TITLE

GRAYSCALE VIDEO COLORIZATION

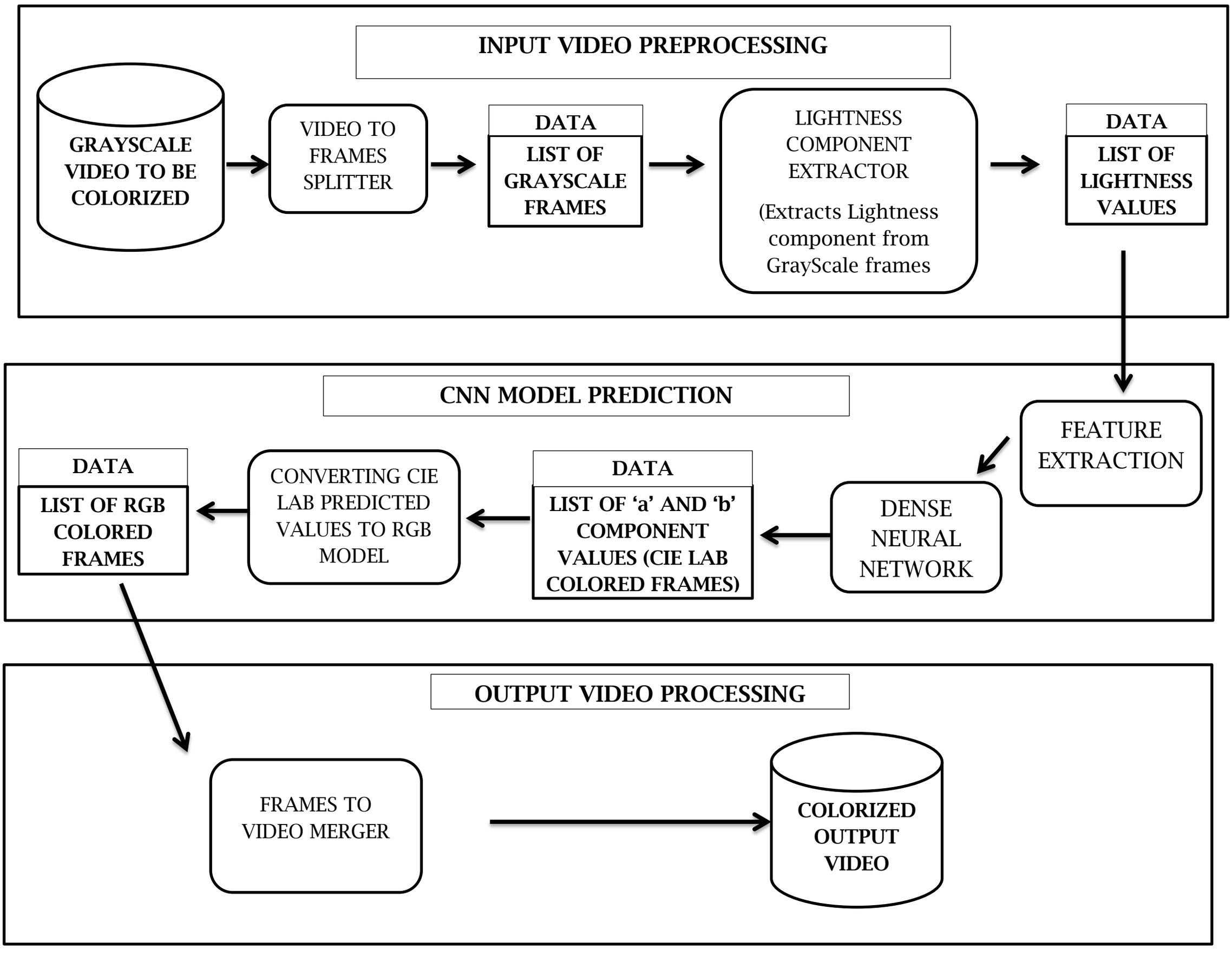
ABSTRACT

Videos in olden days were produced in gray scale format. With major advancements in technologies, there evolved means to take and produce videos in color completely. Color videos add a phenomenal positive impact and enhance user experience than old black and white or gray scale videos. In addition to creating new videos in color, there also arises demand for some old videos to be colorized and be converted into color videos. Such colored old videos make people to relive the nostalgia that they once held towards that movie. With major advancements in Machine Learning and Depp Learning techniques, automation in colorization cantake a huge leap forward. We propose a deep learning technique which automates the process of converting a grayscale video to a fully coloured video. This involves constructing a deep learning model where given a grayscale video as input, it predicts the colour of each pixel based on an input training dataset and emits out a colorized video.

