ADVANCED MACHINE LEARNING

IMAGE COLORISATION USING GAN's

Abstract

Image colorization is a fascinating area in computer vision that aims to automatically add colors to grayscale or black-and-white images. Traditionally, this task required manual intervention by artists or graphic designers, but with the advent of deep learning and Generative Adversarial Networks (GANs), automatic image colorization has become more accessible and efficient. This report provides an overview of image colorization using GANs, a state-of-the-art technique that leverages the power of deep learning to generate realistic and visually appealing color images.

1. Introduction

1.1 Background

Image colorization is a fascinating topic to study in deep learning. It involves computer vision techniques, such as image classification, and image segmentation working together. Traditionally, to colorize an image, a model must first classify the various objects present in the image. Next, the model must locate the object in the image. Last, the model must apply the correct color to all the objects present in the image. One might think that there are too many choices for color, a shirt may be of any color or shade in the RGB spectrum, a wall may be painted in any color and a car may have any paint job. Thus, our problem is multimodal, where we may have multiple valid predictions for a single image. This is partially true - while there exist many choices for common objects, most objects have a specific choice of colors. For example, the sky is either blue or black depending on the time of day, the sun is always yellow, the moon is always white, etc. A common problem faced while training models is the lack of data. This is not a problem for this task, as we can simply convert existing images to grayscale and pair them with their color versions to train the network. Thus, predicting color is more tractable than we think, especially using Deep Learning.

Two popular methods exist to colorize grayscale images using machine learning – user-guided, and automatic colorization. In the first method, a human provides additional semantic input, usually by drawing strokes over the grayscale image, providing hints to the model. This method has historically provided accurate results but requires intensive user interaction. The second model works either by matching a picture to its colored version and learning parametric mappings from grayscale to color from large-scale image data. This is the method we will be using to build our model.

1.2 Problem Statement

Image colorization is a translation problem, where we try to map the input image to an output image, where the images are represented as high dimensional arrays of numbers. It can be seen as a regression problem on individual pixels, where we try to predict the values of each pixel in the output based on the pixel value in the input. Our network will try to generate an output but with different values for the pixels.

1.3 Algorithm

Generative models are unsupervised learning models that model the patterns and irregularities in the input data and can generate new examples that are indistinguishable from the ones in the input data. A Generative Adversarial Network (GAN) is a method to train such generative models by pairing them with a discriminative model. Hence, a GAN has two sub-models, a generator model that generates new examples, and a discriminator model that classifies whether the examples are real (from the input data) or generated by the generator model. The term adversarial implies that the model rewards conflict between the generator model and the discriminator model. There are many applications of such generator-discriminator networks, including generating photorealistic images of people, objects, or scenes.

2. Dataset

We used the CIFAR-10 dataset to train both our models. The images in this dataset have the dimensions 32x32 and are encoded in the RGB colorspace. The 32x32x3 images in the RGB colorspace were converted into the CIE L*a*b colorspace. The l (luminance) channel is sufficient to represent the grayscale image and all the color information is encoded into the a and b channels. Thus we could generate a dataset with input dimensions 32x32x1, and output dimensions 32x32x2 instead of dealing with 32x32x3 images as the input and output images. To get the final image from the output, the l channel is concatenated with the a and b and converted back to the RGB colorspace.

3. Implementation

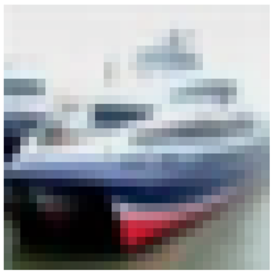
3.1 GAN's

In our scenario, the input to the generator is the grayscale image, and the output we'd like it to generate is a colorized version of the grayscale image. we'd like the network to learn a transformation from the input image to a target image.

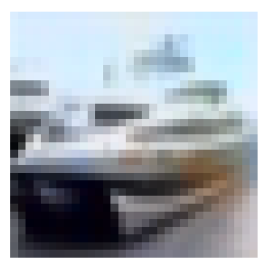
The generator model in our implementation follows the encoder-decoder pattern as in the figure. The input is a grayscale 32-by-32-pixel image represented by the l-channel of the CIE L*a*b* colorspace. The output is a 32-by-32-pixel image represented by the a and b channels of the CIE L*a*b* colorspace. The encoder layers of the generator downsample the input image with a series of encoders into a much smaller higher-level representation of the image features. Each encoder comprises a convolutional layer, followed by a batch normalization layer and an activation function. The decoder stack upsamples this high-level feature representation and reverses the action of the encoder layers. Each decoder comprises a deconvolution (the transpose of a convolution operation), followed by batch normalization and an activation function. The activation function used in the encoder in Leaky ReLU. Batch normalization is applied to the transposed convolution in the decoder. The activation function used in the decoder blocks is ReLU.

The discriminator is a more conventional classifier. It takes in the l-channel from the input data stacked with the ab channels from the generator output or the input data. It returns the probability of the image being generated by the generator.

4. Results



sequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan. Nam liber tempor cum soluta nobis eleifend option congue nihil im placerat facer possim assum. Typi non habent claritatem qui facit eorum claritatem. Investigationes demonstraverunt legunt saepius.



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, ıod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Dolor sit amet

Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan.