garnered significant attention is the colorization of black and white images. The absence of color limits the viewer's ability to fully connect with the historical or personal significance of these images. Image colorization is a task where advantages of the AI and deep learning over hard-coding are most readily apparent which has attracted our interest. The task itself is also relevant, as it can be applied from colorization and re-colorization of old film recordings to historical image data, of which no colorized version survived.

In our project we developed or implemented several **ML models: handcrafted CNN, Pix2Pix, and two VGG-based models**, built and evaluate different model architectures on different metrics, with the eventual goal of finding an optimal model for the task at hand.

2 Project Timeline

- **Weeks 3-5**: We researched the subject, found severals models that we have found interesting for the task at hand;
- **Weeks 6-10**: we trained those models, started the development of our own CNN model for the purpose of image colorization, ;

- **Weeks 11-13**: we modified the CNN model based on feedback received for the intermediary progress reports, evaluated all models and performed the comparison, recording the results in the final review.

3 Division of Responsibilities

- **Ashera Dyussenova**: researcher, trained and implemented the pix2pix model;
- **Mark Zakharov**: developer of the hand-crafted CNN model for image colorization, implementation of metrics for the evaluation;
- **Nikolay Pavlenko**: data analysis and preprocessing, implementation and evaluation the VGG models, primary report writer, project coordinator.

4 Dataset

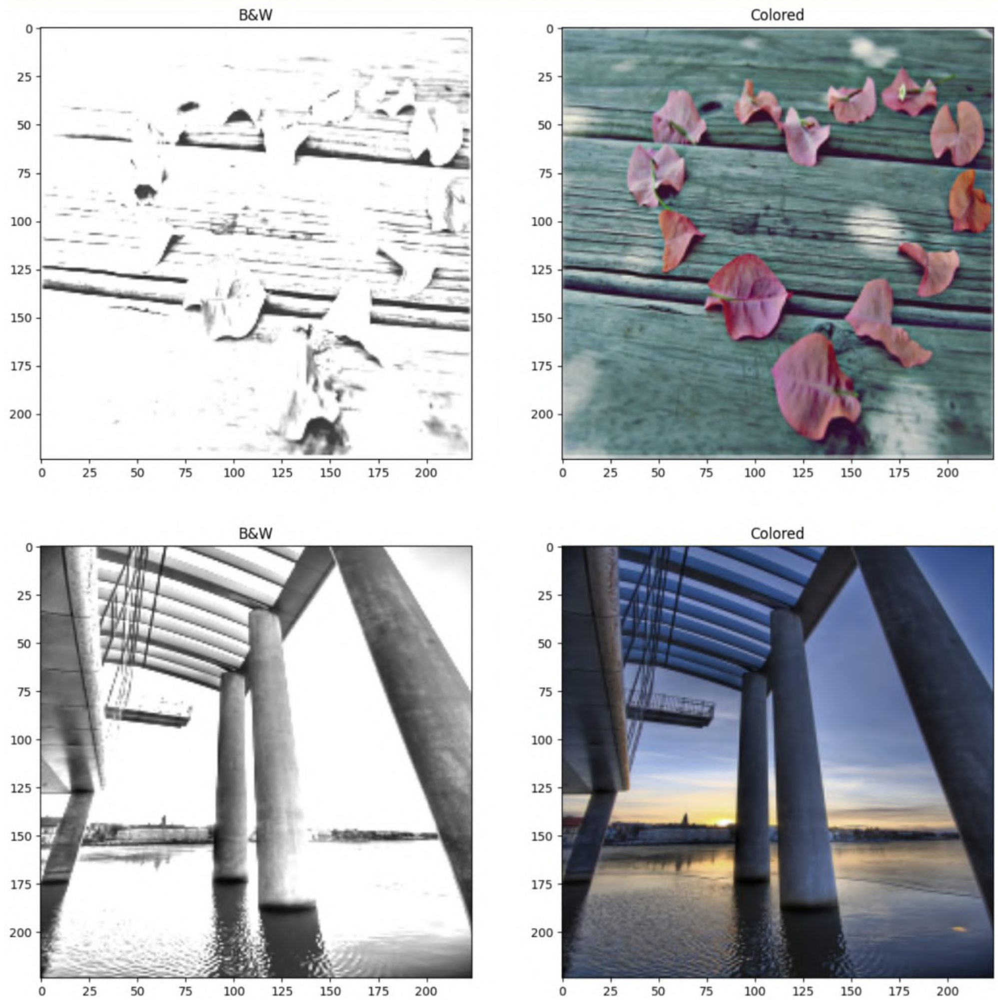
4.1 Datasets Exploration

To embark on this colorization journey, the selection of an appropriate dataset plays a pivotal role in training robust and effective models. Two datasets have surfaced as potential candidates for this purpose: the renowned Fashion-MNIST and a specialized Image Colorization dataset. Each dataset brings unique advantages to the table, presenting a choice between image quantity and uniformity versus image resolution and color diversity.

