Project - Colorizer

colorization_deploy_v2.prototxt *(Caffe Deep learning framework)*

This file describes the architecture or structure of the neural network. It is like a blueprint of how the neural networks are connected.

colorization_release_v2.caffemodel

This file contains the trained weights of the neural network. After the was trained on million of color images, these weights represent the patterns and features the network
learned to predict colors. This is the "Knowledge" the model has.

pts_in_hull.npy

This file contains specific data points(called cluster centers) related to how the model represent colors internally.The model doesn't predict raw 'ab' values directly but rather
work with a classification over these predefined color points. This file helps convert the model's output back into smooth 'ab' color values. It's a specific detail of this particular
mode's design.

The deep learning model being used here is a convolutional neural network model developed by researchers Zhang, et al for image colorization.

We are loading precomputed cluster centers, The ***pts_in_hull.npy*** contains an array where each element is a numerical representation of a specific color captured in **ab** space.

AB space

The **ab** space is a part of LAB color space. The LAB color space is designed to approximate human vision and is composed of three components:

- ***Lightness:*** This channel represents the brightness of the color
- ***a:*** The axis that turns from green to red
- ***b:*** The axis that turns from blue to yellow

We decouple the ***Luminance(L)*** and ***chrominance(a and b)***. This separation helps models generate vibrant colors without being skewed by brightness values, which are varied across images.

How were these colors extracted from training data

1. Conversion: Convert the training images from their usual color space (like RGB) to LAB color space. This gives you access to L, a and b
components.

2. Clustering: Extract a and b values from all the pixels in your training set. With these a and b values as points in a 2D plane, we clustering algorithms(k-means) to find groups of similar colors. Each group will have a central point which best represents that group.

3. Storing the palette: These central points are often saved (.npy) as your palette of common colors. During the model inference, the model can
reference these cluster centers to narrow its color predictions to realistic colors.