School of Computer Science and Engineering

J Component report

Programme : B. Tech (CSE)

Course Title : Image Processing

Course Code : CSE4019

Slot : E1

Title: Re-colorization of Digital Grayscale Images

Team Members: **Mihir Gupte | 19BCE1149**

Siddharth Mehta | 19BCE1485

Shubhmoy Banerjee | 19BCE1300

Faculty: Dr. Geetha S.

Sign:

Date: 30/04/2022

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

School of Computer Science and Engineering

DECLARATION

We hereby declare that the project entitled **"Re-colorization of Digital Grayscale Images"** submitted by me to the School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus, Chennai 600127 in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology – Computer Science and Engineering** is a record of bonafide work carried out by me**.** We further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or university.

Mihir Gupte (19BCE1149)
Siddharth Mehta (19BCE1485)
Shubhomoy Banerjee (19BCE1300)

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

School of Computer Science and Engineering

CERTIFICATE

The project report entitled **"Re-colorization of Digital Grayscale Images"** is prepared and submitted by **Mihir Gupte (19BCE1149), Siddarth Mehta (19BCE1485), Shubhmoy Banerjee (19BCE1300)**. It has been found satisfactory in terms of scope, quality and presentation as partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology – Computer Science and Engineering** in Vellore Institute of Technology, Chennai, India.

Examined by:

Dr. Geetha S.

ACKNOWLEDGEMENT

I am highly indebted to Dr. Ganesan R, Dean of the School of Computer Science & Engineering, VIT Chennai and Principal Dr. Nithyanandam P, Head of the Department (HoD), B.Tech Computer Science and Engineering (SCSE), VIT Chennai, for the facilities provided to accomplish this project work. I would like to thank Dr. Geetha S., Faculty of the School of Computer Science & Engineering and Associate Dean of SCSE, VIT Chennai for their support and advice to get and complete this project work. I am extremely grateful to my family and friends who helped me in successfully completing this project.

Mihir Gupte

Siddharth Mehta

Shubh Moy

CONTENTS

ABSTRACT

Colorization of an image is very important as it gives a natural feeling to the pixels on the screen. Grey scale images have been in existence since 1826. Image colorization is the process of taking a grayscale (black and white) image as input and producing a colourized image that accurately depicts the input's semantic colours and tones (for example, an ocean on a clear sunny day must be convincingly "blue" – the model cannot colour it "hot pink").

Previously, black and white image colorization relied on manual human annotation, which sometimes resulted in desaturated images that were not "believable" as true colorizations. Zhang et al. opted to take on the subject of image colorization by utilising Convolutional Neural Networks to "see" how a grayscale image would look when colourized.

Zhang et al. started with the ImageNet dataset and transformed all images from RGB to Lab colour space to train the network. The Lab colour space has three channels, just like the RGB colour space. Lab, on the other hand, encodes colour information differently than RGB:

- Only the brightness intensity is encoded in the L channel.
- Green-red is encoded by the a channel.
- Blue-yellow is encoded on the b channel.

We can utilise the L channel as a grayscale input to the network because it only encodes intensity. The network will next have to learn to predict the a and b channels. We can now create our final output image using the input L channel and the anticipated ab channels.

1 INTRODUCTION

In this project, we implement Deep Learning models and teach them to learn colorization for pixels of grayscale images. Automatic image colorization often involves the use of a class of convolutional neural networks (CNN) called autoencoders. These neural networks are able to distil the salient features of an image, and then regenerate the image based on these learned features.

Using any colour photo as a training example is as simple as taking the image's L channel as input and the ab channels as the supervisory signal. Others have remarked on the ease with which training data is available, and earlier research has used enormous datasets to train convolutional neural networks (CNNs) to predict colour.

However, the prior attempts' outcomes have a desaturated appearance. Other researchers utilise loss functions, which encourage conservative forecasts, as one explanation. These losses come from conventional regression issues, where the goal is to reduce the Euclidean error between an estimate and the ground truth.

