AMERICAN UNIVERSITY OF BEIRUT
FACULTY OF ARTS AND SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

Project report

Student name: *Mohamad Tfaily (201902722)* *Sarah El Annan (201800076)*
Salim Osman (201802254) *Ziad Osman (201902597)*

Course: *Artificial Intelligence (Cmps 287)* – Professor: *Dr. Shady Elbassuoni*
Due date: *Sunday, 22 March 2020*

Abstract

Provide a summary (typically two paragraphs) describing the problem you would like to tackle and its significance.

For our group project, we are interested in taking a gray scale image and colorizing it. That is to say, our algorithm should take an image that solely consists of the colors gray black and white (of different shades), and colorize it based on the real colors of the objects portrayed in it. Other than nostalgic significance, where you can make old photos of your loved ones come to life with colors, our project proves useful for any field requiring more detail for their images. For example, our project is proving very useful in forensic science, where some cases date back to the 80s, making it so that any pictorial evidence is in black and white. On a similar note, security cameras, that opt out to take footage in low resolution black and white (to save storage), are proving to be less than useful in finding incriminating evidence. This under-performance of security cameras is hindering arrests, and with a project like ours, security footage can suddenly become immensely more useful. Finally, our project helps bring historical images to life, in turn helping historians better understand the time eras of the past.

Introduction

Describe your problem in details and summarize your approach and findings.

As with most image based machine learning problems, we settled on a CNN architecture. rather than representing our images in RGB, we opted to represent them in LAB space, here's why:

in LAB space, similar to RGB, we have 3 components: L, a* and b*. However, unlike in RGB, the L component in LAB space represents lightness, which is the same for an image whether the image was colored or in greyscale. What that means is, when the user inputs a greyscale image, we already have the component L, which means we only have the two remaining values a* and b*. Note that if we were representing our images in rgb, we would have to predict three values Red, Blue and green, making LAB space the obvious choice.

with our method in place, we learned how to convert images into matrix form, and learned how to convert from grey-scale to L, and, eventually, learned how to convert from LAB space to RGB in order to plot the output.

Dataset

Describe the dataset you used to train your models.

For our Dataset, we scoured the internet and ended up finding 2 complete datasets of images. we joined the two datasets together to form our own dataset made up of 21,000 colored images. After doing so, we converted the images to matrix form. the matrix form images were in RGB, and so we used a function to turn them from RGB to LAB space. Also, we split the matrix form images into two sub-matrices, one containing the L values, and the other containing the a* and b* values. Finally, we fed our CNN with L values and let it output predicted a* and b* values (which the CNN compared to the real a* and b* values).

Model

Describe the models you used to tackle your problem. Please be specific and concise. Also, provide rational behind your choices.

Since our model's performance is based on predicting accurate numbers (the pixels), we found that a deep CNN was the way to go. To stay on the simple side, and to prevent our computers from dying, we decided to use 12 convolutional layers. In addition, we used 3 up-sampling layers to reduce the pixelation effect and so that the model can zoom more on small regions. And so, our Up-sampling layers improve accuracy. Its important to note that, since we care about each pixel separately, we opted out of using pooling layers as we thought it would negatively affect our results (since it only considers limited info out of a set of pixels). After training our model, we input a black and white image. This image gets converted into a LAB space matrix, where the L is put into the CNN that predicts a* and b*. with our L, a* and b* values ready, we create an array to represent the predicted colored image, turn the array from LAB space to RGB and finally plot the predicted colored image.

Results

Describe the experiments you ran to train and test your models. Also, provide all results of your experiments and discuss them.

We fed our CNN 21,000 images in LAB matrix form, specifically the L value. the CNN outputs predicted a* and b* values, upon which the CNN compares to the real values, after which it uses gradient descent to fix the weights accordingly. We ran the model for 50 epochs for a total of 8 hours of training time (using GPU). We found that 21,000 images was not enough to train our model to predict accurate numbers. We also found our hardware to be a limiting factor in how much we could train our model, as we zigzagged around OOM (out of memory) errors every step along the way. How we went about preventing those is we split our data into subsets and trained our model on those subsets separately. The results of our trained model showed promise, we were seeing good blue predictions and okay green predictions. We are confident that with a bigger dataset and, more importantly, a stronger hardware (enabling a much higher number of epochs of around 1000), we would have been able to train a really solid model.