Coloring Black and White Images

Chirag Jain
University of Georgia
Athens, Georgia
Chirag.Jain123@uga.edu

Pranay Reddy Armoor
University of Georgia
Athens, Georgia
pranayreddy.armoor@uga.edu

Vamsi Nadella
University of Georgia
Athens, Georgia
vamsi.nadella25@uga.edu

Abstract

Here we propose to create a deep learning convolutional neural-nets that will help to automate image colorization that is adding colors to gray-scale images. Our reason for choosing convolutional neural-nets is that the color of each pixel is strongly dependent on the features of its neighbors. Motivated by the recent success of deep learning techniques in image processing, we propose a feed-forward, two-stage architecture based on Convolutional Neural Network that predicts the U and V color channels. Given a grayscale image as an input, we try to generate plausible color version of the image. We try to eliminate the need for manual coloring of gray-scale images using existing software like photoshop. Also, we demonstrate how different architecture produce different results when significant changes were made in architecture.

1.Introduction

Automatic image colorization addresses the problem of adding colors to monochrome images without any user intervention. There are plenty of practical applications of colorization such as colorizing old movies or photographs, color recovering, artist assistance and visual effects. Apart from that, there are a huge number of applications where we want to predict values or different distributions at each pixel of an arbitrary input image, exploiting information only from this input image.

Neural Network has been an emerging technique for computer vision/image processing tasks. One such part of Neural network is Convolutional Neural Net(CNN). These deep learning techniques have shown outstanding results on many computer vision problems such as image classification object detection and tracking, handwritten character classification and many more. CNN can do tasks which were a few years ago could only be done by humans. The reason CNN have become popular is because they are simple as compared to the tasks they perform. We attempt to solve one such task - the task of 'coloring' a black and white image without human intervention. Automated colorization of black and white images has been subject to much research within the fields of computer vision and machine learning. Coloring a gray-scale image is an important problem because this task is not an easy task for an average human brain and the fact that a neural network is able accomplish this indicates that neural networks today are performing certain complex tasks better than humans.

Consider the grayscale image in Figure 1. At first glance, hallucinating their colors seems daunting, since so much of the information has been lost. The task is to predict colors for the grayscale image in a way that it is not easily recognized by a human eye that previously it was a grayscale image. Convolutional Neural Networks helps in doing this magic. Our goal for this project is to color an image in such a way that a human observer cannot recognize it is done by automated colorization. Therefore, our task becomes much more achievable: to model enough of the statistical dependencies between the semantics and the textures of grayscale images and their color versions to produce visually compelling results. In this project, we attempted three CNN architectures for coloring black and white images without human intervention. Our architectures exploit the most recent advances in CNN design and training techniques.

Fig 1: Black and White image(left), Color Image(right)

2. Related work

Early approaches of colorizing an image involves human effort for identifying a source color image which can be used to color the target image either by transferring the colors or by taking help of human annotator to server as a set of hints.

In recent methods, the CNN are trained to convert a gray scale image to a single-color image. The models are trained with L2 or L1 loss which results in washed-out colorization of an image. This happens as the

model predict the average color. Some approaches use per-pixel cross entropy on the softmax CNN output which results in more colorful images. These approaches lack in giving same color to all the pixels in a region, since the model predict each pixel color independently.

Cheng et al. [3] proposed a fully-automatic colorization method based on three-layer deep neural network and hand-crafted features. There were three levels of features that were extracted from each pixel of the training images which were then concatenated and used to train a deep neural network. Ryan Dahl [18] proposed a CNN-based approach and utilized a pretrained CNN for image classification [22] as a feature extractor. Their approach was to apply trained residual encoder that provides color channels.

Our project is inspired in part by Ryan Dahl's CNN based system for automatically colorizing images. In terms of results, Dahl's system has outstanding performance in colorizing foliage, skies, and skin. The noticeable thing in this task is that the images generated by the system are predominantly sepia-toned and muted in color. We note that Dahl formulates image colorization as a regression problem wherein the training objective to be minimized is a sum of Euclidean distances between each pixel's blurred color channel values in the target image and predicted image. Although regression does seem to be well-suited to the task due to the continuous nature of color spaces, in practice, a classification-based approach may work better.

In regression-based system, the predicted pixel value loss is minimized using L2 loss taking the mean pixel value. Accordingly, the predicted pixel ends up being an unattractive, subdued mixture of the possible colors. A regression-based system would tend to generate images that are desaturated and impure in color tonality, particularly for objects that take on many colors in the real world, which may explain the lack of punchiness in color in the sample images colorized by Dahl's system.

