VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

August 2022

Colourization of Grayscale Images Using Deep Learning

A Project Report

Under the Guidance of

Prof. B. Gladys Gnana Kiruba

By

19BCB0039– Anirudh A
19BCE0504- Anoushka Dalal
19BCE2285- Gunik Luthra

DECLARATION BY THE CANDIDATE

We hereby declare that the project report entitled **"Colourization of Grayscale Images Using Deep Learning"** submitted by us to Vellore Institute of Technology, Vellore in partial fulfilment of the requirement for the award of the degree of **B. Tech (Computer Science and Engineering)** is a record of J- component of project work carried out by us under the guidance of **Prof. B.Gladys Gnana Kiruba**. We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore Institute of Technology, Vellore.
Date : 16th August 2022

Signature of the faculty Signature of the Candidate

TABLE OF CONTENTS

1. Introduction

1.1 Abstract

Colourization is a procedure of adding shades of color to a monochrome picture or film. The procedure includes typically fragmenting pictures into areas and following these districts crosswise over picture successions. It requires extensive client mediation and is a monotonous, tedious and costly assignment. For a considerable length of time, numerous film makers restricted colorizing their high contrast motion pictures and thought of it as vandalism of their craft. The innovation itself has moved from meticulous hand colourization to the present to a great extent, robotized strategy. We aim to use deep learning technology and convoluted neural network to develop a code that can predict the colours of a monochrome image and colour it for us. This can be used to not only colour old school films and pictures but also improve the quality of face detection in biometric machines.

Keywords: Colorization, monochrome image, convoluted neural network, deep learning

1.2 Background

For colouring a gray-scale image, we first need to understand what the different colour spaces are and the difference between a coloured image and a grey-scale image. Colour images are just a collection of information about the intensity at a particular point. So, for a greyscale image, the only information required to make the whole image is the intensity of the pixel that can vary from 0 to 255. As each pixel needs to store the information about one parameter, the pixel size is normally 8 bits, as only one value is stored. For a coloured image, more information needs to be handled. Now along with the intensity, the colour information for each pixel also needs to be stored. For an RGB colour space, this involves storing the information about the Red, Green and Blue colours of the pixel. These are the three main primary colours that can be mixed in different proportions to obtain all the visible colours. Now as three pieces of information are needed to be stored, and a single information takes 1 byte of space (8 bits), where the value can again range from 0 to 255, a single pixel in a coloured image stores 3 bytes (24 bits) of information. Converting a grayscale image into a coloured image involves generating these 3 pieces of information (depending on the colour space chosen), from the given single byte of information in one pixel of a grayscale image. Generating these values accurately has been a difficult problem over the years as different conditions need to be taken into account and many a times, some details are always left behind. The actual ab color space is not discrete, it is continuous, and for a CNN model to train in such data would have taken a very long time. Nevertheless, as we have made it to be divided into 313 spaces, the training can be done in a standard laptop itself. There are two popular methods for colorization : one in which a colour is assigned to an individual pixel based on its intensity as learned from a colour image with similar content, and another in which the image is segmented into regions, each of which are then assigned a single hue. Assigning colour to a grayscale image is a difficult problem. Given a certain image, there is often no “correct” attainable colour. The applications of such a method allow for a new appreciation of old, black and white photographs and cinema, along with allowing better interpretation of modern grayscale images such as those from CCTV cameras, astronomical photography, or electron microscopy.

2. Overview and Planning

2.1 Proposed work

We propose an image colourization technique using deep learning algorithms. The main objective is to create a convoluted neural network to predict the colors of a monochrome image given to us as the input and to produce a colored image. Different layers of the CNN model are used in order for this colorization. The input for the model will be grayscale images while the output will be the colorized images. The true ab colour space is continuous, not discrete, and training a CNN model with such data would have taken a lengthy time. We have separated it into 313 areas, which can be completed on a regular laptop. Convolutional neural networks are multilayer perceptrons that are biologically inspired and meant to replicate visual brain function. By using the high spatially local correlation observed in natural pictures, these models circumvent the restrictions of the MLP architecture.