While at first we have considered using MNIST, in the end in our project we have used a specialized Image Colorization dataset from Kaggle, as it was more interesting to test models on a large and diverse dataset with vibrant color schematics, rather than images of digits that are black and white by default. The chosen dataset contains 25k 224x224 grayscale and colored images. While it has fewer datapoints than the Fashion-MNIST dataset, its images have better resolution and have colored ones, making it possible to train and test our models on it.

4.2 Describing the Data

Image Colorization dataset consists from images in **LAB** format: **L** - Lightness, **a** - Red/Green Value, **b** - Blue/Yellow Value. L and ab values are stored in different numpy arrays, so to get an image representation they have to be loaded and possibly combined. If we want to see a B&W image we only need to load the lightness values, while if we want to see a colored target image we also need to load the colors. Pictures in the dataset aren't strictly limited to a single topic, so models developed on this dataset are more generalized than if they had been developed on MNIST dataset. We provide some of the B&W pictures and their colored alternatives below:



5 Picking the Models

Image colorization is the technique of adding color to grayscale or black-and-white images. This process involves converting intensity values to a color space like RGB and filling in missing color data to create a full-color image. Various methods, including image segmentation, texture synthesis, and machine learning, are used for this purpose. One popular method employs deep learning, specifically convolutional neural networks (CNNs), trained on a dataset of color images. Another approach utilizes Generative Adversarial Networks (GANs), where a generator network produces color versions of grayscale images, and a discriminator network distinguishes between generated and real color images. Recent advancements in deep learning-based methods have yielded impressive and high-quality colorization results across diverse images. For this reason one of the models we picked for our project is Pix2Pix - a generative adversarial network. Finally, we have picked a simple 4-layered CNN model.

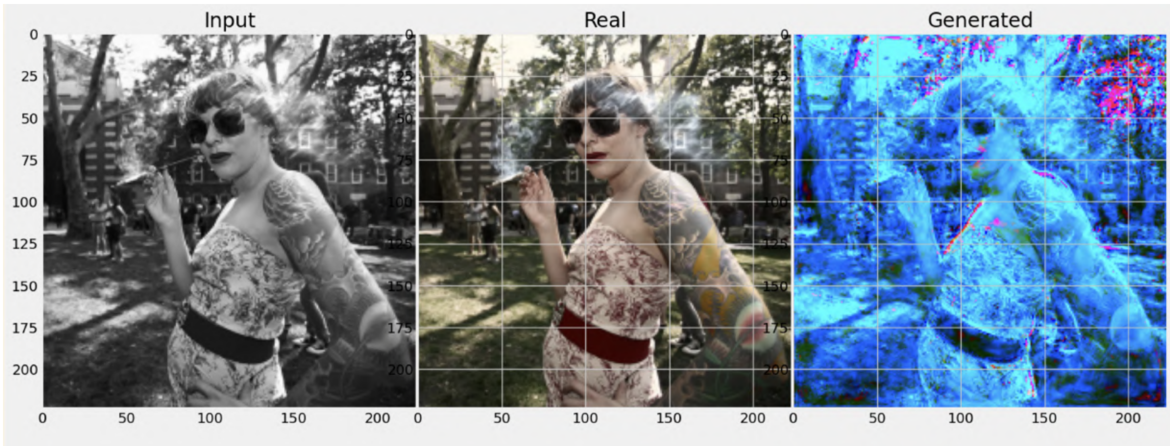
5.1 Pix2Pix

Authors of [2] have introduced a novel technique for image-to-image translation using a conditional Generative Adversarial Network (GAN) framework. This method employs a generator network trained to transform images from one domain, such as sketches, to another, such as pictures. The training involves adversarial loss and L1 loss. The generator takes an input image from the source domain and random noise, producing an image in the target domain. Simultaneously, a discriminator network assesses the authenticity of the generated image by comparing it to a target image from the desired domain, using a patch GAN architecture with 70x70 patches. Their study has proven the effectiveness of their approach across multiple tasks, which is why we have selected this model to colorize a portion of our dataset.