INTRODUCTION

The traditional colorization method implements a process where humans manually determine the colour to be included in every frame of the video. This is a hugely time consuming and a costly job. If there is high demand for colorization of an old video, the budget would also be high and hence colorization is viable. But for colorization for videos which are of low demand, the budget cannot afford it. Automation in old video colorization is needed for videos to be colorized with little cost and less time spent. The RGB colour model is a famous model for specifying any colour. It is a model where red, blue and green colours mixed in different proportions emit different colours. **Another colour model named CIE Lab colour model is a suitable model for this project.** In CIE Lab model, similar to RGB model, there are 3 channels available. But here, colour information is encoded only in the 'a' (green-red component) and 'b' (blue-yellow component) channels. The third channel L (lightness) encodes intensity information only. The grayscale image wanted to be coloured has values of the L channel of the image in the Lab colour space and our objective to find the 'a' and 'b' components. The Lab image so obtained can be transformed to the RGB colour space using standard colour space transforms. The Deep Learning model predicts 'a' and 'b' channels taking in the 'L' channel of Gray scale images as input. The predicted **'a', 'b'** channels and the input gray scale **'L'** channel form a **CIE Lab image**. The CIE Lab image can then be converted into an RGB image.

BLOCK DIAGRAM (Colorizing Gray Scale Video)



MODULES

MODULE 1: INPUT VIDEO PREPROCESSING

❖ **Description:**

The functionality of the module is to split an input gray scale video into its constituent gray scale frames.

❖ **Pseudo Code:**

(i) Start
(ii) Get the Input Gray Scale video location from the user
(iii) Convert the video into a list of constituent Gray scale frames
(iv) For each Gray scale Frame:
    (i) Save it in the file system
(v) Stop

MODULE 2: CNN MODEL PREDICTION

❖ **Description:**

The functionality of the module is to use the Trained CNN model to colorize the frames. The CNN model takes the Lightness channel value in Gray scale frames as and predicts a and b channel value of CIE Lab image model. The CIE Lab images are then converted into RGB images.

❖ **Pseudo Code:**

(i) Start

(ii) Load the stored Gray scale frames into a pytorch dataset.

(iii) For each Gray scale frame:

// Using the Trained CNN model for prediction of a // and b components of CIE Lab image model

(i) Apply the filter to the image to obtain feature map // convolution

(ii) Apply ReLU activation

(iii) Apply Pooling to reduce the dimension of feature map

(iv) Flatten the feature map

(v) Give the flattened map as input to the DNN which predicts a and b channels of CIE Lab model

(vi) Convert the CIE Lab image to an RGB image

(vii) Store it in the file system

(iv) Stop

MODULE 3: OUTPUT VIDEO PROCESSING

❖ **Description:**

The functionality of the module is to merge the list of colored frames into an output color video.

❖ **Pseudo Code:**

(i) For each stored RGB color frame:

(i) Add it to the intermediate array 'arr'

(ii) Convert the frames in 'arr' into a video which is the output color video.

(iii) Release the video so that it gets stored in the file system.

DATASET AND IMPLEMENTATION DETAILS

DATASET DETAILS

DESCRIPTION:

Our project is colorizing the video available in grey scale format into a **coloured video**. So, we need a dataset which contains set of images to train and test and then using this, the **videos (frame by frame)** are colorized.

Here, the **grey scale images** are taken as input and then they are colorized which is given as output. This dataset contains two set of files in which one file is filled with a set of grey scale images and another file consists of a set of '**a' and 'b' channels of CIE LAB colour space images.**

Link:

https://www.kaggle.com/shravankumar9892/image-colorization

SAMPLE DATASET:

The dataset consists of features and labels.

Feature Attributes:

The Features are Grayscale images. The L channel values of Grayscale exist. The images are of size 224 x 224 with each pixel having Lightness information and there are 25000 images available.