2.2 Software Requirements

Kaggle API

2.3 Hardware Requirements

PC with a modern processor

3. Literature Survey and Review

3.1 Literature Summary

End-to-End Conditional GAN-based Architectures for Image Colorization

Mingming He, Dongdong Chen, Jing Liao, Pedro V. Sander, Lu Yuan

Input:

The user gives a grayscale image as input to the model. For training the model, training examples generated from ImageNet dataset containing 50,000 RGB images were used.

Output:

The model produced colorized images as output with accuracy of 89% and Peak Signal-to-Noise Ratio(PSNR) value of 26.77 dB.

Methodology:

They proposed a mapping convolutional model trained using adversarial methodology with conditional GANs using pix2pix framework and used a novel generator-discriminator setting that adapts the IBN paradigm to encoder-decoder architecture. After this they used Spectral Normalization for improving the generalization of adversarial colorization and used multi scale discriminators for getting improved color generation in small areas and boosted details.

Advantages:

It takes into account the adversarial loss during training the GAN network. It achieves good training stability. This paper shows that by boosting the performance of adversarial framework, reduction of desaturation effect can be achieved.

Disadvantages:

It is a very complex method that requires frequent rebalancing of weights, and the high instability during training when a GAN deals with high resolution images can lead the pix2pix framework to collapse. This reduces the contribution of adversarial loss.

Deep Exemplar based Colorization

Marc Górriz, Marta Mrak, Alan F. Smeaton, Noel E. O'Connor

Input:

The user provides a grayscale image to be colorized. Users can also give some reference images to enhance colorization. They used a training dataset based on ImageNet dataset by sampling from 7 categories: animals (15%), plants (15%), people (20%), scenery (25%), food (5%), transportation (15%) and artifacts (5%).

Output:

They generated colorized images from grayscale images with a top 5 class accuracy of 85.94%.and achieved PSNR of 25.50 dB

Methodology:

They chose the first CNN to directly select, propagate and predict colors from an aligned reference for a greyscale image. First the Similarity sub-net is a preprocessing step that provides the input by measuring the semantic similarity between reference and target using VGG-19 network. Then the Colorization sub-net provides a more general colorization solution for either similar or dissimilar pixel pairs. This employs multi-task learning which share the same network but 2 different loss functions: Chrominance Loss and Perceptual Loss. This ensures proper colorization from large-scale data.

Advantages:

1. A significant advantage of their network is the robustness to reference selection when compared with traditional exemplar-based Colorization

2. Their method benefits from getting references and hence is able to work on unseen images just as effectively and doesn’t fail like previous learning based methods trained on natural images.

3. They achieved a Fooling Rate of 38.08 % .

Disadvantages:

The quality of the final results depends on the choice of the reference samples. Also the perceptual loss based on the classification network (VGG) cannot penalize incorrect colors in regions with less semantic importance.

Collaborative Image and Object Level Features for Image Colorization

Rita Pucci, Christian Micheloni, Niki Martinel

Input:

The user provides a greyscale image to be colorized. For training their model, they used 3 datasets: ImageNet containing 10k images, COCOStuff containing 5k images, and Places205 containing 20500 validation images.

Output:

They successfully colorized the greyscale images with an increase in PSNR of more than 10% as compared to the existing approaches.

Proposed Methodology:

They proposed a single network called UCapsNet that considers the image level features obtained through convolutions and entity level features captures by capsules, then they enforced collaboration between such convolutional and entity factors to produce high quality colored image. Their approach consists of 2 phases: 1. Downsample phase is for learning the image level and entity-level features 2. Upsample phase leverages these features to generate image colorization. Model learns a color distribution over pixels used to predict the color channels.

Advantages:

1. Their method provides a consistent object/background separation also reducing the color blurring on contours thus generating more detailed outputs.

