

## Problem Statement

Our project proposes to colourize black and white photos.

## Data Preprocessing

Our dataset will be from the following Kaggle:

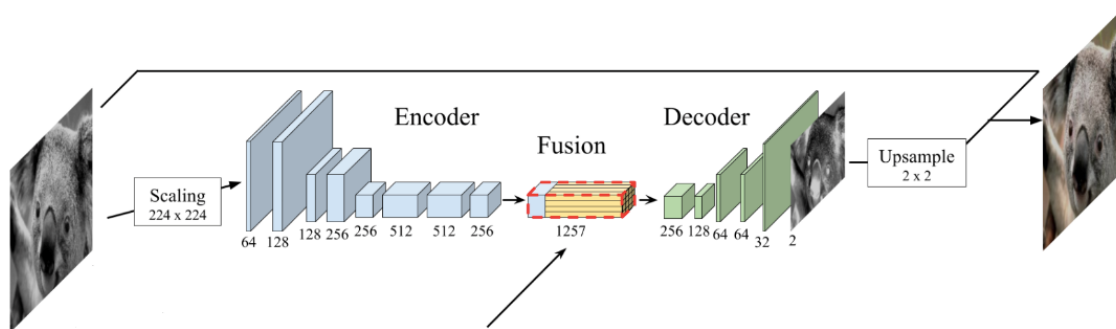
- 1) <https://www.kaggle.com/datasets/shravankumar9892/image-colorization>

This dataset contains 25000 images in the Lab colour space represented as NumPy arrays. Each image was resized to be 224 x 224 pixels in dimension. The training input (L) and training output (ab) vectors were extracted and stored in separate arrays. No further preprocessing was required on our end. However, for future reference, should we need to expand the dataset, these parameters (colour space and dimension) will be kept constant.

## Machine Learning Model

Our chosen model remains to be a convolutional neural network (CNN). We are following the architecture for an autoencoder, which is identified by the coupling of an encoder and decoder. To implement this model, we are using the following libraries:

- File handling:
  - o Numpy: LAB data stored in NumPy arrays with shape (# images, 224, 224, 2)
    - Where the dimensions of our images are 224x224 pixels, and 2 represents the a and b components of the LAB colour space
  - o Matplotlib: for visualization
- Model implementation:
  - o Tensorflow: machine learning platform that allows us to build our model with
  - o Keras: API for TensorFlow used to create our model
  - o Keras.layers: create the convolutional layers to encode and decode our images
  - o Keras.models: allow us to stack convolutional layers
    - A tensor will be inputted, and a tensor will be outputted
  - o Keras.preprocessing: allow us
  - o ImageDataGenerator: turns image information into tensor information to be inputted into the model
- Layers
  - o We are going to go with 8 convolutional layers for the encoding process, passing it through a ReLU function
  - o We will be using 5 convolutional layers for the decoding process, with all layers but the last using the ReLU function
    - We will be using the tanh function on the last layers to ensure that the output will be a value between -1/1



## Preliminary Results

Following ([Image Colorization - Algoritma Data Science School \(algoritmaonline.com\)](https://www.algoritmaonline.com/)), we are using this existing code (as described above) as a starting point in the implementation of our own model. In our exploration, we learned how numerical data can be transformed into an image and vice versa, how encoders and decoders come together to form our model, and how to take the model's output to print an image.

- Go into our evaluation metrics
  - o Will probably perform poorly, look into why that is the case and work from there

## Next Steps

- Figuring out the number of convolutional layers for encoding and decoding that will allow for the best optimization of our model
-