Color prediction is intrinsically multimodal, as many things can be coloured in a variety of ways. An apple, for example, is usually red, green, or yellow, but not blue or orange. We anticipate a distribution of probable colours for each pixel to correctly describe the multimodal character of the challenge.

We also re-weight the loss during training to emphasise unusual colours. This enables our model to take advantage of the complete range of large-scale data on which it is trained. Finally, we create a final colorization by taking the distribution's annealed mean. Colorizations that are more brilliant and perceptually lifelike than prior procedures are the ultimate result.

2 PROPOSED WORK

In our project, we will be experimenting with CIE LAB color spaces. The reason we use these instead of RGB color space is that by separating out the Y/L or the gray-scale component, the neural network only has to learn the remaining two channels for colorization. This reduces size of the network and speed of convergence.

We will use a vanilla autoencoder colorization model as a baseline model. In vanila autoencoders, the encoder used are often not deep enough to extract the global features of the image, which are necessary to determine how to color certain regions of the image. To satisfy competing requirements, two different neural pathways can be used on the encoder side. One path to obtain the global features, and another to obtain a rich representation of the image.

Furthermore we will make use of two CNN-based colorization models proposed by Richard Zhang : ECCV16 and SIGGRAPH17. This approach consists of predicting the image's a and b colour channels in the CIE Lab colorspace based on the lightness channel L. We use large-scale data to solve this challenge. Color prediction has the advantage of having virtually no training data: any colour photo can be used as a training example by simply taking the image's L channel as input and its ab channels as the supervisory signal.

This problem by nature is of multimodal nature, which means that one output can take multiple color ranges. An apple, for example, is usually red, green, or yellow, but not blue or orange.

In this project we do the following things -

1. Train three neural networks based on the architectures of ECCV16 and SIGGRAPH17 and one baseline model consisting of a simple AutoEncoder.
2. Evaluate their results on real-life grayscale images
3. Deploy an end-to-end API to communicate with the model in order to deploy them as a webapp

3 LITERATURE REVIEW

Colorization of images has been an extensively researched problem in the past few years due to the accessibility to extensive hardware resources and good research done in Deep Learning. Traditional image colorization required a significant amount of human effort. Autoencoders are a type of convolutional neural network (CNN) that is frequently used in automatic image colorization. These neural networks are capable of distilling an image's most important elements and then regenerating the image using these learned properties.

We referred to the following papers in order to make this project -

1. Colourful Image Colorization[1] - In this paper, Richard Zhang creates two models that we implement in our project which are - ECCV16 and SIGGRAPH17. We will discuss their architecture further in the other sections. The main contributions of his paper are: devising an acceptable objective function that addresses the colorization problem's multimodal ambiguity and captures a wide range of colours, and introducing a new framework for evaluating colorization algorithms that could be used for additional picture synthesis tasks.
2. Color-UNet++: A resolution for colorization of grayscale images using improved UNet++ [4] - For the end-to-end solution of various colorization problems, this study introduces Color-UNet++, a deep convolutional network framework. Color-UNet++ has been tweaked to reduce gradient dispersion and explosion during backpropagation by recording more transfer and intermediate data. To tackle the problem of unequal overlap, they tweak the de-convolution structure. They build the model in YUV rather than RGB colour space, with a coloring-problem-appropriate objective function that can capture a wide range of colours. The method's superiority is supported by a significant number of experimental results on the LFW and LSUN datasets.

