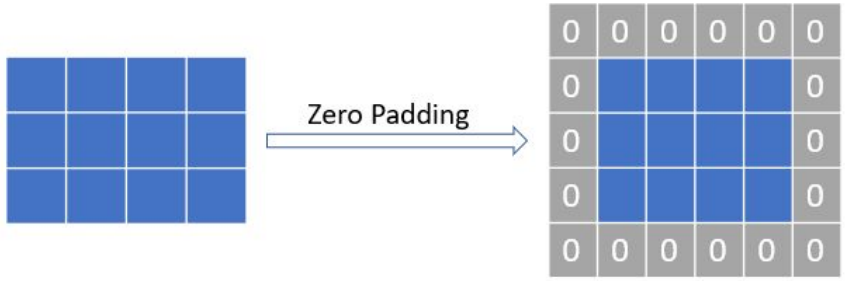
CS 520: Assignment 4
Image Colorizer

197001190: Aditya Lakra (al1247)
196003272: Hari Priya (hp467)
197001859: Twisha Naik (tn268)

1. *Representing the Process: How can you represent the coloring process in a way that a computer can handle? What spaces are you mapping between? What maps do you want to consider? Note that mapping from a single grayscale value gray to a corresponding color (r,g,b) on a pixel by pixel basis, you do not have enough information in a single gray value to reconstruct the correct color (usually).*

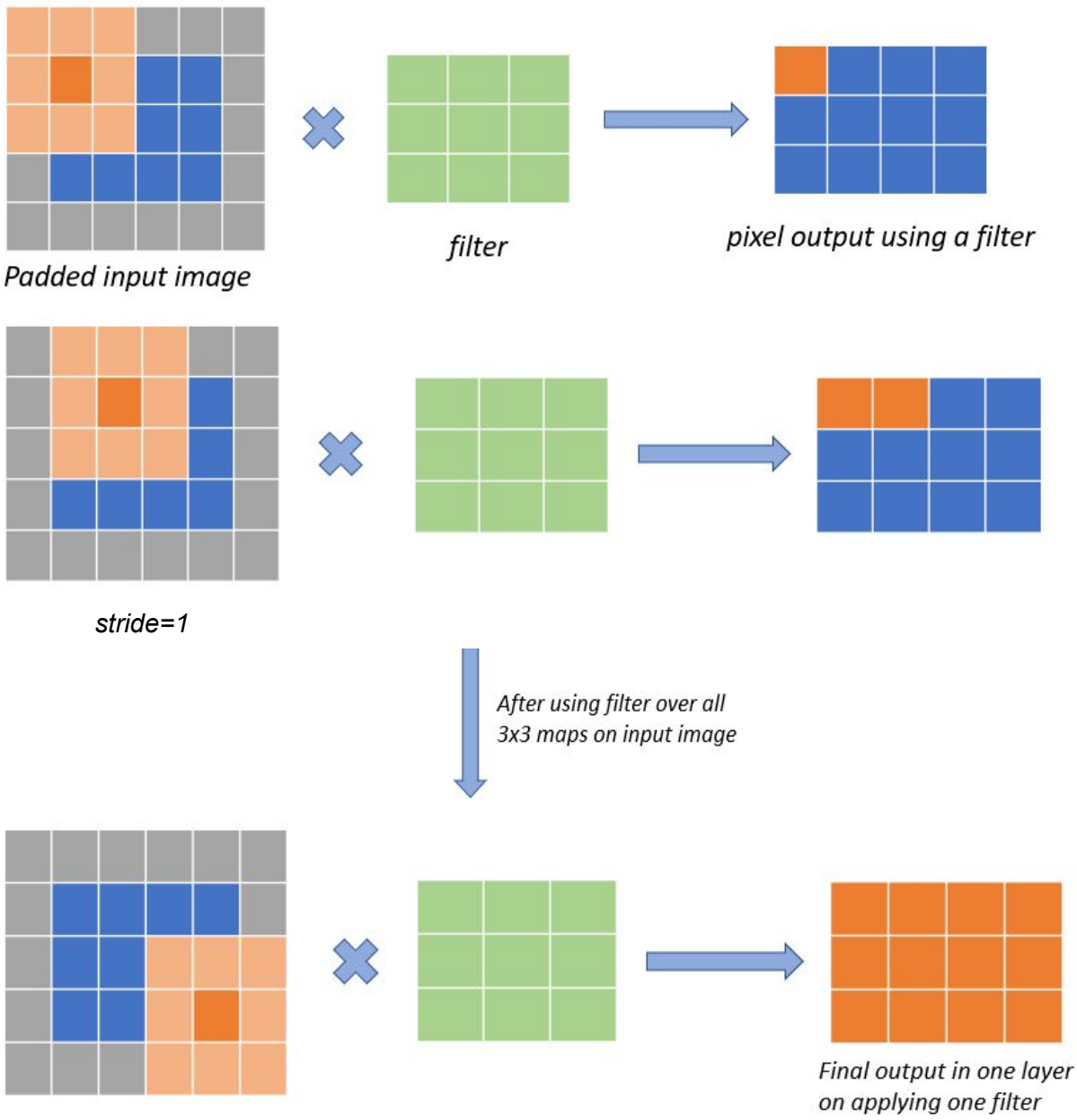
We built a Convolutional Neural Network (CNN) to handle the coloring process. The CNN has 3 layers with 3 filters in each layer.