SNAPSHOT:

Six images entry

```
array([[[237, 135,  75, ...,  35,  37,  63],
        [234, 152,  72, ...,  41,  49,  47],
        [215, 216, 104, ...,  73,  48,  82],
        ...,
        [ 40,  45,  38, ...,  11,   8,   8],
        [ 40,  56,  47, ...,   8,   7,  10],
        [ 42,  60,  79, ...,  12,  13,  15]],

       [[202, 204, 207, ..., 133, 136, 138],
        [199, 199, 200, ..., 123, 128, 134],
        [196, 159, 153, ..., 110,  76, 131],
        ...,
        [222, 200, 194, ...,  98, 108, 166],
        [224, 224, 224, ..., 165, 166, 165],
        [224, 227, 227, ..., 170, 168, 167]],

       [[ 37,  37,  40, ..., 218, 218, 218],
        [ 36,  38,  40, ..., 217, 217, 217],
        [ 38,  39,  40, ..., 217, 217, 217],
        ...,
        [ 13,  56, 144, ..., 220, 221, 216],
        [ 12,  27, 145, ..., 226, 228, 238],
        [ 14,  18,  83, ..., 220, 212, 233]],
```

```
[[143, 179, 179, ..., 179, 178, 143],
 [199, 251, 250, ..., 252, 251, 199],
 [198, 253, 244, ..., 255, 254, 198],
 ...,
 [198, 253, 242, ..., 238, 253, 198],
 [199, 251, 251, ..., 252, 251, 199],
 [143, 179, 179, ..., 178, 179, 144]],

[[127, 131, 159, ..., 124, 118, 128],
 [113, 128, 160, ..., 116, 112, 120],
 [ 95,  86,  96, ..., 118, 115, 115],
 ...,
 [110, 115, 123, ..., 161, 171, 174],
 [110, 114, 121, ...,  71, 133, 172],
 [112, 110, 118, ...,  41,  36,  80]],

[[ 20,  19,  22, ...,  79,  78,  78],
 [ 22,  20,  21, ...,  80,  80,  80],
 [ 23,  23,  26, ...,  81,  81,  81],
 ...,
 [  8,  19,  50, ...,  25,  23,  22],
 [ 10,  45,  68, ...,  21,  21,  20],
 [  6,  38,  68, ...,  18,  17,  17]]], dtype=uint8)
```

Each row has 224 L channel values in Grayscale and there are 224 rows.

```
In [12]: l.shape
Out[12]: (25000, 224, 224)
```

Labels:

The labels are CIE Lab Model ‘a’ and ‘b’ channel values of the corresponding grayscale images. There are totally 25000 CIE Lab images present in this label. But only ‘a’ and ‘b’ channel values are only present as L channel value comes from grayscale images. Each CIE lab model image is of size 224 x 224 with ‘a’ and ‘b’ channel values in each pixel.

SNAPSHOT:

One image entry

```
array([[[[129, 136],
         [126, 134],
         [126, 135],
         ...,
         [125, 132],
         [124, 134],
         [124, 135]],

        [[127, 137],
         [126, 136],
         [126, 133],
         ...,
         [127, 131],
         [125, 133],
         [123, 137]],

        [[127, 138],
         [127, 134],
         [127, 134],
         ...,
         [125, 131],
         [126, 131],
         [128, 133]],

(i)      ...,

       [[126, 131],
        [126, 131],
        [127, 131],
        ...,
        [126, 130],
        [126, 130],
        [126, 130]],

       [[126, 131],
        [127, 131],
        [126, 132],
        ...,
        [126, 130],
        [126, 130],
        [126, 130]],

       [[127, 131],
        [127, 133],
        [128, 135],
        ...,
        [126, 130],
        [124, 130],
        [126, 130]]],
```

```
[[[121, 124],
  [121, 125],
  [121, 126],
  ...,
  [141, 125],
  [145, 125],
  [144, 125]],

 [[119, 125],
  [122, 124],
  [123, 124],
  ...,
  [138, 123],
  [138, 124],
  [142, 126]],

 [[123, 121],
  [116, 119],
  [115, 118],
  ...,
  [135, 106],
  [138, 113],
  [139, 121]],

 ...,
```

(ii)

```
[[125, 128],
 [124, 128],
 [124, 129],
 ...,
 [117, 126],
 [121, 126],
 [124, 125]],

[[127, 128],
 [127, 129],
 [126, 129],
 ...,
 [123, 125],
 [122, 126],
 [123, 125]],

[[127, 128],
 [127, 129],
 [126, 130],
 ...,
 [123, 126],
 [122, 126],
 [123, 126]]], dtype=uint8)
```