2. Their method produced images with no splotches.

3. They conducted user study with 200 random images, on collecting votes, their images were preferred over other author’s models (53% vs 47%)

Disadvantages:

Occasionally their method fails to predict colors for some local regions. Their network cannot colorize objects with unusual or artistic colors.

Automatic Image Colorization using Deep Learning

Abhishek Pandey, Rohit Sahay, C. Jayavarthini

Input:

A single black and white image in 256 x 256 pixels was given as input. 18 gigabytes of images from ImageNet database are being used for training. All the images are being rescaled to 224 x 224 and 299 x 299 for encoding and inception.

Output:

The model produces images with realistic colors.

Methodology:

Their proposed methodology included building a deep convolutional neural network which includes 4 parts- the encoder component to produce mid-level features, the feature extraction component to produce high-level features, these two are then merged into the fusion layer and the output is generated using the decoder component.

Advantages:

It is an efficient way of coloring images using deep CNN. Nature elements involving rivers, trees, etc. have above 80% accuracy

Disadvantages:

Most of the colors were low saturated because of less diverse data set. Some of the objects are not colorized well and for that model has produced next probable colors.

Auto-Coloriz ation of Historical Images Using Deep Convolutional Neural Networks

Joshi MR, Nkenyereye L, Joshi GP, Islam SMR, Abdullah-Al Wadud M, Shrestha S.

Input:

The dataset created to train and test the model contains historical, heritage and cultural image repositories of Nepal. The images were collected from the ImageNet database and the internet. 1200 images of 256 x 256 were collected.

Output:

Colorized images were generated

Methodology:

The model starts with dataset collection and preprocessing the images to remove images having unusual aspect ratios and resizing images to 256 x 256. Then the images are converted into CIE L*a*b* where L is one layer for luminance and has packed three RGB layers into two chroma layers (a* and b*). Then the CNN model is implemented as a learning pipeline. It takes greyscale images as input along with the luminance. The a* and b* channels are extracted as the target values. L*a*b* to RGB conversion is applied as the final output.

Advantages:

The framework offers image colorization with improved or comparable performances as compared to various existing approaches from enhanced signal energy perspectives. The Mean Squared Error (MSE) was 6.08%, Peak Signal-to-Noise Ratio (PSNR) 34.65 dB, and 75.23% model accuracy.

Disadvantages:

Poor performance for some images due to small data size and variability of images in training set. Poor coloring results were obtained.

Auto Colorization of Gray-Scale Image Using YCbCr Color Space

Nidhal K. El Abbadi, Eman Saleem

Input:

A greyscale image to be colorized is provided as the input. An RGB image for reference is also provided.

Output:

Colorized images were generated.

Methodology:

A reference image has to be selected which is similar to the greyscale image in content and structure. Then both the images are converted into YCbCr color space. For each pixel present in the greyscale image, it is compared with the pixels of the reference image using Euclidean distance and the pixel with the minimum distance in the reference image, their color information is transferred to the greyscale image. Just like this the whole image is converted into colorized image.

Advantages:

This model provides a fully automated method to colorize images based on YCbCr color space and a reference image which provided good results. RMSE (Root Mean Square Error) equal to 10, average PSNR (Peak Signal-to-Noise Ratio) value equal to 30 and average MD (Maximum Difference) value as 100.

Disadvantages:

Color deformation takes place when there is a difference in the color histogram between black and white image and the reference image

Automatic Image Colorization Via Multimodal Predictions

Guillaume Charpiat,Matt hias Hofmann,Ber nhard Schölkopf

Input:

The input for the image described is a grey scaled image taken from a colored image to test its accuracy. The paper uses the grey scaled image of Monalisa by Da Vinci to test its accuracy.

Output:

The model reproduces colors for greyscale images. The multimodalit y framework proves extremely useful in areas such as Mona Lisa’s forehead or neck where the texture of skin can be easily mistaken with the texture of sky at the local level.

Methodology:

Using a multimodal model which helps in distinguishing similar objects of different colors to help achieve a greater accuracy in terms of color correction and image visibility.