Two methods to boost the performance of Pix2Pix model include employing Wasserstein GAN (WGAN) and utilizing a U-Net architecture with residual blocks. WGANs use the Wasserstein distance metric for generator and discriminator training, improving stability of the pictures produced, as well as enhancing their realistic feel. U-Net is integrated and tailored specifically for image segmentation, while residual blocks enable detailed learning from input images - it is especially useful in colorization tasks. This approach enhances stability and the capacity to discern complex image edges, leading to more authentic-looking photos.

Examples of colorization by Pix2Pix are given below:

To test how well the GAN would be at colorization if it was re-trained, rather than pre-trained on our dataset, we completely re-trained it and used that version of the model in our evaluation. Quality of colorization became markedly lower:

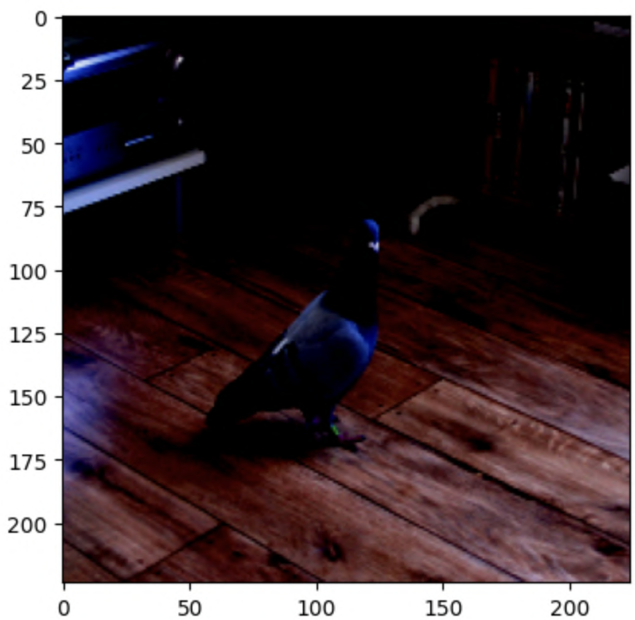


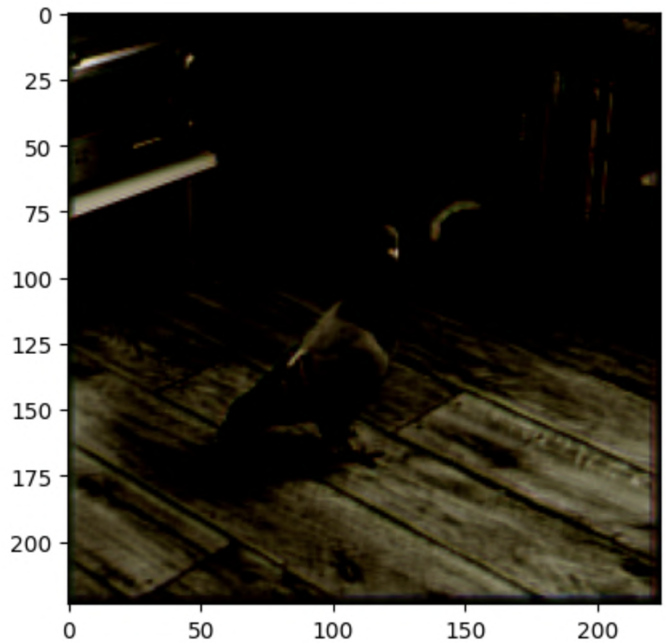
5.2 CNN

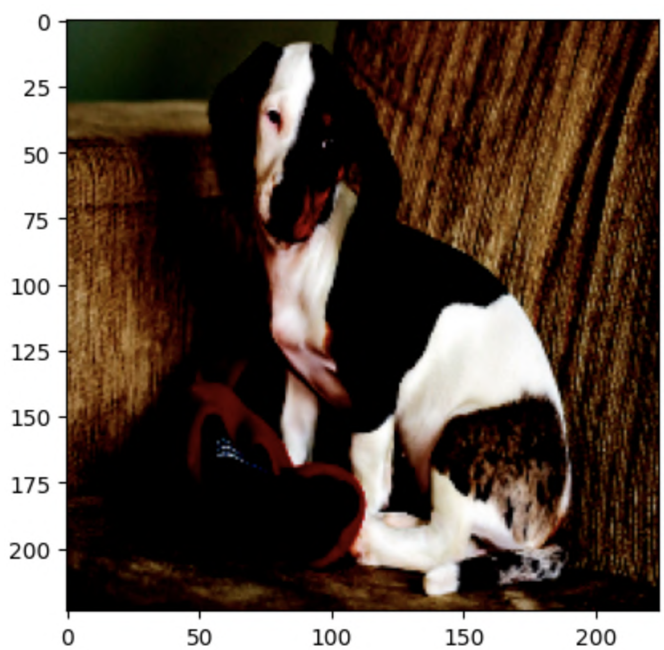
The architecture of the model we've implemented for colorizing black and white images is a straightforward Convolutional Neural Network (CNN) composed of four layers. The initial layer in the model is a convolutional layer equipped with 12 filters, a kernel size of 3, and a stride of 1. The activation function employed here is the Rectified Linear Unit (ReLU). This layer applies 12 distinct filters to the input images, each generating a feature map that encapsulates certain attributes of the image. The second layer mirrors the first in terms of configuration and hyperparameters. The third layer is a deconvolutional layer, also known as a transposed convolutional layer. It has 12 filters, a kernel size of 3, and a stride of 1. This layer operates in reverse to a standard convolutional layer, upscaling the feature maps from the preceding layer rather than downscaling them. This is commonly used in models that need to augment the spatial dimensions of their inputs, such as in tasks like image colorization or segmentation. The fourth and final layer is another deconvolutional layer, this time with 3 filters. This layer generates the final output of the model, which are the colorized images. The model is compiled using the Adam optimizer and the Mean Squared Error (MSE) loss function. The Adam optimizer is a favored choice for training deep learning models due to its efficiency and minimal