SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

A

MINI PROJECT REPORT

ON

"Colorizing Old B&W Images"

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE FULFILLMENT OF THE REQUIREMENT

OF

Laboratory Practice VI

Fourth Year Computer Engineering

Academic Year 2024-25

BY

Name of Students:	**Exam No.:**
Abhang Pratik Sunil	B1905704201
Borase Darshana Ravasaheb	B1905704228
Darekar Omkar Balasaheb	B1905704232
Gunde Chaitanya Diliprao	B1905704256

Under the Guidance of

Mr. K. A. Shinde

Sinhgad Institutes

DEPARTMENT OF COMPUTER ENGINEERING

STES'S SINHGAD INSTITUTE OF TECHNOLOGY AND SCIENCE

NARHE, PUNE – 411041

Sinhgad Institutes

Department of Computer Engineering
Sinhgad Institute of Technology and Science, Narhe, Pune

CERTIFICATE

This is to certify that,

Name of Students:	**Exam No.:**
Abhang Pratik Sunil	B1905704201
Borase Darshana Ravasaheb	B1905704228
Darekar Omkar Balasaheb	B1905704232
Gunde Chaitanya Diliprao	B1905704256

studying in BE Computer Engineering Course SEM-VIII has successfully completed their LP-VI Lab Mini-Project work titled **"Colorizing Old B&W Images"** at Sinhgad Institute of Technology and Science, Narhe in the fulfillment of the bachelor's degree in engineering of Savitribai Phule Pune University, during the academic year 2024-2025.

Mr. K. A. Shinde	Dr. G. S. Navale	Dr. S. D. Markande
Guide	**Head of Department**	**Principal**

SINHGAD INSTITUTE OF TECHNOLOGY AND SCIENCE, NARHE, PUNE– 411041

Place : Pune

Date :

ACKNOWLEDGEMENT

We take this opportunity to acknowledge each and every one who contributed towards our work. We express our sincere gratitude towards guide **Mr. K. A. Shinde**, Assistant Professor at Sinhgad Institute of Technology and Science, Narhe, Pune for his valuable inputs, guidance and support throughout the course.

We wish to express our thanks to **Dr. G. S. Navale,** Head of Computer Engineering Department, Sinhgad Institute of Technology and Science, Narhe for giving us all the help and important suggestions all over the Work.

We thank all the teaching staff members for their indispensable support and priceless suggestions. We also thank our friends and family for helping in collecting data, without their help **Colorizing Old B&W Images** report have been completed. At the end our special thanks to **Dr. S. D. Markande,** Principal Sinhgad Institute of Technology and Science, Narhe for providing ambience in the college, which motivates us to work.

Name of Students: **Sign:**

Abhang Pratik Sunil

Borase Darshana Ravasaheb

Darekar Omkar Balasaheb

Gunde Chaitanya Diliprao

Sr. No.	Title	Page No.
1.	Problem Statement	1
2.	Objective	1
3.	Pre-Requisites	1
4.	Theory	1
5.	Process	2
6.	Code	3
7.	Conclusion	6

1. Problem Statement

Colorizing Old B&W Images: color old black and white images to colorful images

2. Objective

To build a deep neural network model that can Colorizing Old B&W Images: color old black and white images to colorful images using CNN.

3. Pre-Requisites

Knowledge of Python programming.
Knowledge of Deep learning algorithms.
Model Files:

colorization_deploy_v2_prototxt.
pts_in_hull.npy.
colorization_release_v2.caffemodel.

4. Theory

Colorizing black and white images using deep learning models is a task that has been extensively researched in recent years. Deep learning models use the available color images to learn the relationship between the black and white and color images. Once trained, these models can colorize any black and white image by predicting the color channels from the brightness channel. There are various deep learning models that can be used for image colorization, including Generative Adversarial Networks (GANs), Convolutional Neural Networks (CNNs), and Autoencoders.

In this project, a pre-trained deep learning model is used to colorize a black and white image. The model used in this code is a Convolutional Neural Network (CNN), which has been trained on a large dataset of color images. The model predicts the AB channels of the LAB color space from the L channel, where the L channel represents the brightness information of the image, and the AB channels represent the color information.

The pre-trained model is loaded into memory, and the required files are read from the disk. The centers for the AB channel are loaded, which were obtained by clustering the AB channels of the color images in the training dataset. These centers are then used to colorize the input image.