Advantages:

The model is also based on the user input and user can interact, add more color points if needed, until a satisfying result is reached, or even place color points strategically in order to give indirect information on the location of color boundaries. Accuracy of the model is based on the time and dataset provide for the current dataset in the paper the accuracy for images such as a zebra is 95% and above but for images with a lot of objects the accuracy ranges from 70-85%.

Disadvantages:

With an image with many different objects and textures, such as a brick wall, a door, a dog, a head, hands, a loose suit. Because of the number of objects, and because of their particular arrangement, it is unlikely to find a single color image with a similar scene that we would use as a learning image.

Probabilistic Image Colorization

Amelie Royer, Alexander Kolesnikov, Christoph H. Lampert

Input:

Grayscale image to an embedding, which can be encoded with color information.

Output:

A colorized image from an input greyscale image.

Methodology:

At training time, all variables in the factors are observed, so a model can be efficiently trained by learning all factors in parallel. The paper used gated residual blocks as the main building component for the both networks.

Advantages:

The main innovation is that they treat colorization as a classification rather than regression task,combined with class-rebalancin g in the training loss to favor rare colors and more vibrant samples. The model is highly competitive with other approaches and tends to produce more saturated colors on average.

Disadvantages:

Though the model is able to capture the vibrance of the image it is unable to predict the correct color for a specific image as it is trained in a different way than most neural networks.

Image Colorization Using the Global Scene Context Style and Pixel-Wise Semantic Segmentation

Tram-tran nguyen-quyn h , Soo-hyung kim and Nhu-tai do

Input:

Greyscale image

Output:

A colorized image based on the training set provided.

Methodology:

Using an auto encoder architecture along with applying semantic segmentation based on pixel level to get a greater accuracy to detect areas of an object.

Advantages:

It can be paired with other machine learning and deep learning models to better improve the accuracy and color correction. The accuracy of the approach is: 0.823.

Disadvantages:

This approach can introduce some red noise in the image leading to a badly colorized image with undefined edges.

Affective Image Colorization

Xiao-Hui Wang, Jia Jia, Han-Yu Liao & Lian-Hong Cai

Input:

Greyscale image and reference images

Output:

Colorized image

Methodology:

We firstly jointly use text labels to semantically filter internet images with correct references Then secondly , we select a set of color themes in accordance to the affective word based on art theories.

Advantages:

Different image colorizations are generated from the varied reference pictures, and a graphical computer program is provided to simply choose the required result.

Disadvantages:

Selection of color themes leads to build up of bias.

4. Methodology

4.1 Method Used

Colorization of grayscale images provides insightful innovation on the domain of corrective computer vision. Enabling RGB images extracted from grayscale images is an innovative idea as this innovation can be used and applied in several problems of real life.

Convolutional neural networks are biologically inspired forms of multilayer perceptrons that are designed to mimic visual brain activity. These models overcome the limitations of the MLP design by utilizing the significant spatially local correlation found in natural images. CNNs, as opposed to MLPs, have the following characteristics:

1. 3D volumes of neurons: Neural networks are practically constructed by filling and connecting neurons in 3 dimensions: height, width and depth. Each neuron has a limited understanding of the field and it is connected with the layer before it. To build a CNN architecture, these neurons are stacked and connected. 2

2. Local connectivity: CNNs exploit spatial locality by establishing a local connection pattern between neurons in adjacent layers, much like receptive fields. The learned "filters" produce the strongest response to a spatially confined input pattern as a result of the architecture. Non-linear filters become progressively global by stacking many of these layers, allowing the network to first build representations of small parts of the input, then assemble representations of larger areas from them .