The input to the first layer will be a grayscale image which is a two-dimensional matrix of pixel values. Each filter is of 3x3 dimension. To have enough information for mapping a grayscale pixel value to a colored one, we consider the immediate neighbors of each pixel. We add a zero padding to the input image to prevent the loss of any information and retain the dimension.

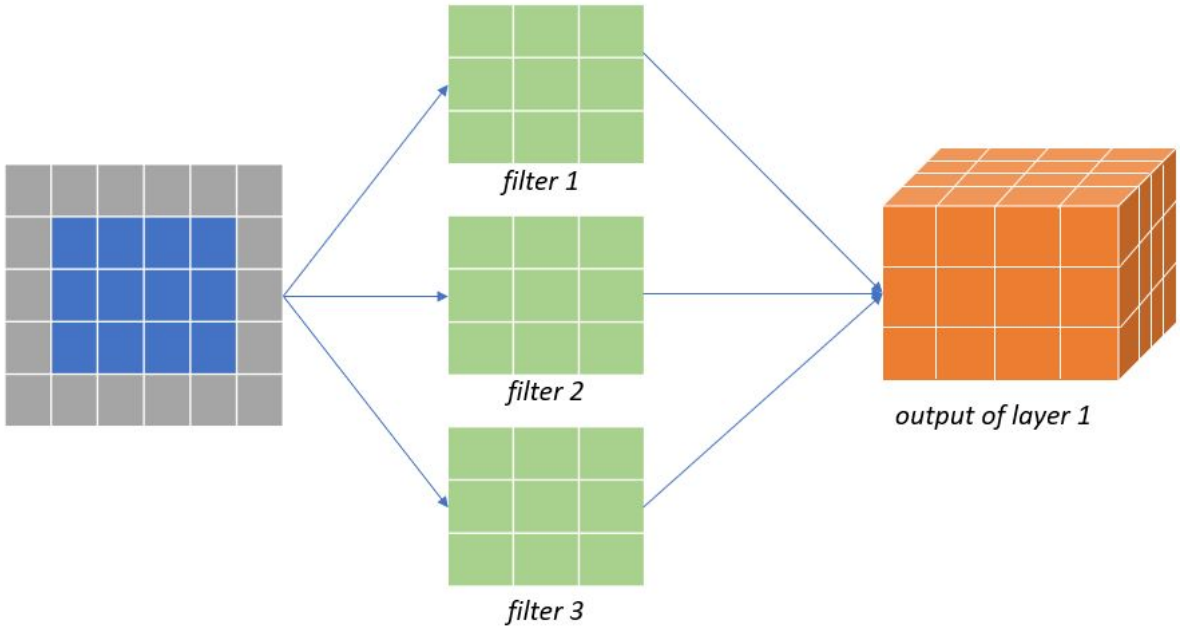


input grayscale image

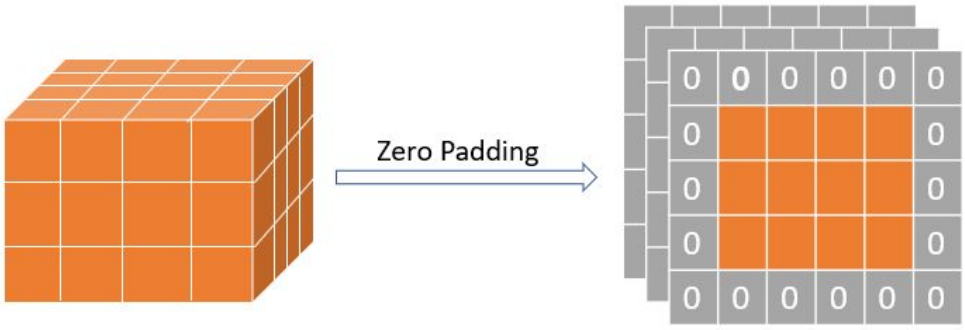
The filters are then applied to all the 3x3 maps of the padded image with an implicit stride of 1.



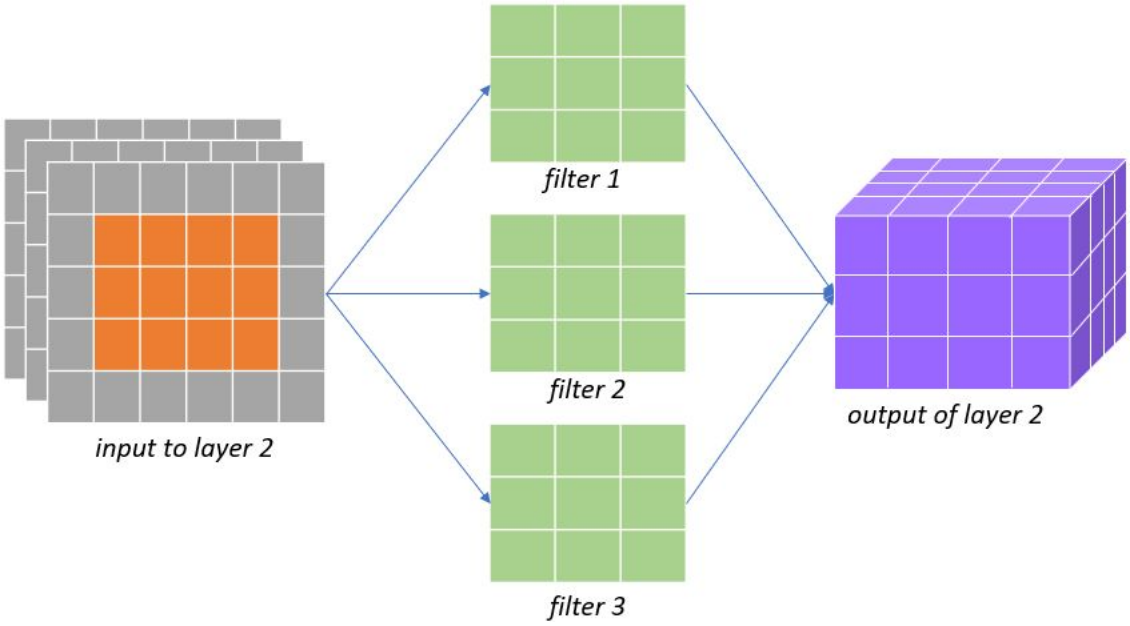
We apply all the 3 filters in each layer as depicted above. Hence, if the input image is of dimension mxn, each layer gives a 3D output of mxnx3 dimension.