The input image is converted to the LAB color space, and the brightness information is extracted from the L channel. The brightness information is then passed to the pre-trained model, which predicts the AB channels. The predicted AB channels are then concatenated with the L channel to obtain the colorized image. Finally, the colorized image is converted back to the BGR color space and displayed to the user.

5. Process

First, the required libraries are imported, including NumPy, argparse, OpenCV (cv2), and os.

Then, the code loads the pre-trained model files from the specified directory and sets up an argument parser to allow the user to provide the path to the input black and white image.

After that, the code loads the pre-trained model, followed by loading the centers for the AB channel. The AB channel is the color information of the image, and the pre-trained model uses a clustering technique to determine the centers of the AB channel. These centers are then used to colorize the input image.

Next, the code loads the input image, converts it to the LAB color space, and resizes it to 224x224 pixels. The L channel, which represents the brightness information, is then subtracted by 50 to reduce the brightness.

The code then uses the pre-trained model to colorize the input image by passing the L channel to the model, obtaining the predicted AB channels, and concatenating them with the L channel to obtain the colorized image. The colorized image is then converted back to the BGR color space, clipped between 0 and 1, and multiplied by 255 to obtain the final colorized image.

Finally, the code displays the original and colorized images using OpenCV's imshow function and waits for the user to press a key to close the windows.

In summary, this code performs image colorization using a pre-trained deep learning model, demonstrating the use of libraries like NumPy, argparse, and OpenCV to load and process the image and model files.

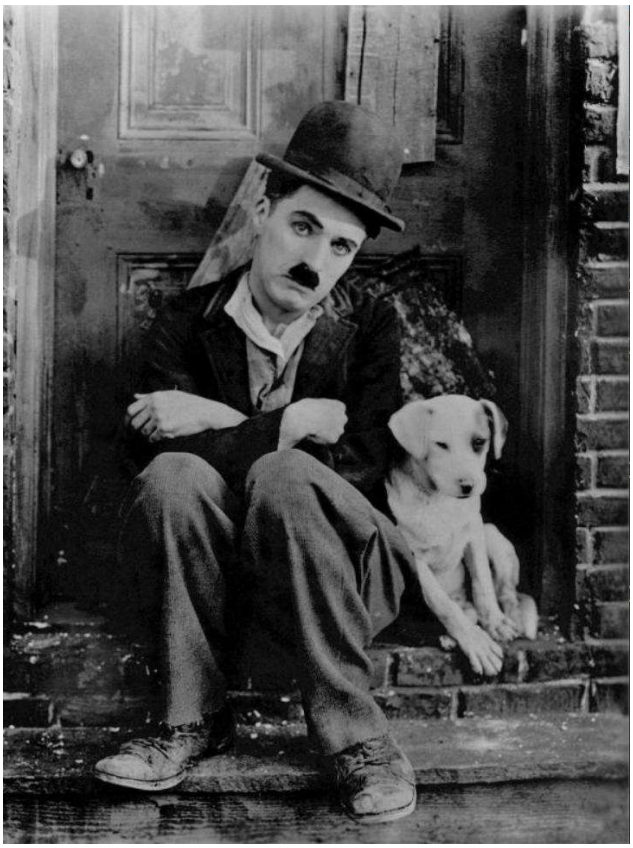
6. Code

Input:

```
# Import Modules
import numpy as np
import argparse
import cv2
import os
'''
Training a colorization model is computationally expensive and requires a large dataset.
Instead, we use a pre-trained model.

Required files:
1. colorization_deploy_v2.prototxt
2. pts_in_hull.npy
3. colorization_release_v2.caffemodel
   (Download: https://www.dropbox.com/s/dx0qvhhp5hbcx7z/colorization_release_v2.caffemodel?dl=1)
'''


# File paths
DIR = r"./colorize"
PROTOTXT = os.path.join(DIR, "colorization_deploy_v2.prototxt")
POINTS = os.path.join(DIR, "pts_in_hull.npy")
MODEL = os.path.join(DIR, "colorization_release_v2.caffemodel")

# Argument parser
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", type=str, required=True, help="Path to input black and white image")
args = vars(ap.parse_args())

# Load the pre-trained model
print("[INFO] Loading model...")
net = cv2.dnn.readNetFromCaffe(PROTOTXT, MODEL)
pts = np.load(POINTS)

# Populate cluster centers as 1x1 convolution kernel
pts = pts.transpose().reshape(2, 313, 1, 1)

# Set cluster centers in the model
net.getLayer(net.getLayerId("class8_ab")).blobs = [pts.astype(np.float32)]
net.getLayer(net.getLayerId("conv8_313_rh")).blobs = [np.full([1, 313], 2.606, dtype=np.float32)]

# Load input image
image = cv2.imread(args["image"])
if image is None:
    raise FileNotFoundError(f"Image not found at {args['image']}")

scaled = image.astype("float32") / 255.0
lab = cv2.cvtColor(scaled, cv2.COLOR_BGR2LAB)

# Resize and preprocess the L channel
resized = cv2.resize(lab, (224, 224))
L = cv2.split(resized)[0]
```

