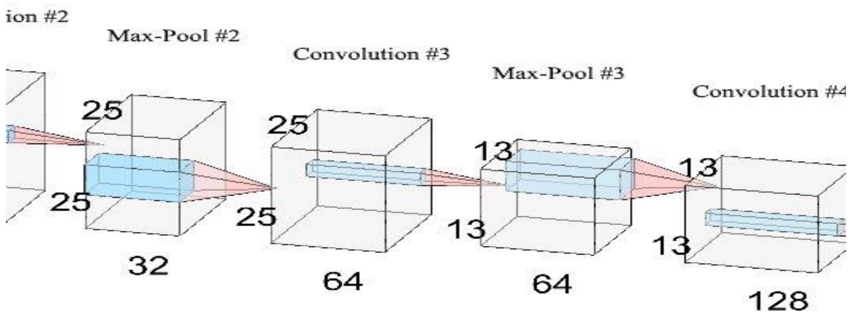
Colorization of old black and white images

Naveen Pratap (11804642)
Lovely Professional University (BTech. CSE)

Aman Kumar Upadhyay (11804635)
Lovely Professional University (BTech. CSE)
Project link

ABSTRACT : In this paper colorization of old black and white images using artificial neural networks and keras is proposed. The model takes old black and white images as input and ouputs them as colored images.

CONVOLUTIONAL NEURAL NETWORKS : Convolutional Neural Networks, or commonly known as CNNs, are the product of Artificial Neural Networks and convolution set of operations combined together.



It is a much advanced version of Neural Networks, with high efficiency and has proved its usefulness in image related problems. Artificial Neural Networks are composed of artificial neurons which stimulate biological neurons in a limited way. Let us have a set of elements, namely M M = {m1, m2, …,mn} …(1)

and set of input itights, namely Wt respectively
Wt = {wt1, wt2, …..,wtn} …(2)

and an input bias . Thus the output is represented as

N = f(∑i (mi * wti) + bias) …(3)

Thus it have a single output for a series of inputs. This concept of Artificial Neural Network is used in Convolutional Neural Networks as convolution operation. When an image is given as input, it apply some mask or filter on it, to obtain the desired output. Every image is made up of pixels, that is, some numeric values. Now these masks, of a very small size, are moved on the image such that every pixel becomes an input to these masks. [9] Fig 3: Pictorial representation of Convolution Neural Networks The input part of the image, say

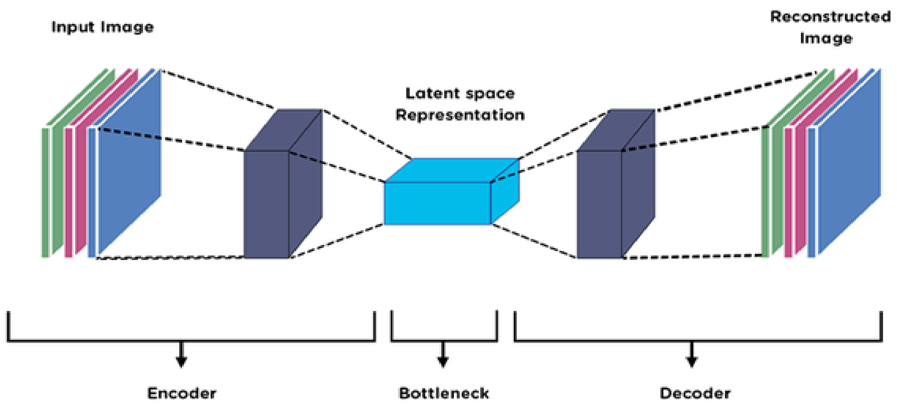
In(0,0), In(0,1), In(0,2) In(1,0), In(1,1), In(1,2) In(2,0), In(2,1), In(2,2) …(4)

Is masked on with the values of the mask or the filter, and the final output is a single value given by

N = M1 * In(0,0) +M2 * In(0,1) +M3 * In(0,2) + M4 * In(1,0) +M5 * In(1,1) +M6 * In(1,2) + M7 * In(2,0) +M8 * In(2,1) +M9 * In(2,2) …(5)

Thus the masked value at point I on the image is replaced by Z in the new image. By keep on implementing masks or filters on the image, the models figures out the different features of the image, be it basic features like lines, shapes etc., or advanced ones like eyes, ears and so on. This becomes the base of the Convolutional Neural Networks, one of the most widely used techniques in Deep Learning or Advanced Machine Learning. With the help of CNNs, various researches are carried out solving various image problems. This project also uses CNNs as the base of both the models. A convolution 2D layer of Keras was taken into consideration to downsize the image and extract important features, thus to optimizing the colorization of the greyscale images.