4 MODULES OF THE PROJECT

Module 1: Training CNN models to colourize digital images

As discussed previously, we have implemented three deep learning CNNs for this project. In this section, we will discuss them briefly and show their architecture. The models are as follows -

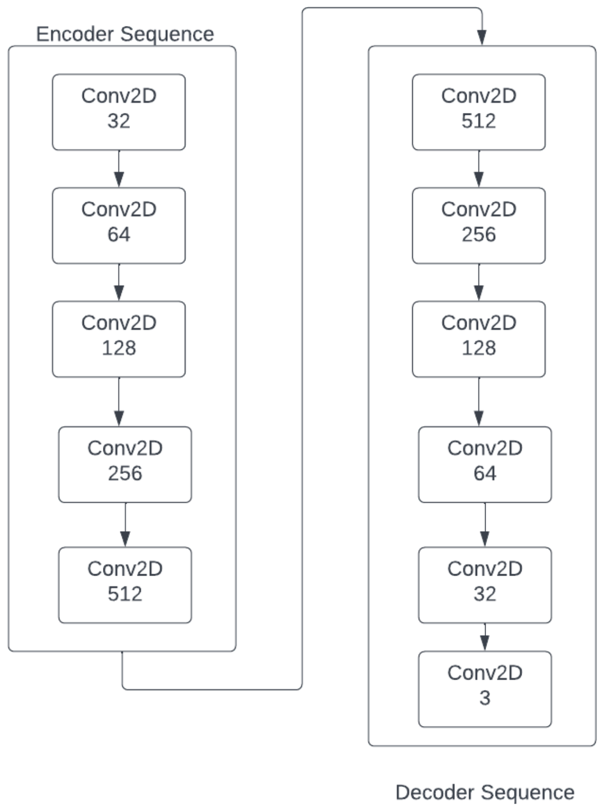
1. Basic Autoencoders -

   Autoencoders were created with the goal of generating an efficient representation of an image or other sort of input using unsupervised machine learning. The encoder and the decoder are the two pieces of an autoencoder. The encoder learns a reduced dimensional representation of the input data through a succession of CNN and downsampling. The decoder then attempts to recreate the data from these representations by using CNN and upsampling. If a well-trained decoder can generate data that is identical to or as close to the original input data as feasible, this indicates that the encoder was successful in finding a compressed version of the original input data.

   The model consists of Conv2D layers and after each layer it is normalised using BatchNormalization and then ReLU is used as the activation function.

   The final performance of the model was subpar. Output images were very desaturated as compared to the original images, hence more training was required but in the interest of time and resources we stopped at that stage.