The other related work in this task is Fully automatic image colorization based on Convolutional Neural Network [5] paper. This paper introduced a fully automatic colorization algorithm based on VGG-16 and a two-stage CNN. VGG-16 provided multiple discriminative, sematic information which was used to train a two-stage CNN architecture without pooling layers which proved it a richer representation by adding information from a preceding layer. The U and V color channels were predicted in the algorithm because the LUV space had been developed to add color channels to the Luminance value (L). In addition to this, LUV minimizes the correlation between the three coordinate axes since the only need to predict the two channels U and V using L channel.

3. Approach

Black and white images can be represented in grids of pixels. Each pixel has a value that corresponds to its brightness. The values span from 0 - 255, from black to white respectively. RGB Color images consist of three layers: a red layer, a green layer, and a blue layer. Just like black and white images, each layer in a RGB color image has a value from 0 - 255. The value 0 means that it has no color in this layer. If the value is 0 for all color channels, then the image pixel is black. A neural network creates a relationship between an input value and output value. In our colorization task, the network needs to find the traits that link grayscale images with colored ones (Figure 2). We are searching for the features that link a grid of grayscale values to the three-color grids.

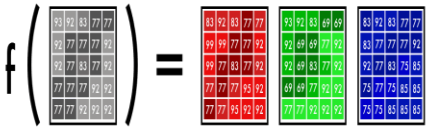

Figure 2: [3] link a grid of grayscale values to the three-color grids

During training time, our program reads images of any size, we resize it to dimension of 224 X 224 pixels and 3 channels corresponds to red, green, and blue in the RGB color space. The images are converted to CIELUV color space. The black and white luminance L channel is fed to the model as input. The U and V channels are extracted as the target values. During test time, the input to the model is a 224 X 224 X 1 black and white image. It generates two tensors, each of dimension 224 X 224 X 1, corresponding to the U and V channels of the CIELUV color space. The three channels are then concatenated together to form the CIELUV representation of the predicted image. Then, the LUV image is converted to RGB image which is desired result.

In our CNN model, we have used ReLU as the activation function in all layers of the model except for the last layer, which has softmax as activation function. The ReLU has been empirically shown to greatly accelerate training convergence. Also, it is much simpler to compute than many other

conventional activation functions. For these reasons, the rectified linear unit has become standard for convolutional neural networks.

In the architecture, during the leftmost column of layers which are inherited from a portion of the VGG16 network, the size (height and width) of the feature map shrinks while the depth increases, and model learns a rich collection of higher-order abstract features. During the right layers, the network successively up scales the preceding layer output, merges the result with an intermediate output from the VGG16 layers via an elementwise sum, and performs a two-dimensional convolution on the result. This enables the network to realize the more abstract concepts with the knowledge of the more concrete features so that the creating process will be both creative and down to earth to suit the input images.

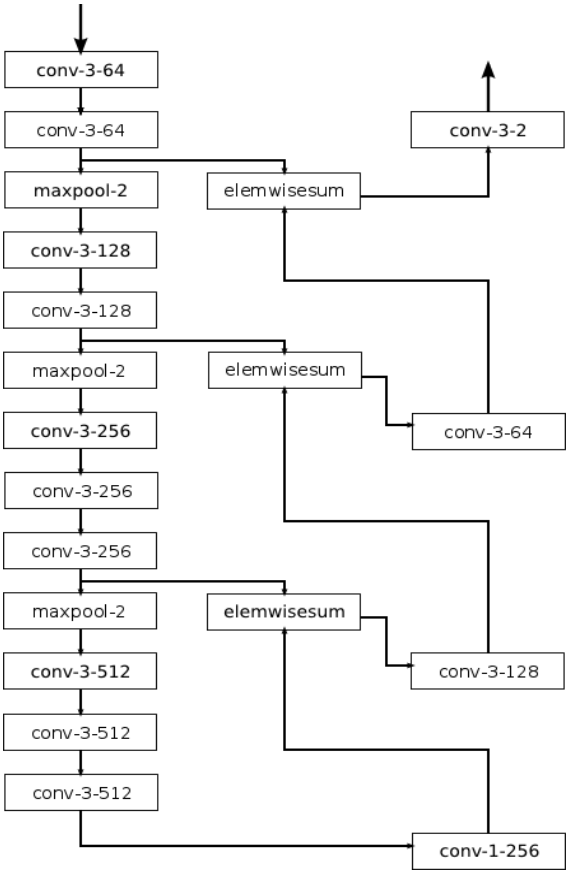

Figure 3: shows the structure of our Regression baseline model. [4]