AUTO ENCODERS : Auto encoders are neural networks that provide easy entries to understand and comprehend more complex concepts in machine learning. Auto encoders give us the output with same values as the input, after applying a series of operations on the data.



Mathematically, let us say, Input data is Ai , Compressed data is Bi , And the output data is Ai `, Then it can say that,
B i = (Itight1*A i) + bias1 …(6) A i ` = (Itight2*B i) + bias2 …(7)

Our aim is to have Ai ` and Ai as similar as possible, without much loss in the data, it can use the following objective function,
Q(Wt1 , Bias1 , Wt2 , Bias2) = ∑n x=1 (Ai ` - A i) 2 = ∑ n x=1 (Wt2 *Bi + Bias2 -A i) 2 =∑n x=1(Wt2 ((Wt1 *Ai) + Bias1) + Bias2 - A i) 2 …(8)

Auto encoders have proved their usefulness in areas like dimensionality reduction of images. Once it have a more condensed representation of a multi-dimensional data, it can easily visualize it and do further analysis of it. It can also be used in classification, anomaly detection and so on. This project uses the techniques of stacked up auto encoders which parse the features into small encodings that are then decoded using the decoder unit. This reduces the dimensionality and helps in learning the features in an unsupervised manner, hence making it easier in the colorization process.

RECTIFIED LINEAR UNIT :

Rectified Linear Unit, commonly known as ReLU, is an activation function. An activation function defines the output for a set of given inputs. Rectified Linear Units commonly defines the output as linear with slope 1 if the input is greater than 0, rest 0. [5]

F(x) = { mx+c x>=0 0 x<0 } …(9)

ReLU is one of the most commonly used activation function in Machine Learning or Deep Learning. Mathematically, it can show ReLU with deep learning as: [4]
Y(ф) = - ∑ g * ln (maximum (0, фc + d)) …(10)

Let the input c be replaced by penultimate activation output
u, ∆Y(ф)∕∆u = (ф*g)∕((maximum (0, фc + d)) * ln10) …(11)

This project uses Rectified Linear Unit as an activation function between layers of the model. The output based on ReLU on one layer becomes the input for the next layer, and so on. Thus it increases the efficiency of the model, with lesser loss.

Model and Architecture :

It works on the principle of Convolution Neural Networks with Auto encoders. The Model is the first approach towards colorization of greyscale images. It uses (5,5) filter in the initial layer of the model, besides working on the principles of CNN.

Dataset Used: The model is trained on the Flower Dataset. The dataset contains around 10,000 images of various flower species. It provides us with high variety of images to get optimized results and minimum error.

Optimizer: The optimizer used in the Model is Adaptive Moment Optimization, or commonly known as Adam. Adam Optimizer combines the heuristics or gradient descent with momentum algorithms and Root Mean Square Propagation.
P t = (Ω1 * Pt-1) – (1- Ω1)*Gtt …(12) Q t = (Ω2 * Qt-1) – (1- Ω2)*Gtt*2 …(13)

Where, P t : Exponential Average of Gradients Q t : Exponential Average of Gradient Squares Gtt : Gradient at time t Ω: Hyper parameters Exponential Average of Gradients, that is, Pt can also be written as:
P t = (1- Ω2) ∑ n x=1 Ω2 t-x * Gtt*2 …(14)

Hence the expected value of the exponential moving average at time t is [3]:
Exp[Pt] = Exp [(1- Ω2) ∑ n x=1 Ω2 t-x * Gtt*2] = Exp [Gtt*2]*(1- Ω2) ∑ n x=1 Ω2 t-x +c = Exp [Gtt*2]*(1- Ω2) + c …(15)

Thus Adam showcases promising results with the dataset by increasing the efficiency in colorizing them into RGB format.

Architecture: The Model uses stacked up auto encoders for converting greyscale images into coloured ones. Based on the mechanisms of Convolutional Neural Networks, it also includes dropouts to introduce noise, thus to prevent overfitting. It also includes initial three convolution layers, followed by an up sampling layer, then six convolution layers and again an up sampling layer. In the end it has three more convolution layers before the output layer. Loss: The Model inures a loss of about 0.0415 and a value loss of about 0.0388. Thus it shows that using these parameters, as used in the model, the loss between the final output images as compared to the input image, was low.