Machine Learning Engineer Nanodegree
Capstone Project Proposal – Image Colorization

Maxime GENDRE

Udacity

Abstract- In computer vision, there are many typologies of problems to resolve, like object detection, image segmentation, classification, image generation. A very exciting thematic is about how to get a good quality picture, with a noisy / bad quality image, or colorize a black and white / old picture.

I. Domain Background

For a long time, scientists, developers and researchers have been working on : How to generate colors in a black and white image. Nowadays, we have a lot of papers which is the result of several years of their work, to guide and allow people to do this task more easily, and efficiently. Moreover, a lot of open data are available today, especially in computer vision (MS Coco, Open Image Dataset V6, …)

Machine Learning tasks in image processing generally requires a large amount of data to achieve a good result in a specific task, due to the large number of factors that define a "good" and "usable" image.

Color photography is photography that uses media capable of capturing and reproducing colors. By contrast, black and white (monochrome) photography records only a single channel of luminance (brightness) and uses media capable only of showing shades of gray. (source: Wikipedia)

The first method to colorize a photography in 3 channels has been released in a 1855 paper, by Scottish physicist James Clerk Maxwell.

As you can see, since a long time, humans are working on this problem.

After some research, I found these papers:

- Image-to-Image Translation with Conditional Adversarial Networks (Authors: Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros)
- Colorful Image Colorization (Authors: Richard Zhang, Phillip Isola, Alexei A. Efros)

II. Problem Statement

A first method to realize our colorization is to use $L*a*b*$.
$L*a*b*$ is a color space, defined by the CIE (*Commission Internationale de l'Eclairage*)
Respectively, $L*$ indicates lightness, $a*$ is the red/green coordinate, and b* the yellow/blue.
Secondly, we have RGB, which is commonly used today to attribute colors to images. R is Red, G is Green, and B is Blue. On an image, we will have for each pixel, a value for those three variables.
There are more color space than those two, but I won't describe them in this project.

To train a model for colorization, we have X, which is our train data, and y, our target. In our case, we have a grayscale image as input, which don't have any color space, and as output, our colors.
The problem is to determine with a grayscale image / value, a corresponding color.

We can use many methods to do it, for example, let's take a look to the $L*a*b*$. We will assume that X is our lightness factor known as $L*$ and y our target is $a*b*$ which describe colors. Once our model predicts colors, we just have to concatenate predictions with lightness, and here we are. Our black and white image has been colorized.
A second hypothesis is to use RGB color space. Here again, we will try to predict our 'R' 'G' 'B' values, from our input, which is a grayscale image.

III. Datasets and Inputs

The dataset which I will use will be the MS COCO 2017 dataset to obtain a great variety of pictures. "COCO is a large-scale object detection, segmentation, and captioning dataset.". To download data, we can use **fastai** python package. To feed my Train set and Test set, I will only take 11000 images (9000 Train samples, 3000 Test samples). We can find several classes in COCO Dataset, but I won't target to do object detection or classification, so I'll not take consideration about classes and I will just use random images. I need a great diversity of scenes, landscapes, brightness images.

IV. Solution Statement

There are many ways to complete our job with machine learning. Some papers do classification models, others use regression approach. In my case, I'm going to use **Conditional Adversarial Networks**, with **pix2pix** (Image-to-Image Translation with Conditional Adversarial Networks) paper. Two losses are used in this paper: **L1 loss**, to do regression, and an **adversarial loss** (GAN). The proposed solution is to use conditional **GAN** for our task to colorize images. A GAN has a generator, and a discriminator model, to learn how to solve a problem together. **Generator** will take black and white images ($L*$) and produces 2 channels images ($a*b*$). **Discriminators** aim to determine if it's a fake, or a real image, with all previous channels concatenated.

In this paper , the loss of conditional GAN is a function which works like this: $\boldsymbol{x}$ is our grayscale, $\boldsymbol{y}$ our 2-channels output that the generator produces, $\boldsymbol{z}$ as input noise for the generator, $\boldsymbol{G}$ the generator model and $\boldsymbol{D}$ the discriminator. This loss function will help to produce "real" colorful images.

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

Image from paper – Objective of a conditional GAN can be expressed like this

Another loss that we will use is the **L1 loss**, known as **mean absolute error**. In the paper, they are combining the previous loss we saw with the **L1 loss** to assure that the model's color choices are the best.

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1].$$

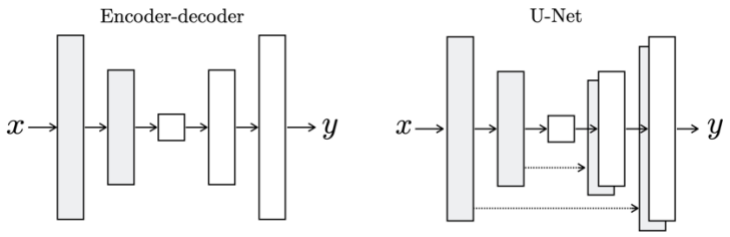
Image from paper – L1 loss

The final objective with the combination of those two functions is:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L1}(G).$$

$\boldsymbol{\lambda}$ is a coefficient to balance the contribution of the two losses to the final loss.