The architecture for our model is as follows -

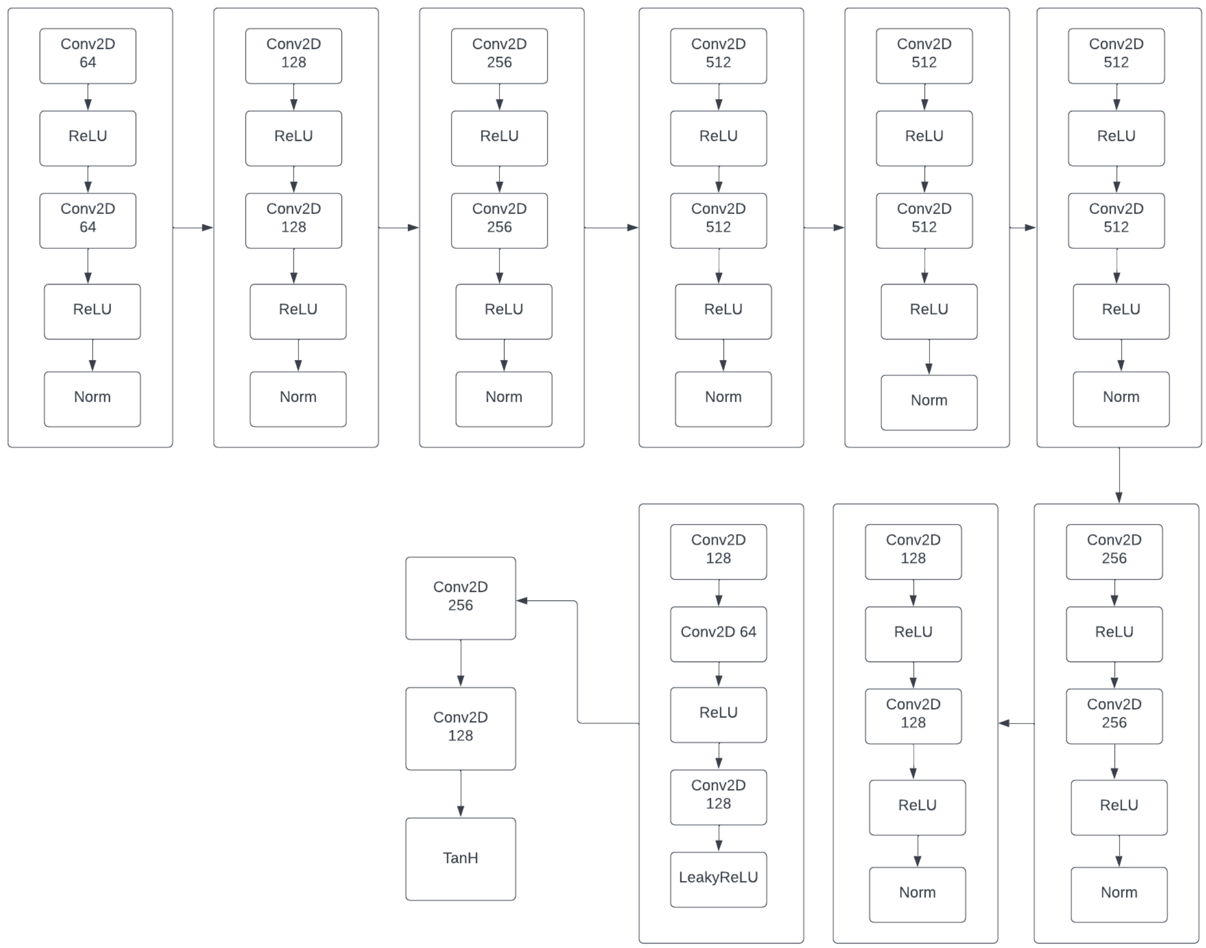


2. ECCV16 -

This model also utilises an autoencoder structure, but implements it in a block form, where one block consists of a series of Conv2D and Normalisation layers instead of separate layers. The author also specifies that their model is “similar to an autoencoder form” except that the input and output are different image channels, suggesting the term cross-channel encoder.

The performance of this model is a lot better than the previous one. It gives a very vibrant feel to most of the images that are colourized. The only drawback is that sometimes these images are too vibrant giving off an unrealistic tint and feel to them. This will be clear in the examples shown in the next module.

The architecture of the model is as follows -



3. SIGGRAPH17 -

This CNN has a very similar architecture to EVVC16 but there are a few changes to the architecture in order to address the above mentioned problems. Hence this model performs very well and is able to give a very real feel to the images.

The results are discussed in the next module.

Originally, the models were trained on ImageNet dataset, but due to hardware constraints we trained them on a smaller dataset found on kaggle for re-colorization [2].