3. Shared weights: In CNN, every layer is copied using a field of view. These units form maps using the same weights and bias. This means that in the similar region, they respond in a similar way when the convolutional response is fired. By copying units in this way, the feature map built at the end has equal variance under changes in the locations of input features in the visual field,

4. Pooling: Feature maps are partitioned into rectangular subregions in a CNN's pooling layers, and the features in each rectangle are down sampled to a single value, usually by taking their average or maximum value. In addition to lowering the size of feature maps, the pooling method gives the features contained therein a degree of translational invariance, allowing the CNN to be more resilient to changes in their positions. Convolutional neural networks (CNNs) have achieved remarkable results in a variety of fields, including medical studies, and there is growing interest in radiology. Although deep learning has been the preferred method for a range of challenging tasks including picture classification and object recognition, it is not a panacea. Knowing the core ideas and benefits of CNN, as well as the limitations of deep learning, is critical for using it in radiology research with the goal of enhancing radiologist performance.

INPUT, STEPS, EXPECTED OUTPUT

Input: A grayscale Image in LAB Format.

Steps:

1. Import the grayscale images dataset as Numpy array.

2. Convert the grayscale images in LAB (L - Lightness, A - Green and Magenta , B - Blue and Yellow) format to RGB format.

3. Divide the dataset into a Training set and Test Set

4. Create a Convoluted Neural Network using multiple layers in Keras.

5. Use the Adam Optimizer and loss function to see the loss while training the model.

6. Check for errors using a sample image.

7. Train the model against a number of epochs

8. Predict the colorized output of input test images.

Expected Output: Colorized Image

DATASET INFORMATION

Dataset- Kaggle dataset - https://www.kaggle.com/shravankumar9892/image-colorization
Tools:

1. Tensorflow.keras - it is a powerful and easy to use open source library for developing and evaluating deep learning models. It allows us to define and train neural networks in just a few lines of code.

2. Numpy- for mathematical operations on arrays and matrices, it's an extension of numeric python and NumArray.

3. Google Colab - google colab is a cloud based jupyter notebook environment.

ALGORITHM

The project uses the following algorithm to colorize a given grayscale image.

The models used the following Keras APIs to build the hidden layers of the model.

- Conv2D layer with a relu activation function.

- Conc2D layer with a relu activation function.

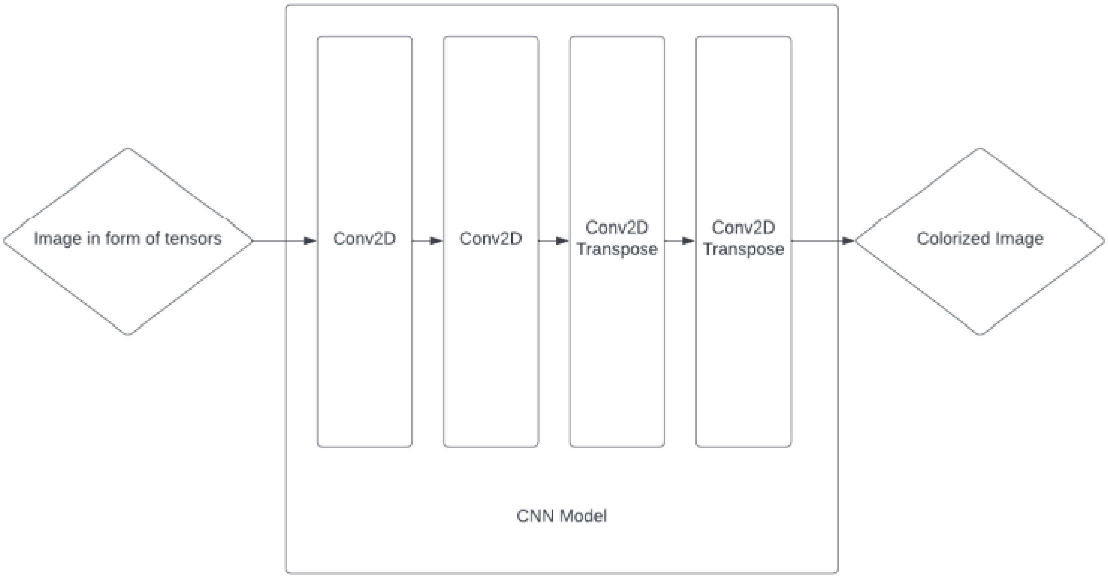
- Conv2D transpose layer with a relu activation function.