memory requirements. The MSE loss function is suitable for regression tasks, which aligns with this task as the model is predicting continuous color values for each pixel in the images. The model is trained with a batch size of 64 over 5 epochs, which is sufficient as we observe that our loss is not rapidly decreasing. This model serves as a preliminary exploration into how a simple CNN performs on this task and is used primarily for experimental purposes. Although the model attempts to achieve some results, the output is predominantly greenish pictures. While this is an interesting outcome, it's not practical for real-world applications.

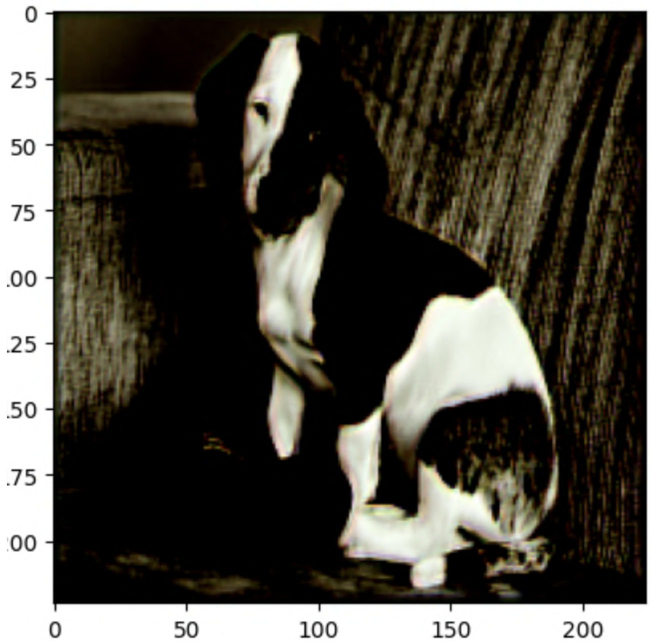
The CNN model was trained and tested on a portion of the dataset - 2500 images. Training took several minutes and 5 epochs, and when it finished the loss was still fairly high. If we look at the results of colorization by this model and compare them to Pix2Pix, we can definitely see that the images are of a lower quality, they seem darker and colours are less well-defined.

Examples of colorization by our CNN model are given below. The first image is test image, the second is the result of colorizing with the help of our model:




0
25
50
75
100
125
150
175
200
0
50
100
150
200


0
25
50
75
100
125
150
175
200
0
50
100
150
200



As we can see, the CNN colorization tends to use more brownish tones, rather than bright ones we would expect.

6 Colorful Image Colorization

In[1] authors presented a fully automatic model using a convolutional neural network (CNN) trained on over a million color images. The approach treats colorization as a classification task, predicting a distribution of possible colors for each pixel to capture the inherent multimodal nature of the problem.

To address the uncertainty, the authors employed class-rebalancing during training to increase color diversity and weight rare colors more heavily. The final colorization was obtained by taking the annealed mean of the color distribution. The proposed method achieves vibrant and realistic colorizations, outperforming previous approaches in a "colorization Turing test," fooling human participants on 32 % of trials, so we believed that it would serve as a good baseline for our own model.

Two versions of the same model with different pre-trained parameters were employed in our task, the ones for presenting the model at ECCV-16 conference, and for presenting it at SYSGRAPH-17 conference.