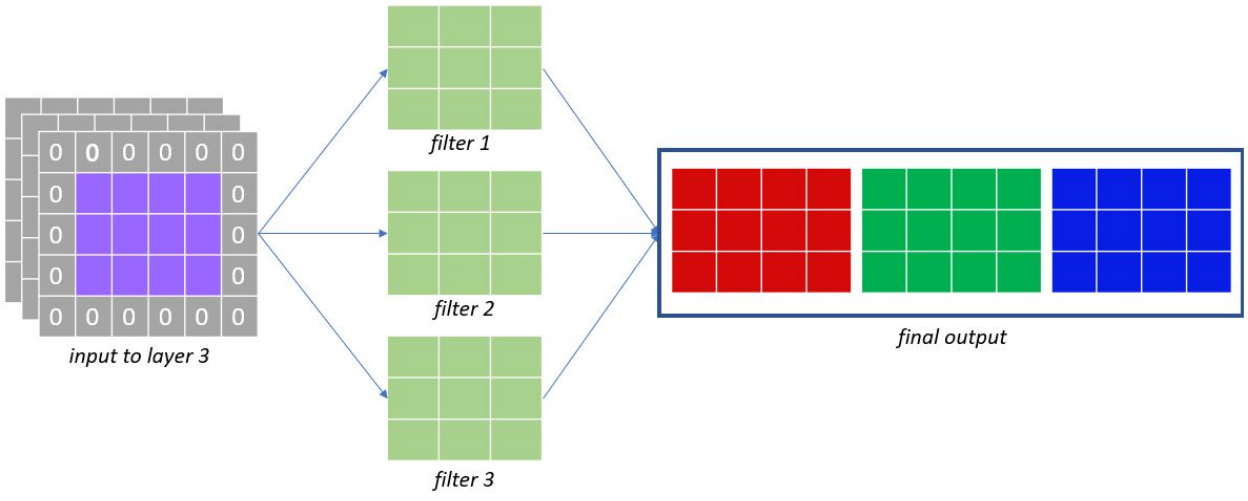
The output of each layer is again zero padded to prevent loss of information in the convolutions.



In the second layer, we again apply 3 filters of 3x3 dimension to all three outputs of the first layer and get an output of dimension mxnx3.



We apply zero padding to the output of second layer, pass it as input to the third layer and repeat the same process with this layer. The final output will be of dimension mxnx3 in which each mxn output will be a matrix of R, G, B pixel values corresponding to the grayscale value of each pixel of the input image.

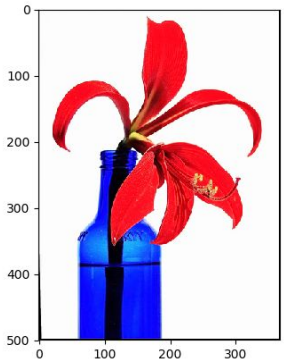


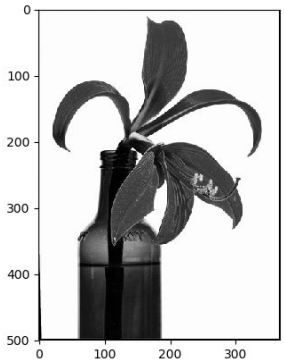
The RGB values from the output of the last layer are used to construct the color image for the input grayscale image.

2. *Data: Where are you getting your data from to train/build your model? What kind of pre-processing might you consider doing?*

We get colored images from **xyz** dataset and convert these into grayscale images using the given color to gray conversion formula:
Gray(r,g,b) = 0.21r + 0.72g + 0.07b.

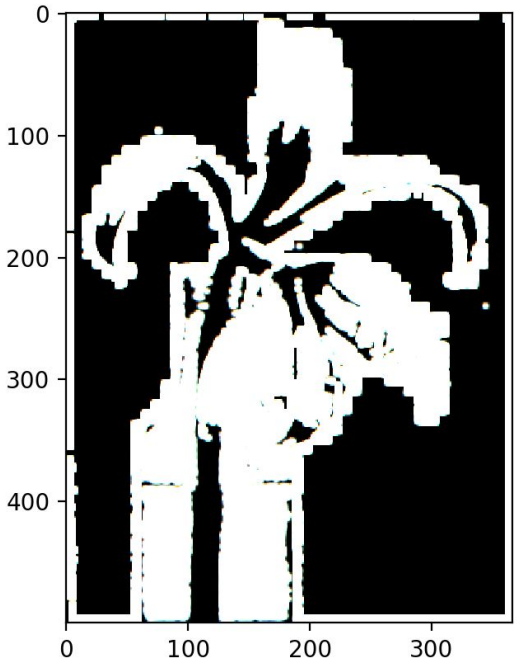
We use these grayscale images as input to train the model we build and then for colorization using the trained model.
Below is a sample colored image which we took and converted into a grayscale image.