- Conv2D transpose layer with a relu activation function.

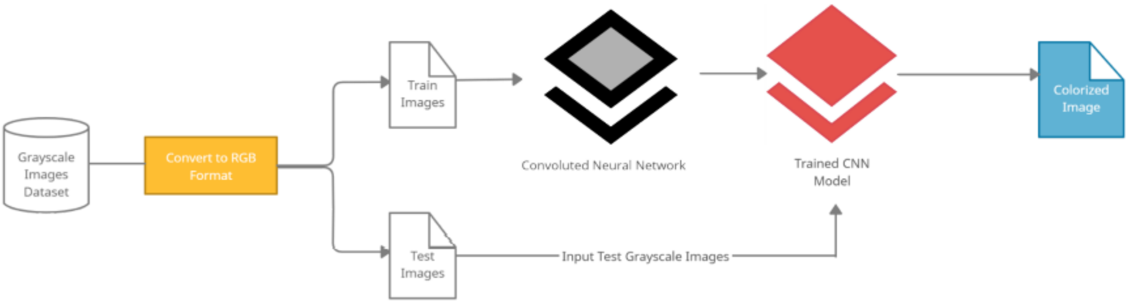
These layers make up the whole Deep Learning Model. The purpose of using the conv2D layers is to this layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs. This process downscales the images therefore to converse the image's quality we upscale it by using a conv2DTranspose layer.

A Conv2DTranspose layer works in the same way as the conv2D layer but the input vector used for it are transposed and then used for inputs, The need for transposed convolutions generally arises from the desire to use a transformation going in the opposite direction of a normal convolution, i.e., from something that has the shape of the output of some

convolution to something that has the shape of its input while maintaining a connectivity pattern that is compatible with said convolution. The 4 layers of convolution make up our CNN model that detects areas and colorizes them and outputs a colored image.



BLOCK DIAGRAM



4.2 Applications

Image colouring has various applications. Few innovative solutions that can be extended through application of this technologies are listed below:

1. Conversion of recorded videography from older times:

   The development of basic videography began in the late 19th century which was limited to fast screen fluxing and black and white compositions. Through the process of colorization, we can construct RGB channels to the image and make them colored.

2. Highly Detailed Image Reconstruction:

   Grayscale to RGB image reconstruction leads to images with high details in higher visual aesthetics. This allows greater illustration of image quality post reconstruction.

5. System Implementation

5.1 Code

```
import os


for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
#Importing the Libraries
%matplotlib inline
import matplotlib.pyplot as plt
import cv2
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from tensorflow.keras.applications.inception_resnet_v2 import InceptionResNetV2,
decode_predictions, preprocess_input
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.layers import Conv2D, Conv2DTranspose, Reshape
import tensorflow as tf




#Importing the images as np array
images_gray = np.load('../input/image-colorization/l/gray_scale.npy') #Input
images_lab = np.load('../input/image-colorization/ab/ab/ab1.npy') #Output


#Function to convert the lab images imported above to rgb images
def get_rbg_from_lab(gray_imgs, ab_imgs, n = 10):
 # Initializing with zeros ( or any random number)
 imgs = np.zeros((n, 224, 224, 3))
 imgs[:, :, :, 0] = gray_imgs[0:n:]
 imgs[:, :, :, 1:] = ab_imgs[0:n:]
```

```
# Changing the data type of the img array to 'uint8' for the model
 imgs = imgs.astype("uint8")

 imgs_ = []
 for i in range(0, n):
     imgs_.append(cv2.cvtColor(imgs[i], cv2.COLOR_LAB2RGB)) #Converting the color space
 imgs_ = np.array(imgs_)
 print(imgs_.shape)
 return imgs_


#Function to pipe line the gray scale images that we imported in cell 2 and convert them to usable format for Keras
def pipe_line_img(gray_scale_imgs, batch_size = 100, preprocess_f = preprocess_input):
 imgs = np.zeros((batch_size, 224, 224, 3))
 for i in range(0, 3):
     imgs[:batch_size, :, :,i] = gray_scale_imgs[:batch_size]
 return preprocess_f(imgs)


#TensorBoard is a visualization tool provided with TensorFlow
tbCallBack = tf.keras.callbacks.TensorBoard(log_dir='./folder_to_save_graph_3',histogram_freq=0, write_graph=True, write_images=True)


#Obtaining the rgb of the lab images for output
imgs_for_output = preprocess_input(get_rbg_from_lab(gray_imgs = images_gray, ab_imgs = images_lab, n = 300))


#Outputting the rgb image we obtained
plt.imshow(imgs_for_output[10])
```

