

# Image Colourizer

Kate Allsebrook  
101142664

## Abstract

I am creating a program that uses a machine learning model to take a black and white image and predict the rgb values of each pixel in the image, and output the coloured image.



Image courtesy of [hotpot.ai](https://hotpot.ai)

## Methods

**Dataset:** For this project, I will be using [this](#) dataset on Kaggle, which is a set of black and white (and corresponding colour) 200x200 images. I may test on some of my own images as well.

## Preprocessing

### 1. Gaussian Filter

My first step will be running all the input/black and white images through a gaussian filter, in order to smooth out some noise that can make accurate predictions more difficult.

### 2. Convert Ground Truth Images

Next I will convert the ground-truth (colour) training images into a tensor with dimensions  $n \times 200 \times 200 \times 3$ , where each entry has the red, green, and blue values of each pixel.

### 3. CNN

Then, I'll create a convolutional neural network with some fully connected layers at the end. The output will be a tensor with the same dimensions as the ground-truth training images, for easy performance testing. In creating this project I will choose the hyperparameters with the results of the model on a validation set of images. Then, using the selected hyperparameters, I'll test my model's performance on a hold-out test set.

## **Output**

### **4. Render Image**

I will convert the output/predicted tensor into a full colour image to be displayed.

## **Technologies To Be Used**

- Python
- Tensorflow
- Keras
- OpenCV

## **Anticipated Results**

I anticipate that the model will work better on landscape images rather than people, since the colours follow more predictable patterns (ie. grass is green) than people (ie. dyed hair). If I find that the model's performance is severely worsened by the inclusion of images including people, I will remove them and create a landscape-only colourizer.