The data labels are stored in three different files.

```
In [14]: print(df1.shape)
         print(df2.shape)
         print(df3.shape)

         (10000, 224, 224, 2)
         (10000, 224, 224, 2)
         (5000, 224, 224, 2)
```

IMPLEMENTATION DETAILS

S.NO.	PACKAGE NAME	DESCRIPTION
1.	Numpy	For dealing with dataset elements and arrays
2.	Matplotlib.pyplot	For displaying images and graphs
3.	From skimage.color import lab2rgb, rgb2lab, rgb2gray	For converting CIE lab image -> RGB, RGB -> CIE LAB image and RGB to Gray scale respectively
4.	Torch	For building easily readable and GPU efficient ML models
5.	From torch.nn import nn	For importing Neural Network Model
6.	import torchvision.models as models	For importing models used for image classification, object detection, etc.
7.	from torchvision import datasets, transforms	For creating a Torch dataset, applying transformations like image resolution downscaling, etc.
8.	Os, shutil	For managing, creating and listing files and directories

MODULE SCREENSHOTS

MODULE 1: INPUT VIDEO PREPROCESSING

❖ **Description:**

The functionality of the module is to split an input gray scale video into its constituent gray scale frames.

❖ **Screenshot:**

```
Enter the video pathC:\Users\Guru Prasad\Documents\mlproject\videos\pexels-jill-burrow-6402550.mp4
---- Splitting Video Frames ----

Video File path is : C:\Users\Guru Prasad\Documents\mlproject\videos\pexels-jill-burrow-6402550.mp4
Splitting the video frames ....
Frame : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,
43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,8
5,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,12
0,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,1
52,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,
184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,21
5,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,2
47,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,
279,280,281,282,283,284,285,286,287,288,289,290,291,292,293,294,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,31
0,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,3
42,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,
374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,40
5,406,407,
Split completed.
Number of frames extracted from video : 406
Frames saved in location : pexels-jill-burrow-6402550/gray_scale_frames
```

MODULE 2: CNN MODEL PREDICTION

❖ **Description:**

The functionality of the module is to use the Trained CNN model to colorize the frames. The CNN model takes the Lightness channel value in Gray scale frames as and predicts a and b channel value of CIE Lab image model. The CIE Lab images are then converted into RGB images.

❖ Screenshot:

```
---- Colorizing Video Frames ----

Loading CNN Trained Model ... Model Loaded.
Colorizing the Frames
Frame: 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,4
3,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,8
5,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,12
0,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,1
52,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,
184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,21
5,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,2
47,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,
279,280,281,282,283,284,285,286,287,288,289,290,291,292,293,294,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,31
0,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,3
42,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,
374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,40
5,
Colorization completed
Total Number of colored Frames 406
Colored Frames stored in location : pexels-jill-burrow-6402550/color_frames/
```

MODULE 3: OUTPUT VIDEO PROCESSING

❖ Description:

The functionality of the module is to merge the list of colored frames into an output color video.

❖ Screenshot:

```
---- Merging Colored Frames ----

Starting to Merge Frames ...
Frame : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,
43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,8
5,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,12
0,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,1
52,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,
184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,21
5,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,2
47,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,
279,280,281,282,283,284,285,286,287,288,289,290,291,292,293,294,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,31
0,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,3
42,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,
374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,40
5,
Merging Completed.
Final video PROJECT.AVI released.
Video file stored in location : C:\Users\Guru Prasad\Documents\mlproject\project.avi

Execution Completed.
```

Input Gray scale frame vs Output Color Frame

PERFORMANCE MEASURES

1. Speed of Colorization:

Calculate the average time required to colorize a frame (image) of a video.

```
Enter the video pathC:\Users\Guru Prasad\Documents\mlproject\videos\Pexels Videos 1879456.mp4
Total Frames : 786
Total Time required to colorize all the images in the video: 123.30691480636597 s
Average Time required to colorize a frame: 0.15687902647120353 s

<Figure size 432x288 with 0 Axes>
```

