Image Colorization (Summary)

When you have a grayscale image and you want to add colors to this image, to predict the color of every object (e.g. car, building, tree) is a very expensive and complex operation, you need to assign to every pixel a color, and it requires to have a lot of similar reference images that stored into a database to predict the color of each pixel in a grayscale image by matching between actual image (grayscale image) and reference image. Deep learning helps us to find techniques to colorize a grayscale image by using DNN (Deep Neural Networks) and CNN (Convolutional Neural Networks), and by having a very large database we can train our neutral network. When man sees a grayscale image and he detects an object such as a tree, he can colorize it in his mind, he has a semantic-awareness so he can colorize it and imagine it in his mind. So, the man have a lot of examples of that object, he compare between the actual image that he had seen before and the target image that he need to colorize, and build a model in his mind, but the problem occurs when he has no information about the object in the target image, so he **predicts**. So, when we build a model we should take care of the concept of semantic-awareness, our model should have that semantic-awareness to give us efficient and high accuracy output, and after the neutral network has trained we use it to colorize a grayscale image. But to make our model very effective, we should cluster (group) our reference images, so that images have similar shape, color, edges, etc. That process is called **adaptive image clustering**, our model will be trained for each cluster, and when we give it the target image. it finds the nearest cluster to start its process of colorization.

How can we learn a neutral network to convert a grayscale image to a color image?
In Deep Colorization paper that written by Zezhou Cheng, Qingxiong Yang and Bin Sheng, says that they used a regular deep learning technique to train the Neural Network model by providing a lot of exemplars to mapping

between the target image and reference image, and they use the YUV space color to minimize the correlation between the U, V axis, so that means that there is no strong relationship between the two, U and V axis.

$$Cp = F(\Theta, Xp)$$

Where Cp are the corresponding chrominance values, and Xp is the feature descriptor extracted at pixel p. Θ are the parameters of the mapping function F to be learned from Λ.

Why do they use feature descriptor? To minimize the unnecessary information in the image so they can extract the necessary pixels in the image, what makes the operation goes easily to the training model. And they used different feature descriptors for tests and to find the most effective and most accurate one that produces an acceptable result to human eyes. [1] Figure 9, 3.

[1] references a limitation of their colorization technique, It depends on a huge set of reference images in the database which contains all possible objects and that is actually impossible. And another problem, when you convert a color image to grayscale image for training, there is a loss of data and that loss can’t be recovered again.

I find that [2] has no difference in colorized image, [2] doesn’t depend on semantic-awareness as we discussed before because it is not always true, and they use different architecture and different color space to colorize images with different objective function to map between two images, grayscale and color image.

REFERENCE

1. Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 415–423
2. Zhang, R., Phillip, I., Alexei, E.: Colorful Image Colorization. In: European Conference on Computer Vision. (2016) 649−666
3. Aishwarya, S.: Feature Engineering for Images: A Valuable Introduction to the HOG Feature Descriptor. In: shorturl.at/jwH12