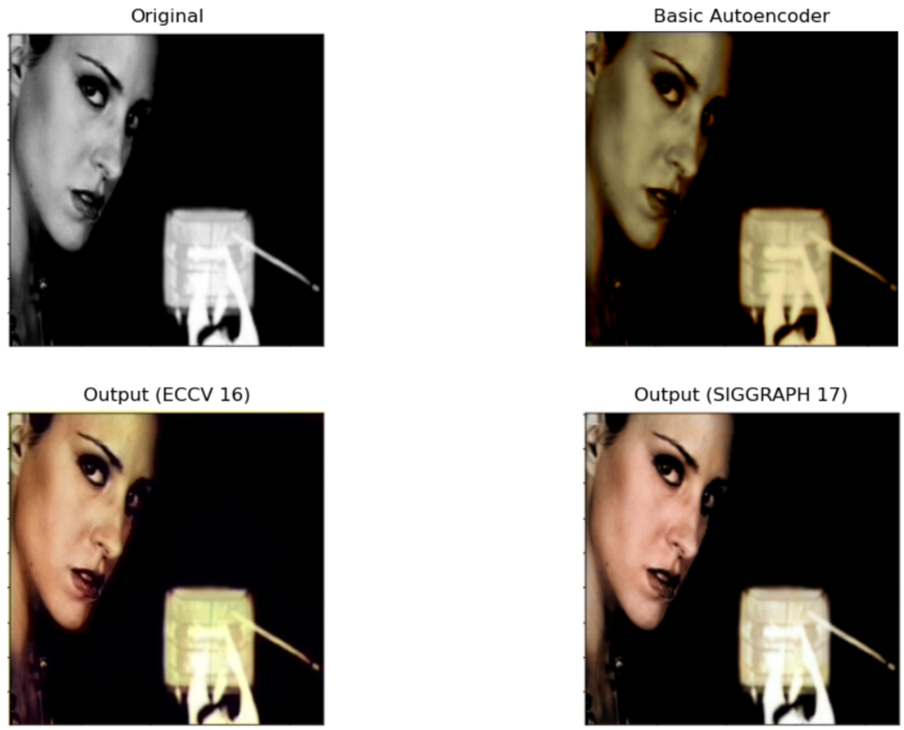
Module 2: Evaluating the results on real-life greyscale images

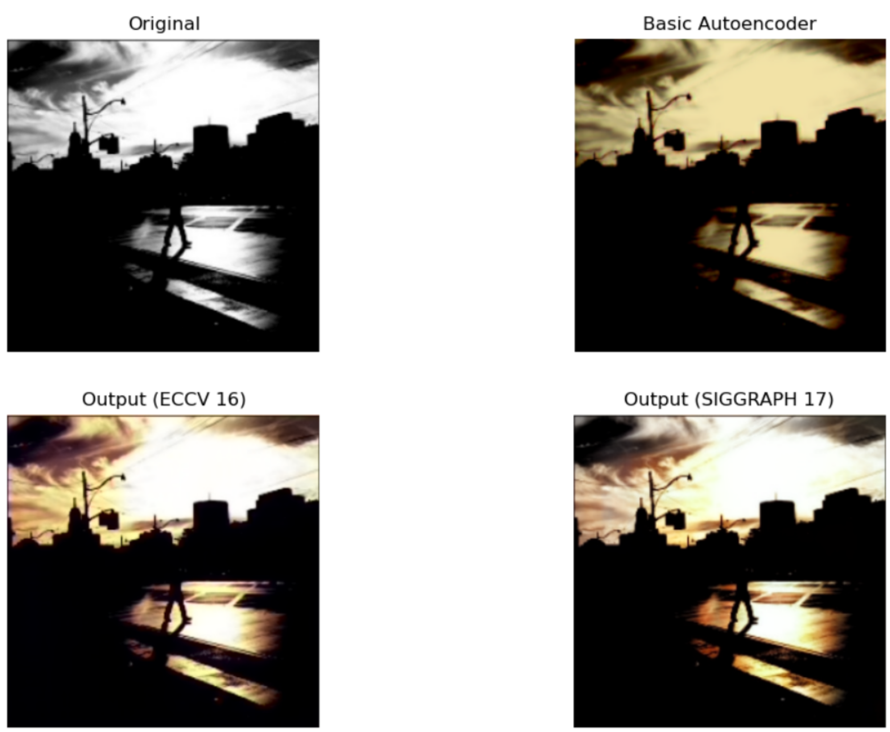
We will now demonstrate the results of the model.

For the first example, we took a grayscale photo of a person in order to evaluate how the model predicts the skin tone with respect to the environment and these were the results.