Bellow, this is a representation of our U-Net, which will be our Generator layers architecture (*U-Net*).



"*Two choices for the architecture of generator. "U-Net" is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks."* - paper

V. BENCHMARK MODEL

It is difficult to find a true benchmark about the GANs because the metrics evaluating the quality of the result are mainly based on human's perception. However, GANs are in constant evolution, and there are some metrics which allow us to do a well-performing approach. Some papers / implementations are using the "Fréchet Inception Distance", "Inception Score", or even the "SSIM". We can also use the MSE, but it's the less used metric. In the following benchmark, you will see 2 examples of metrics benchmark.
In the next section, "Evaluation Metrics", I will detail what are those metrics, and why they are interesting in our context, on GANs.

SAGAN (paper):
- Fréchet Inception Distance: **18.65**
- Inception Score: **52.52**

cDCGANs (medium post):
- SSIM: **0.93**

VI. EVALUATION METRICS

Regardless of the metric used, it is difficult to associate a GAN result with a metric since the quality of the result is primarily based on human perception, the result is more of a subjective assessment. In my implementation, I will try to take a sample of varied SSIM, to see and demonstrate that a low SSIM can still produce a realistic result in order to have a better understanding of the model result.

In this implementation, I will use several metrics, in several purpose.
The first metric is the **Structural Similarity Index** (**SSIM**).
SSIM is a metric used to measure the similarity between two given images. It's a perception-based model that measures changes in the structural information of images as a good approximation of perceived image distortion. The abstract of the paper of **SSIM** show globally the background of SSIM.

"Objective methods for assessing perceptual image quality traditionally attempted to quantify the visibility of errors (differences) between a distorted image and a reference image using a variety of known properties of the human visual system. Under the assumption that human visual perception is highly adapted for extracting structural information from a scene, we introduce an alternative complementary framework for quality assessment based on the degradation of structural information."

Image Quality Assessment: From Error Visibility to Structural Similarity paper - Zhou Wang

There are 3 key features that the SSIM use in an image: **Luminance**, **Contrast**, and **Structure**.

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)}.$$

SSIM Index Formula - Image Quality Assessment: From Error Visibility to Structural Similarity paper

SSIM is based on the comparison between luminance l(x,y), contrast c(x,y), and structure s(x,y), where x and y are the original image and the compared image α, β, and γ are weights for each of those characteristics.
The lowest value for **SSIM** is **-1**, and the highest is **1**. (1 means a perfect similarity)
This measure will be used to compare two images, and calculate a score for similarity between them with window sliding (local features sensibility).

My second measure that I will use is the **Inception Score.**
The Inception Score is an objective metric for evaluating the quality of generated images, specifically synthetic images output by generative adversarial network models. So, in our case, this metric fit to our needs. Inception Score lowest value is **1.0**, and the highest value is the **number of classes** supported by the classification model. (Max = 1000)

Inception Score paper - Improved Techniques for Training GANs - Tim Salimans

The last measure is the "**Fréchet Inception Distance**". A well-performing approach to measure the performance of GANs is the "Inception Score" which correlates with human judgment. The FID (Fréchet Inception Distance) will be applied on the whole generated images from the test set, versus the target test set, to make an FID Score.

"For the evaluation of the performance of GANs at image generation, we introduce the "Fréchet Inception Distance" (FID) which captures the similarity of generated images to real ones better than the Inception Score."

Paper which introduces the FID (inspired from Inception Score).

VII. Project Design

The workflow of my project will be decomposed in 9 parts:

A. Collect a heterogeneous image dataset (MS Coco 2017) in order to have a wide variety of images and to obtain a general model
B. Visualize, analyze the dataset, and evaluate some preprocessing techniques to choose which one to use to convert my colored pictures into grayscale. See impact of preprocessing in the histogram, and luminance parameters.
C. Then, I will start to implement deep learning neural networks, Generator, Discriminator, as I explained earlier.
D. I need to implement my metrics methods, and some utils methods to add them into my train / evaluate future methods for the next step.
E. At this stage, I will probably try to do a first train, see results, and move some hyperparameters, epoch number, in order to improve my models.
F. My evaluation process will be like this:
    a. Use my Test set with generated images versus targets images and see what my metrics are.
    b. Analyze and do some statistics of my metrics to get some conclusions about my model's performance
    c. Take a look to my results, without considerations of my metrics (GANs require this, as I explained earlier).
G. After some iterations, I will get a result, and I will try to demonstrate as I said, an example of bad SSIM metric, but realistic result even if the score is bad.
H. The last step will be to identify what I can do to improve this model, and what are the Pro / Cons of this implementation in order to challenge it.
I. A conclusion to explain if my solution was the good one for our problematic, which is: Colorize grayscale images.