EXPLORING COLORIZATION WITH NEURAL NETWORKS

ARTICLE EXPLORATION REPORT

CONCEPTS & MAIN STEPS

Grayscale images are colorized by conversion into Lab colorspace instead of RGB. Lab represents colour information in 3 channels, L: Lightness, a: Green to Red, b: Blue to Yellow. This is helpful as the Grayscale image can be used for the L channel and we only must predict 2 channels. Also, our eyes are more sensitive towards brightness.

Instead of traditional Neural Networks, convolutional filters are used, initially we use many filters and then we narrow it down to 2 channels. The alpha version implemented is trained and tested on the same image and is therefore not good at generalization as it simply memorizes the data.

We try to fix this in the beta version. The architecture of the neural network here is Feature Extraction from the Input Image and then Colorization from Feature Extraction. Hence, we first keep downsizing the image for Feature Extraction and then we upsize it for Colorization.

In standard classification neural networks pooling layers are often used to increase information density. However, they distort the image and therefore for image colorization we instead use stride=2 which reduces image size without introducing distortion. Upsampling layers and utilizing padding = 'same' ensure that the image size remains the same.

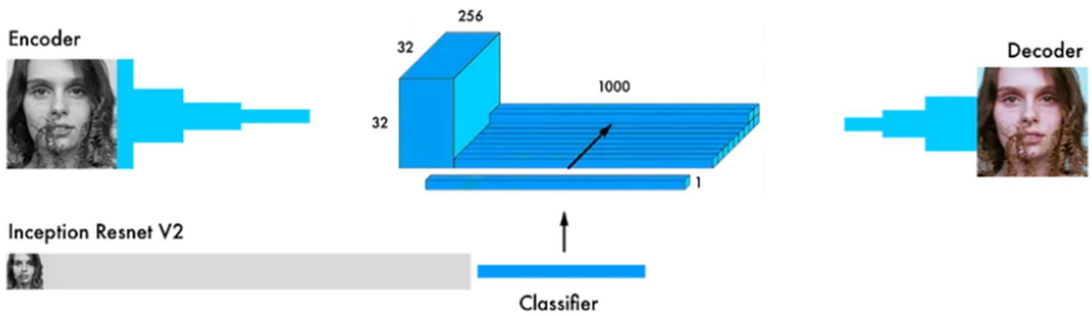
The issue with this implementation is that the model is unable to distinguish between objects and ends up colouring every image brown and is incapable of generating nuanced colours.

This is resolved in the full implementation which parallelly uses the Inception ResNetV2 classifier to help the decoder. A Fusion component takes in the encoder and classifier output and feeds it into the decoder for a more sophisticated output.

NEURAL NETWORK ARCHITECTURE & LOSS FUNCTION

The final implementation has 4 distinct components,

1. Encoder: Facilitates Feature Extraction from input image by downsizing.
2. Inception Resnet V2: A classifier which helps in a nuanced colorization instead of just brown.
3. Fusion Layer: This aids in combining the output of the classifier and the encoder which is fed into the Decoder.
4. Decoder: Generates the 2 channels 'a' and 'b' for colorization from the output of the fusion layer by upsizing.



This architecture is called ‘Fusion Layer Architecture’.

The loss function used here is MSE or Mean Squared Error.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Limitations & Potential Solutions

1. Limited Colour Information in Grayscale Images: The network needs to infer colours based on context and prior knowledge. Incorporating additional information such as the classification result from Inception Resnet V2 might alleviate this issue.
2. Quality and Diversity of the Training Dataset: This can significantly impact the model's ability to generalize to various colorization tasks. Thus, curating diverse and high-quality training datasets is essential.
3. Memory and Computational Cost: Deep neural networks are inherently resource intensive. Employing architectures with downsampling and upsampling layers, optimizing model efficiency, and using techniques like transfer learning can help manage computational constraints.