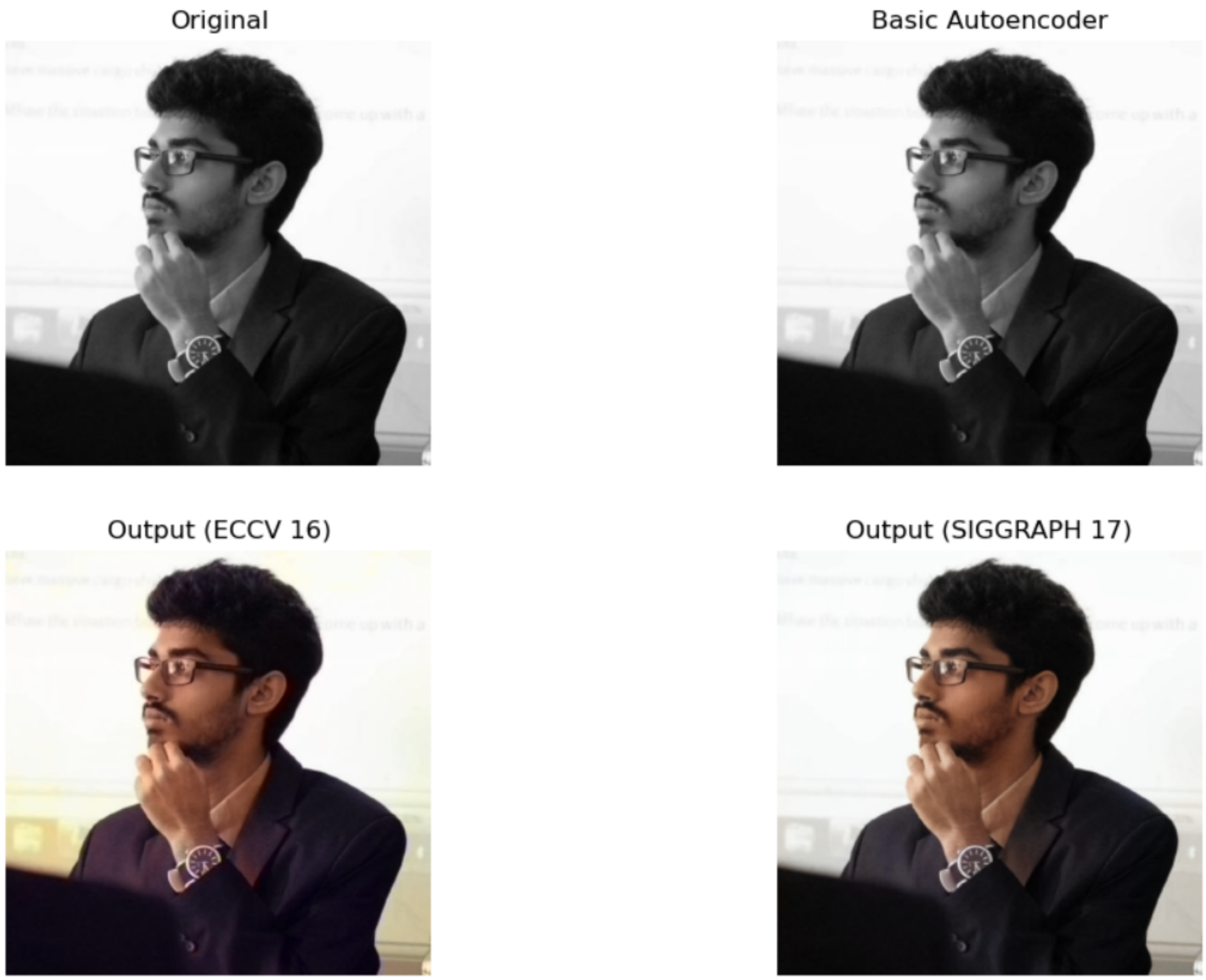
As we can see the autoencoder produces a very desaturated image while the other two models produce a more realistic image. Even in those, ECCV16 produces an image with very vibrant colours while SIGGRAPH produces an image with toned colours and has a more realistic feel to it.