In order to perform classification on continuous data, the targets U and V from the CIELUV color space values in interval [-100; 100] is discretized into 50 equi-width bins by applying a binning function to each input image prior to giving it as an input to the network. The function returns an array of the same shape as the original image with each U and V value mapped to some value in the interval [0; 49]. The network outputs two separate sets of the most probable bin numbers for the pixels, one for each channel. We used cross-entropy loss on the two channels. Talking about the architecture, it's almost similar to regression baseline model with some additional layers like a concatenation layer concat, is used which combines multiple intermediate feature maps to increase prediction quality, producing finer details and cleaner edges. The concatenation layer is followed by three 3 X 3 convolutional layers, which are in turn followed by the final two parallel 1 X 1 convolutional layers corresponding to the U and V channels. These 1 X 1 convolutional layer's act as the fully-connected layers to produce 50 class scores (bins) for each channel for each pixel of the image. The classes with the largest scores on each channel are then selected as the predicted bin numbers. Via an un-binning function, we then tried to convert the predicted bins back to numerical U and V values using the means of the selected bins.

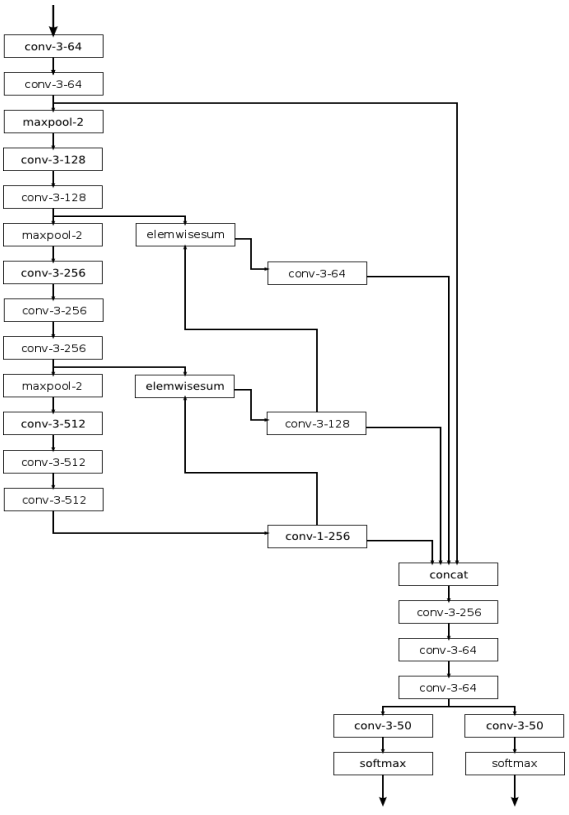

Figure 4: shows the structure of our Classification model [4]

The third model we used to do the colorization task is from the paper Fully automatic image colorization based on Convolutional Neural Network [5]. In their work, they have used the pretrained VGG-16 CNN model with slight modifications. This model incorporates a huge amount a semantic information, since it was already trained on a dataset which consists of more than 1 million images. The input of VGG-16 is a fixed-sized 224 X 224 RGB image. Since the input in our case is a single-channel grayscale image, we concatenated the input grayscale image one after another three times. We only used first eight layers from VGG-16 as was described in the paper. Then we upscaled and merged the layers to give 224 X 224 X 451 size matrix – named as T. This T matrix is followed by two convolutional layers. The outputs of these two convolutional layers were concatenated and this process results in a 224 X 224 X 144 -size matrix which is denoted by Q. The matrix Q is followed by two convolutional layers and the outputs of the second convolutional layer are the predicted U and V channels