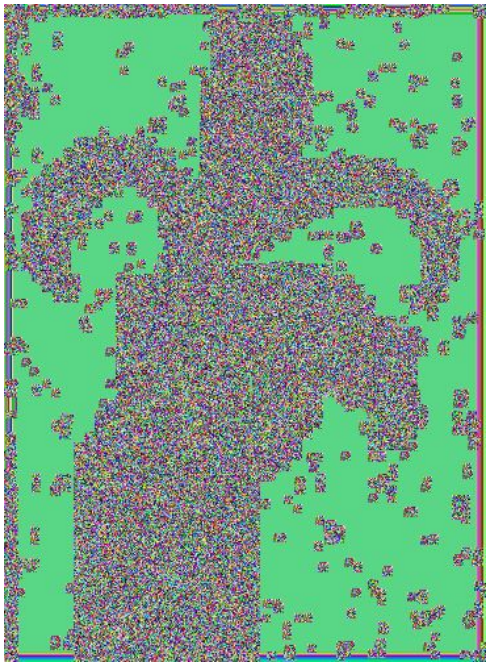


Results:

1. Output using the model where no activation is applied.



2. Output using the model with sigmoid activation is applied.



3. Output using the model with ReLU activation is applied.

3. *Evaluating the Model: Given a model for moving from grayscale images to color images (whatever spaces you are mapping between), how can you evaluate how good your model is? How can you assess the error of your model (hopefully in a way that can be learned from)? Note there are at least two things to consider when thinking about the error in this situation: numerical/quantified error (in terms of deviation between predicted and actual) and perceptual error (how good do humans find the result of your program).*

We are using regression to predict (R, G, B) values of the colored image.

The loss function used is the difference of the predicted values as compared to the original values.

We are trying to minimize the difference between the loss values.

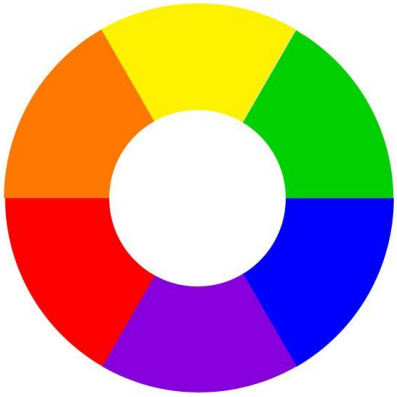
4. *Training the Model: Representing the problem is one thing, but can you train your model in a computationally tractable manner? What algorithms did you draw on? How did you determine convergence? How did you avoid overfitting?*

We have used basic gradient descent algorithm to train the model weights.

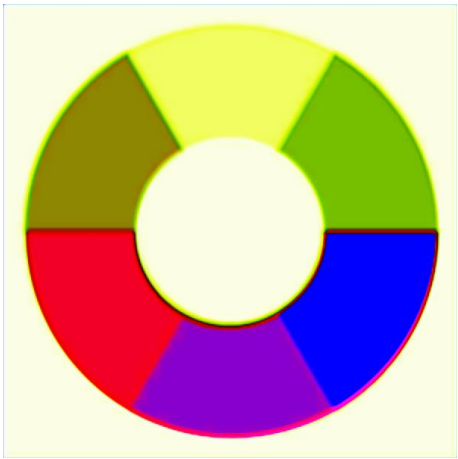
The learning takes place using the backpropagation algorithm. We start from the output layer and then backpropagate the loss till the input layer.

The best result we got so far (trained only on a single image):

Original Image:



Colorized Image:



We are still working to make the model better.

Currently, as a sanity check we trained our model on a single image and used the same image for prediction.

We are working to make the model generic which is being trained on 50 diverse images and then giving it a completely new image for prediction.