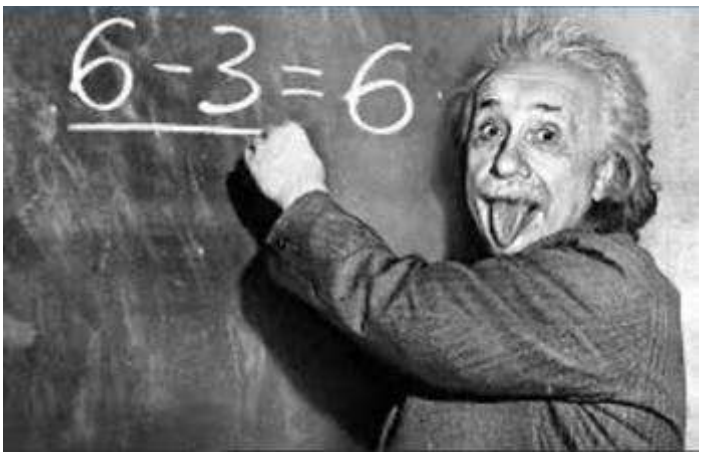
```
L -= 50

# Run the model
print("[INFO] Colorizing image...")
net.setInput(cv2.dnn.blobFromImage(L))
ab = net.forward()[0, :, :, :].transpose((1, 2, 0))

# Resize the ab channels to match the original image size
ab = cv2.resize(ab, (image.shape[1], image.shape[0]))
L = cv2.split(lab)[0]

# Concatenate L with ab and convert to BGR
colorized = np.concatenate((L[:, :, np.newaxis], ab), axis=2)
colorized = cv2.cvtColor(colorized, cv2.COLOR_LAB2BGR)
colorized = np.clip(colorized, 0, 1)
colorized = (255 * colorized).astype("uint8")

# Display the results
cv2.imshow("Original", image)
cv2.imshow("Colorized", colorized)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

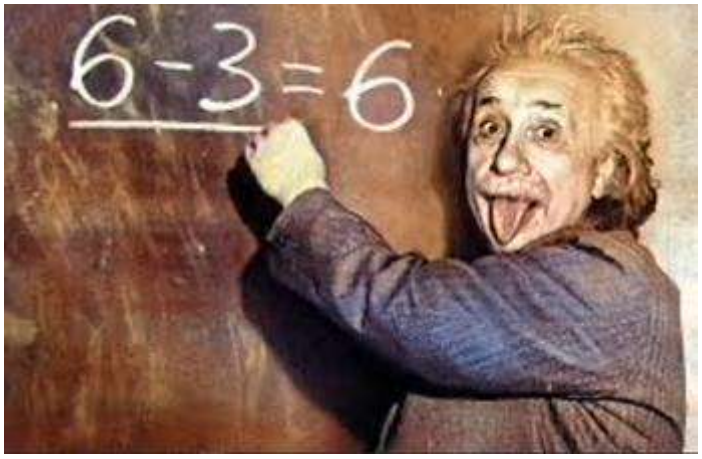
Output:




HEBARD


HEBARD


6-3=6


6-3=6

7. Conclusion

Thus, Black and White images were converted to color images using CNN. In this project, we successfully demonstrated the process of converting black and white images into color images using a pre-trained Convolutional Neural Network (CNN). By leveraging a deep learning model trained on a large dataset of color images, we were able to predict the missing color information (AB channels) from the brightness component (L channel) of the input image in the LAB color space. The model output was post-processed and converted back to a full-color image in the BGR format.