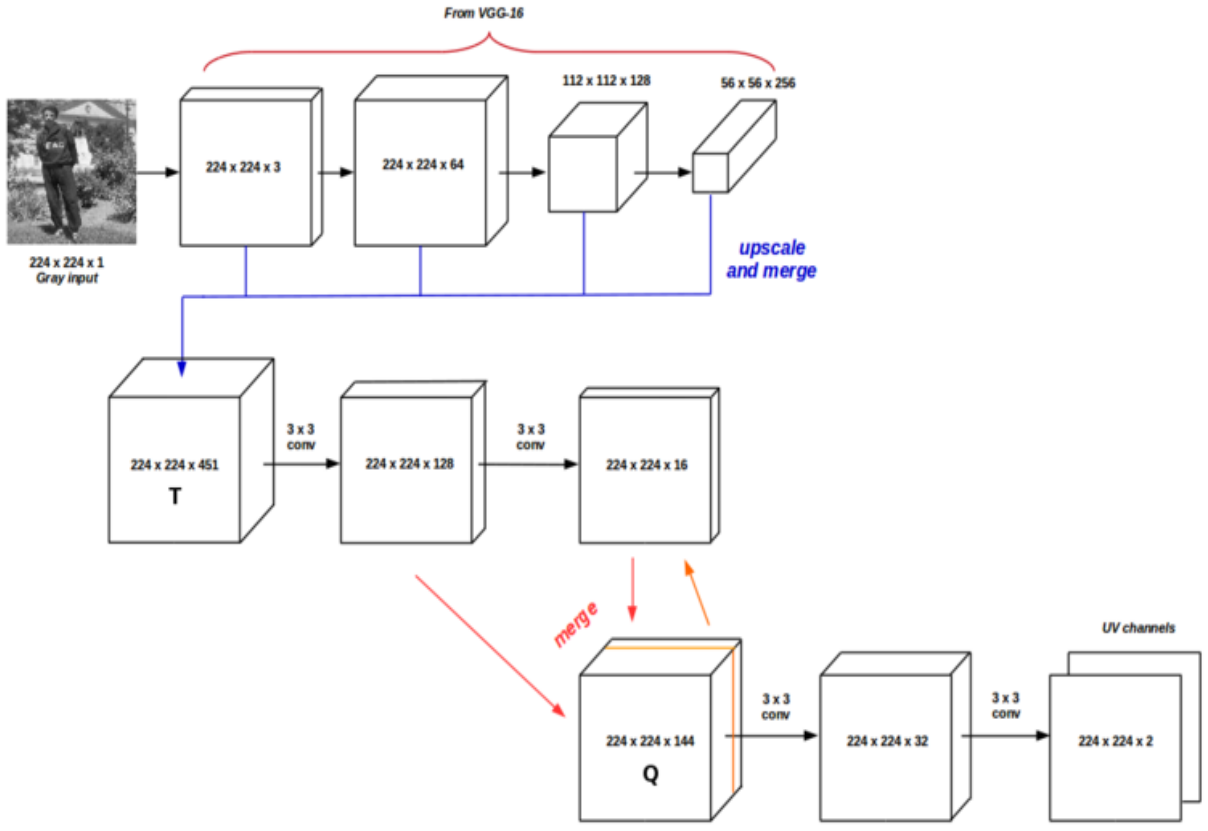

Figure 5: Architecture of Automatic Colorization model. [5]

5. Results

In Figure 6, in first row shows the actual colored images. The second row shows the grayscale images. The third row shows the images generated from our model. The first image in third row is the output of our regression model and second image is the output of our VGG16 CNN model.

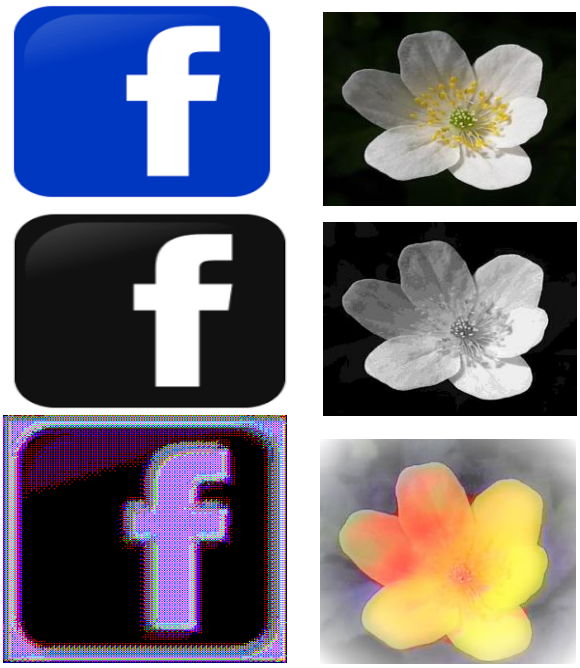

Figure 6: Outputs of our model

6. Conclusion

Our first approach has shown the efficiency of deep convolutional neural network to colorize black and white images. The process has shown how formulating a regression problem can produce colorized images that seems a ray of hope to colorize the image.

We also tried classification based method but somehow it didn't work for us. So, we tried a different approach to do this colorization task.

The second method used not only provides a useful graphics output, but can also be seen as a pretext task for representation learning. As the network learned representation to color images, it can also be applied for object classification, and detection. The second approach was based on fully automatic colorization algorithm based on VGG-16 and a two stage CNN architecture. Without pooling layers, two-stage CNN was trained by VGG-16 which provides multiple discriminative and sematic information. The two-stage CNN gave a stronger representation by adding information from the preceding layer. The prediction of U and V channels helped to add more color channel to the Luminance value.

7. References

[1] R. Dahl. Automatic colorization, 2016

[2] Cheng et al. [3] proposed a fully-automatic colorization

[3]https://blog.floydhub.com/static/function_equal_to_three_grids-5b6dd2436a04744fcfb0af8798fe1ca3-77609.png

[4] Jeff Hwang, Image Colorization with Deep CNN

[5] Domonkos Varga, Tamas Sziranyi: Fully automatic image colorization based on Convolutional Neural Network