Some other images include -





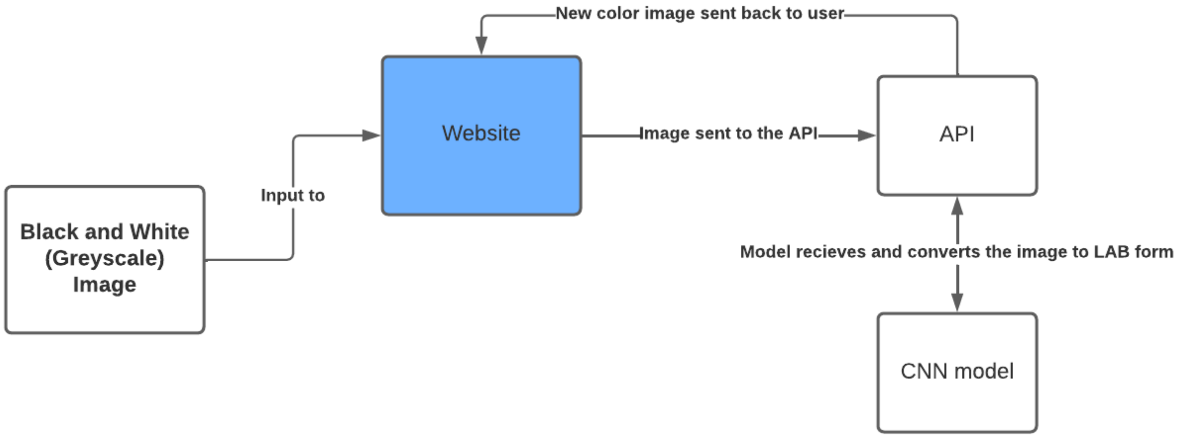
This next image is a sample of colourizing one of our image -



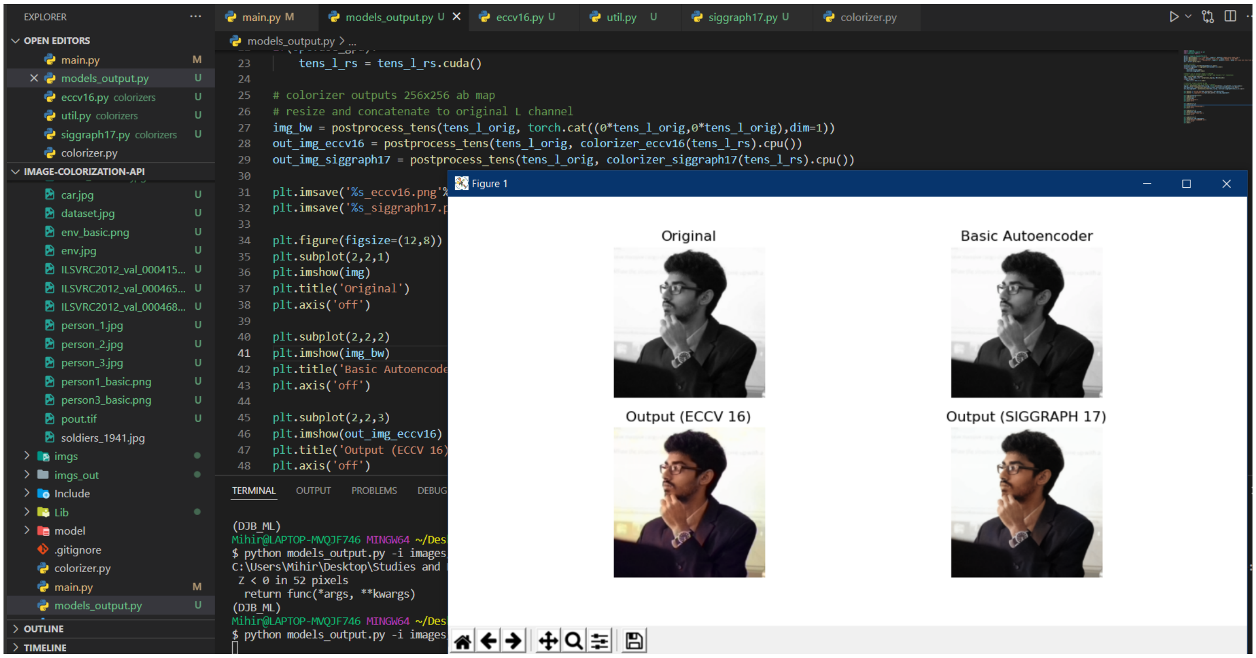
Module 3: Implementing an API to deploy the ML application