n-Test Split

imgs_for_input_train, imgs_for_input_test,imgs_for_output_train,imgs_for_output_test = train_test_split(imgs_for_input, imgs_for_output, test_size=0.10,random_state=42)

#Making the simple CNN network using Keras (4 layers)

model_simple = Sequential()

model_simple.add(Conv2D(strides = 1, kernel_size = 3, filters = 12, use_bias = True, bias_initializer = tf.keras.initializers.RandomUniform(minval=-0.05, maxval=0.05) , padding = "valid", activation = tf.nn.relu))

model_simple.add(Conv2D(strides = 1, kernel_size = 3, filters = 12,use_bias = True, bias_initializer = tf.keras.initializers.RandomUniform(minval=-0.05, maxval=0.05) , padding = "valid", activation = tf.nn.relu))

model_simple.add(Conv2DTranspose(strides = 1, kernel_size = 3, filters = 12, use_bias = True, bias_initializer = tf.keras.initializers.RandomUniform(minval=-0.05, maxval=0.05) , padding = "valid", activation = tf.nn.relu))

model_simple.add(Conv2DTranspose(strides = 1, kernel_size = 3, filters = 3, use_bias = True, bias_initializer = tf.keras.initializers.RandomUniform(minval=-0.05, maxval=0.05) , padding = "valid", activation = tf.nn.relu))

#Adding the optimiser to optimise our neural network and Loss funcion to see the loss while training our model and compiling it

model_simple.compile(optimizer = tf.keras.optimizers.Adam(epsilon = 1e-8), loss = tf.losses.mean_squared_error)

imgs_for_s = np.zeros((300, 224, 224, 1))

imgs_for_s[:, :, :, 0] = images_gray[:300]

#Chekcing if we have added the number of input and output nodes properly by running a sample image for errors on its size

prediction = model_simple.predict(imgs_for_input)

prediction.shape

#Training our model with 100 epochs and batch size 16

model_simple.fit(imgs_for_input_train, imgs_for_output_train, epochs = 100, batch_size = 16)

#Predicting the output of the input test images using our model whch the model has never seen before

out = model_simple.predict(imgs_for_input_test)

#The input b/w image

plt.imshow(np.squeeze(imgs_for_input_test[10,:])) # Input

#Image Colorised by our model

plt.imshow(out[10,:]) # Ouput

5.2 Results and Discussion

Original Image	Colourised Image	PSNR Value
		PSNR = 34.36
		PSNR = 37.26
		PSNR = 42.21

Hence for this model, we have analysed the PSNR values of 10 sample images which have fallen in the range of 30 to 50 dB, 3 of which have been shown above. Images with higher PSNR values indicated high quality.

6 Conclusion

6.1 Conclusion

The purpose of this study is to design a new, fully autonomous colorization method that employs convolutional neural networks to reduce human effort and reliance on example colour photos. As informative yet discriminative features, a patch feature and a new semantic feature are extracted and fed into the neural network. An adaptive image clustering algorithm is adopted to incorporate global image information. The output chrominance values are further adjusted using combined bilateral filtering to achieve colorization quality. Because the proposed colorization is completely automated, it is stronger and more stable than older methods. It does, however, uses machine learning techniques and has its own set of limitations. It's expected to be 4 trained on a huge reference photo library that includes all possible objects.