An example of colorization performed by this model is given below:

7 Hypertuning of the CNN model

7.1 HSL format

HSL (Hue, Saturation, Lightness), as opposed to RGB (Red, Green, Blue) are two different color models used in digital imaging, each offering distinct advantages. While RGB works by combining various intensities of red, green, and blue light to produce a broad spectrum of colors, HSL is a cylindrical-coordinate representation of colors, where Hue refers to the type of color (e.g., red, blue, green), Saturation defines the purity or vividness of the color, and Lightness determines the brightness from black to full intensity.

In the context of image colorization task at hand, a colorization to HSL could offer more intuitive control over color adjustment due to its characteristics. HSL separates the representation of colors into components that are more relatable to human perception compared to the direct combinations of red, green, and blue channels in RGB. For colorization, HSL's separation of color (Hue), purity (Saturation), and lightness provides a significantly more natural way to manipulate and control the hues and saturation of colors in an image. It allows easier adjustments to the vividness and brightness without impacting the underlying color relationships, making it particularly useful in tasks like color grading or recoloring images where preserving the natural look and feel is crucial. The decoupling of intensity (lightness) from the pure color properties (hue and saturation) also could also be advantageous in image colorization by a deep neural network.

7.2 Changing Hyperparameters

In order to test the new colorization to HSL we have created three CNN models that share the same architecture but have different hyperparameters:

1. **model1** has the same hyperparameters as the model that was employed in image colorization the previous week

2. **model2** has stride of 2, rather than 1, to check how much of an influence it has on the accuracy of colorization

3. **model3** has stride of 1, but it has a greater number of filters than model1, possibly allowing it to catch more rare and unique image features.

7.3 Colorization Results:

Figure 1: Original pictures/RGB colorizations

Figure 2: model1 colorizations

Figure 3: model2 colorizations

Figure 4: model3 colorizations

7.4 Conclusions

While switching the colorization to HSL worked in removing the brown tint from the colorization, all the colorizations are now of a decidedly greenish tint, and many of them have artifacts, which are especially visible and egregious on results of model1 and model2. An increase in stride value has also led to a degradation of the quality of the image, so we do not intend to deviate from stride equal to 1 in the future, as the benefit from a faster convolution process are outweighed by the reduction in quality and accuracy. Filter number could be changed in the future, if more tests on filter size are conducted.

8 Results

To correctly estimate the results of our evaluation, the process has to be described more fully. Two metrics were calculated in the evaluation: MSE - mean squared distance between the colorized image and actual color photo, and CIE76 - mean of squared difference between pixel values in the LAB format. Those are both metrics widely used in estimating how far the colorized image is from the actual colorful image.

Dataset was split into training and testing parts. With 20k datapoints relegated to the training and 5k for the testing.

The models used in evaluation are as follows:

- **Custom CNN** - our model for image colorization, trained on a portion of the dataset,
- **ECCV** - the CNN model from [1] with pre-trained parameters for ECCV-16 conference
- **SIGGRAPH** - the CNN model from [1] with pre-trained parameters for SIGGRAPH-17 conference
- **CWGAN** - the generator model from [2] re-trained on our dataset with few epochs due to very high training time and limited computational resources

Model	**MSE**	**CIE76**
Custom CNN	0.047	41.18
ECCV	0.012	0.155
Siggraph	0.011	0.13
CWGAN	0.387	0.597

Table 1: Metrics

As we can see from the results, our CNN model achieves satisfactory results on the MSE metrics. It is worse than the pre-trained CNN on a million of ImageNet images, but not by a lot, which signifies that the model we have developed is quite good at colorization, even though it's still not as effective as the existing alternative. CWGAN, on the other hand, has a very high error rate, and if we view the images that it colorized, we see that the results of that model are objectively the worst. This can be explained by the fact that CWGAN is a generative adversarial network that was trained on a small dataset for just a few epochs, which is why the generator wasn't able to be trained sufficiently well to present a viable alternative to the CNNs. From this we can conclude that CNNs are a better option for image colorization when time and resources are constrained.

The CIE76 results are curios as well, as they indicate that our custom CNN model has significantly worse results in approximation to the images in LAB format. This could be caused by the colorizations produced by our model being too dark, hence deviating a lot from the expected lightness values. This could be changed by training on images in LAB format, rather than RGB.

9 Repository Link

https://github.com/Daru1914/B-W-Colorising-Models

10 References

References

[1] Zhang R., Isola P., Efros A. A. Colorful image colorization //Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. – Springer International Publishing, 2016. – C. 649-666.

[2] Isola P. et al. Image-to-image translation with conditional adversarial networks //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2017. – C. 1125-1134.