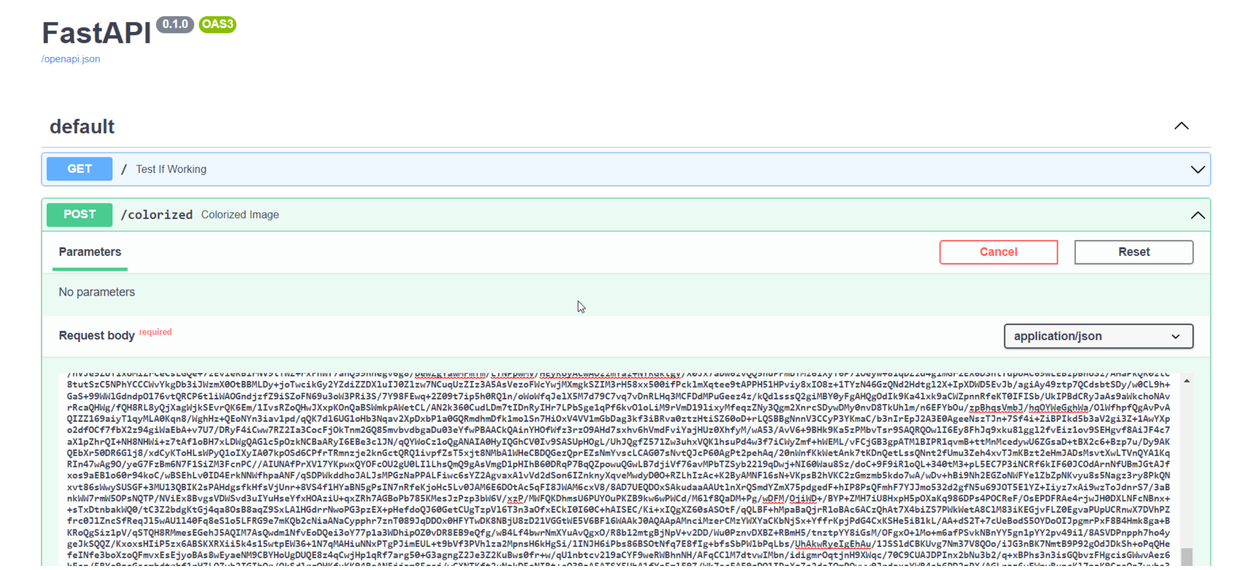
In this module, we use Fast-API to deploy our CNN model. Since we cannot directly feed an image through the front end, we convert the images to Base64 and send it as a string to the backend where we decode the image and then read it. From here we feed the image to the model and then the output image again goes through the same process in a reverse order.

The architecture of the entire workflow can be summarised by this image -



5 SCREENSHOTS OF THE PROJECT





6 CONCLUSION & FUTURE WORK

Based on these results, we conclude that we can easily obtain a colourized image using the CNN model which we used. An easier method to build our model would be using Autoencoder and Keras dataset. Moreover, the result we obtain using the model can be further used for colorization of videos as well.

In future, thus we can extend this method to be used in videos and streams to colourize them on the backend, which will lead to a lot of optimisation in recording and storage as well.

References and Citations

[1] https://arxiv.org/pdf/1603.08511.pdf
[2] https://www.kaggle.com/datasets/shravankumar9892/image-colorization
[3] https://github.com/richzhang/colorization/
[4] https://link.springer.com/article/10.1007/s11042-021-10830-2