2. Measurement of Correctness(i):

For regression, we quantify the closeness of the generated image to the actual image as the sum of the `2 norms of the difference of the generated image pixels and actual image pixels in the U and V channels: Lreg. = ||Up − Ua|| + ||Vp − Va||

```
Closeness of Image 0 : 14.592848
Closeness of Image 1 : 28.269836
Closeness of Image 2 : 13.674025
Closeness of Image 3 : 34.178337
Closeness of Image 4 : 21.08085
Closeness of Image 5 : 27.015827
Closeness of Image 6 : 32.195503
Closeness of Image 7 : 41.561493
Closeness of Image 8 : 14.019655
Closeness of Image 9 : 25.616985
Closeness of Image 10 : 13.54761
Closeness of Image 11 : 8.31824
Closeness of Image 12 : 38.395844
Closeness of Image 13 : 13.564478
Closeness of Image 14 : 31.789658
Closeness of Image 15 : 39.290203
Closeness of Image 16 : 30.270748
Closeness of Image 17 : 35.008446
Closeness of Image 18 : 12.017047
Closeness of Image 19 : 11.798961

Average Closeness 24.310329723358155
```

3. Semantic interpretability (VGG classification):

Does our method produce realistic enough colorizations to be interpretable to an object classifier? We test this by feeding our fake colorized images to a VGG network that was trained to predict ImageNet classes from real color photos. If the classifier performs well, that means the colorizations are accurate enough to be informative about object class.

```
Colorizing 20 images stored at location accuracy/
Image :0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,
Finished Colorizing
Colored Images stored at location accuracy_colored/
Loading Detection Model ... Finished Loading
```

	image	objects
0	0	zebra
1	1	horse
2	2	cup;bowl;dining table
3	3	person
4	4	person
5	5	person
6	6	clock
7	7	dog
8	8	chair;dining table
9	9	chair
10	10	bed
11	11	laptop
12	12	boat;person
13	13	train
14	14	airplane
15	15	car
16	16	chair
17	17	cow
18	18	vase
19	19	couch

```
Image 0
Prediction: ['zebra']
1 / 1 objects have been identified
Image 1
Prediction: ['horse']
1 / 1 objects have been identified
Image 2
Prediction: ['bowl', 'dining table']
2 / 3 objects have been identified
Image 3
Prediction: ['person', 'bottle', 'clock', 'bottle', 'person']
1 / 1 objects have been identified
Image 4
Prediction: ['person']
1 / 1 objects have been identified
Image 5
Prediction: ['person']
1 / 1 objects have been identified
Image 6
Prediction: ['clock', 'clock', 'clock', 'clock']
1 / 1 objects have been identified
Image 7
Prediction: ['dog']
1 / 1 objects have been identified
```

```
Image 8
Prediction: ['chair', 'chair', 'chair', 'chair', 'dining table', 'chair', 'dining table', 'chair']
2 / 2 objects have been identified
Image 9
Prediction: ['chair']
1 / 1 objects have been identified
Image 10
Prediction: ['bed']
1 / 1 objects have been identified
Image 11
Prediction: ['laptop']
1 / 1 objects have been identified
Image 12
Prediction: ['person', 'boat']
2 / 2 objects have been identified
Image 13
Prediction: ['train']
1 / 1 objects have been identified
Image 14
Prediction: ['airplane']
1 / 1 objects have been identified
Image 15
Prediction: ['car']
1 / 1 objects have been identified
Image 16
Prediction: ['chair']
1 / 1 objects have been identified
Image 17
Prediction: ['cow']
1 / 1 objects have been identified
Image 18
Prediction: ['vase']
1 / 1 objects have been identified
Image 19
Prediction: ['couch', 'vase', 'potted plant', 'chair']
1 / 1 objects have been identified
Out of 24 objects in 20 images, 23 objects where identified
Accuracy: 95.83333333333334
```

CONCLUSION / FUTURE SCOPE

In this project, a Convolutional Neural Network (CNN) model has been built to turn gray scale videos into colored videos. The gray scale video is split into frames and stored in a temporary storage. The CNN model predicts a and b channels of The CIE Lab image model when a gray scale frame Lightness channel is given as input. CIE Lab images are then converted into RGB images. The RGB frames are merged to produce the color video.

In order to take the performance of the model to the next step, FUSION NETWORK can be built where colorizer and classifier networks are seperated. By inputting the model with a 'object with context', the model can colorize it with less number of training steps and errors and the implementation part can be enhanced in order to prevent flickering results.

REFERENCES

Aditya Acharya, Anand Gopi, Karthik Iyer and Dr.Lakshmisudha Kondaka, "Implementation of Deep Learning Model for Video Colorization", 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pg.235-239, doi:10.1109/ICESC48915.2020.9156 047