6.2 Future Work

One extended application can be that we can implement the same model as reinforcement learning. The model currently utilized shared weights, to improve accuracy of the model we can implement more niche implementations such as more layers or for more optimized performance less attributes and more layers. The current edge detection model does not perfectly perform segmentation boundaries as seen in previous and thus a better model can be used to fix such issues. Though the library used to train the model is of 3 GB, in the domain of Machine Learning and AI it is still a small amount of data to be trained on, this can be fixed by either providing a larger dataset or by using a previous method suggested.

7 References

[1] He, M., Chen, D., Liao, J., Sander, P. V., & Yuan, L. (2018). Deep exemplar-based colorization. ACM Transactions on Graphics (TOG), 37(4), 1-16.

[2] Blanch, M. G., Mrak, M., Smeaton, A. F., & O'Connor, N. E. (2019, September). End-to-end conditional gan-based architectures for image colourisation. In 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP) (pp. 1-6). IEEE.

[3] Pucci, R., Micheloni, C., & Martinel, N. (2021). Collaborative Image and Object Level Features for Image Colourisation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2160-2169).

[4] Abhishek Pandey, Rohit Sahay, C. Jayavarthini. Automatic Image Colorization using Deep Learning. March 2020. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-6

[5] Joshi, Madhab R., Lewis Nkenyereye, Gyanendra P. Joshi, S. M.R. Islam, Mohammad Abdullah-AlWadud, and Surendra Shrestha. 2020. "Auto-Colorization of Historical Images Using Deep Convolutional Neural Networks" Mathematics 8, no. 12: 2258.

[6] Eman Saleem, Nidhal K. El Abbadi. Auto Colorization of Gray-Scale Image Using YCbCr Color Space. 2020. Iraqi Journal of Science, 2020, Vol. 61, No. 12, pp: 3379-3386.

[7] Charpiat G., Hofmann M., Schölkopf B. (2008) Automatic Image Colorization Via Multimodal Predictions. In: Forsyth D., Torr P., Zisserman A. (eds) Computer Vision – ECCV 2008. ECCV 2008. Lecture Notes in Computer Science, vol 5304. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978- 3-540-88690-7_10

[8] Royer, A., Kolesnikov, A., & Lampert, C. H. (2017). Probabilistic image colorization. arXiv preprint arXiv:1705.04258.

[9] T. -T. Nguyen-Quynh, S. -H. Kim and N. -T. Do, "Image Colorization Using the Global Scene-Context Style and Pixel-Wise Semantic Segmentation," in IEEE Access, vol. 8, pp. 214098-214114, 2020, doi: 10.1109/ACCESS.2020.3040737

[10] Wang XH，Jia J，Liao HY et a1.　Affective image colorization.　JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 27(6)：1119—1128 Nov.　2012.　DOI 10.　1007／s11390—012—1290—4

[11] Reinhard E., Adhikhmin M., Gooch B., & Shirley P. Color transfer between images. IEEE Computer graphics and applications, 21(5) : 34-41.

[12] Halder S.S., De K., Roy P.P. (2019) Perceptual Conditional Generative Adversarial Networks for End-to-End Image Colourization. In: Jawahar C., Li H., Mori G., Schindler K.

(eds) Computer Vision – ACCV 2018. ACCV 2018. Lecture Notes in Computer Science, vol 11362. Springer, Cham.

[13] Subjective evaluation of colorized images with different colorization model [2020 Wiley Periodicals LLC] Authors : Xiao Teng,Zhijiang Li,Qiang Liu,Michael R. Pointer,Zheng Huang,Hongguang Sun

[14] Ahmad S. Alhadidi. An Approach for Automatic Colorization of Grayscale Images. International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064

[15] Saeed Anwar , Muhammad Tahir, Chongyi Li, Ajmal Mian, Fahad Shahbaz Khan, Abdul Wahab Muzaffar. Image Colorization: A Survey and Dataset