



# INFORMATION TECHNOLOGY

## Coloring Black and White Images

Team 6

K.Shivani, 21wh1a1293  
Arsheeya Tabassum, 21wh1a1294  
K.Sarayu, 21wh1a1295  
Kurmala Vigna, 21wh1a1296  
Shaik Aasima Afroz, 21wh1a1297



## PROBLEM STATEMENT

To colorize black and white images using CNN(convolutional neural network) The project aims to develop a deep learning model to add color to a black and white image. This project helps in addressing the need for enhancing historical and monochromatic visual content by adding life to it.



## OBJECTIVE

The objective of this project is to develop a CNN-based model that accurately colorizes grey-scale images, minimize the loss of original content and optimize the project for real-time coloring.



# Framework used in model

- Caffe Model
- LAB space



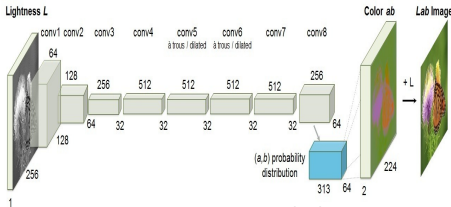
## Packages used in the code

- **tkinter**: Used for creating the graphical user interface (GUI).
- **PIL** (Python Imaging Library):  
Used for image processing, including loading and resizing images.
- **os**: Used for working with the operating system, such as file path manipulation.
- **numpy**:  
Used for numerical operations and array manipulation.
- **cv2** : Used for computer vision tasks, including image processing and manipulation.
- **uuid**: Used to generate unique identifiers for the output image file.
- **font**: Used to set the font size in the GUI.
- **pygame**: Used for playing sound effects.



# Algorithm used in the code

- **CNN:** A Convolutional Neural Network (CNN) is a specialized deep learning architecture designed primarily for processing and analyzing visual data, such as images and videos.
- CNNs introduce a set of powerful techniques for feature extraction and hierarchical learning.





# Framework used for Deployment

- **Tkinter** Tkinter is a Python library for creating graphical user interfaces (GUIs). It provides a set of tools and widgets to design and build interactive desktop applications it's easy to use and has cross-platform compatibility.



# Implementation of code

```

1 import numpy as np
2 import cv2
3
4 print("loading models.....")
5 net = cv2.dnn.readNetFromCaffe('colorization_deploy_v2.prototxt', 'colorization_release_v2.caffemodel')
6 pts = np.load('pts_in_hull.npy')
7
8
9 class8 = net.getLayerId("class8_ab")
10 conv8 = net.getLayerId("conv8_313_rh")
11 pts = pts.transpose().reshape(2,313,1,1)
12
13 net.getLayer(class8).blobs = [pts.astype("float32")]
14 net.getLayer(conv8).blobs = [np.full([1,313],2.606,dtype='float32')]
15
16
17 image = cv2.imread('human_bw.jpeg')
18 scaled = image.astype("float32")/255.0
19 lab = cv2.cvtColor(scaled,cv2.COLOR_BGR2LAB)
20
21
22 resized = cv2.resize(lab,(224,224))
23 L = cv2.split(resized)[0]
24 L -= 50
25
26
27 net.setInput(cv2.dnn.blobFromImage(L))
28 ab = net.forward()[0, :, :, :].transpose((1,2,0))
29
30 ab = cv2.resize(ab, (image.shape[1],image.shape[0]))
31
32 L = cv2.split(lab)[0]
33 colorized = np.concatenate([L[:, :, :, :].newaxis, ab, axis=2)
34
35 colorized = cv2.cvtColor(colorized,cv2.COLOR_LAB2BGR)
36 colorized = np.clip(colorized,0,1)
37
38 colorized = (255 * colorized).astype("uint8")
39
40 cv2.imshow("Original",image)
41 cv2.imshow("Colorized",colorized)
42 cv2.waitKey(0)

```





# Execution of code





## Project Contribution

Roll no	Name	Role
21wh1a1293	K.Shivani	Main GUI code
21wh1a1294	Arsheeya Tabassum	Back-end code, model research, GUI support
21wh1a1295	K.Sarayu	Analyzing data, Main GUI Code, PPT using LaTeX
21wh1a1296	Kurmala Vigna	GUI support, analyzing data, back-end code
21wh1a1297	Shaik Aasima Afroz	Analyzing data, GUI support, back-end code



# THANK YOU!