**IMAGE AND VIDEO COLORIZATION SYSTEM**

***A project report submitted in partial fulfillment of the requirements***

***for the award of the degree of***

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

BY

**Gaurav Sharma(BE/25052/16)**



**DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING**

**BIRLA INSTITUTE OF TECHNOLOGY, MESRA**

**JAIPUR CAMPUS, JAIPUR**

**MO-2019**

**DECLARATION CERTIFICATE**

This is to certify that the work presented in the project entitled “Image And Video Colorization System**”** in partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering in Computer Science and Engineering of Birla Institute of Technology, Mesra, Ranchi, Extension Center Jaipur is an authentic work carried out under my supervision and guidance.

To the best of my knowledge, the content of this project does not form a basis for the award of any previous degree to anyone else.

Date:

Dr. Madan Mohan Agarwal

(Assistant Professor)

Birla Institute of Technology, Mesra ,Ranchi

Extension Centre,Jaipur

**ACKNOWLEDGEMENT**

We express our deep and sincere gratitude towards our mentor Dr. M.M Agarwal, Asst. Professor, Dept of Computer Science and Engineering, BIT Mesra, Jaipur Campus for assigning us the project **Image and video colorization system**. Without his constant guidance, experience and suggestions at the various stages of the project, it would not have been possible.

We also extend our sincerest gratitude to our Project faculty Dr. Vishwambhar Pathak,  Asst. Professor, Dept of Computer Science and Engineering, BIT Mesra, Jaipur Campus for his timely guidance.

Lastly, we would like to extend our sincerest gratitude to Dr. Madhavi Sinha, Incharge, Dept of Computer Science and Engineering, BIT Mesra, Jaipur Campus for her guidance and encouragement.

Team

Gaurav Sharma (BE/25052/16)

**ABSTRACT**

**Abstract**

Given a grayscale photograph as input, this project attacks the problem of predicting a plausible color version of the photograph. This problem is clearly under constrained, so previous approaches have either relied on significant user interaction or resulted in de-saturated colorizations. We propose a fully automatic approach that produces vibrant and realistic colorizations. We embrace the underlying uncertainty of the problem by posing it as a classification task and use class-rebalancing at training time to increase the diversity of colors in the result. The system is implemented as a feed-forward pass in a CNN at test time and is trained on over a million color images. Moreover, we show that colorization can be a powerful pretext task for self-supervised feature learning, acting as a cross-channel encoder. This approach results in state-of-the-art performance on several feature learning benchmarks.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| 1 | INTRODUCTION | 6 |
| 2 | OBJECTIVE | 6 |
| 3 | REQUIREMENT SPECIFICATION | 7 |
| 4 | DESIGN SPECIFICATION | 13 |
| 5 | IMPLEMENTATION | 17 |
| 6 | FIGURES | 22 |
| 7 | RESULT | 22 |
| 8 | CHALLENGES | 23 |
| 9 | FUTURE SCOPE | 23 |
| 10 | CONCLUSION | 23 |
| 11 | REFERENCES | 24 |
|  |  |  |

**1.Introduction**

Automated colorization of black and white images has been subject to much research within the computer vision and machine learning communities. Beyond simply being fascinating from an aesthetics and artificial intelligence perspective, such capability has broad practical applications ranging from video restoration to image enhancement for improved interpretability. Here, we take a statistical-learning-driven approach towards solving this problem. We design and build a convolutional neural network (CNN) that accepts a black-and-white image as an input and generates a colorized version of the image as its output; Figure 1 shows an example of such a pair of input and output images. The system generates its output based solely on images it has “learned from” in the past, with no further human intervention. In recent years, CNNs have emerged as the de facto standard for solving image classification problems, achieving error rates lower than 4% in the ImageNet challenge. CNNs owe much of their success to their ability to learn and discern colors, patterns, and shapes within images and associate them with object classes. We believe that these characteristics naturally lend themselves well to colorizing images since object classes, patterns, and shapes generally correlate with color choice.

**1.1 Artificial Intelligence**

Artificial intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using rules to reach approximate or definite conclusions) and self-correction.

**1.2 Artificial Neural Networks**

Artificial neural networks are one of the main tools used in machine learning. As the “neural” part of their name suggests, they are brain-inspired systems which are intended to replicate the way that we humans learn. Neural networks consist of input and output layers, as well as (in most cases) a hidden layer consisting of units that transform the input into something that the output layer can use. They are excellent tools for finding patterns which are far too complex or numerous for a human programmer to extract and teach the machine to recognize.

**2. Objective**

* Colorization of grayscale image using deep neural network.
* Building a more efficient model than RGB for Image colorization
* Extending the Image model for video colorization.

**3. Requirement Specification**

**3.1 Introduction**

**3.1.1 Purpose**

Given a grayscale photograph as input, this project attacks the problem of hallucinating a plausible color version of the photograph. We propose a fully automatic approach that produces vibrant and realistic colorizations.

**3.1.2 Document Conventions**

Font – Arial

Heading Size – 18 and bold

Sub heading – 14 and bold

Text body size - 11

CNN – Convolutional Neural Network

**3.1.3 Intended Audience**

This product can be used by researchers for developing a colorized version of some of the old, famous black and white Images like first flight Image.

It can act as a starting idea for Machine Learning Scientists to develop a real time Image Colorizer.

**3.1.4 Product Scope**

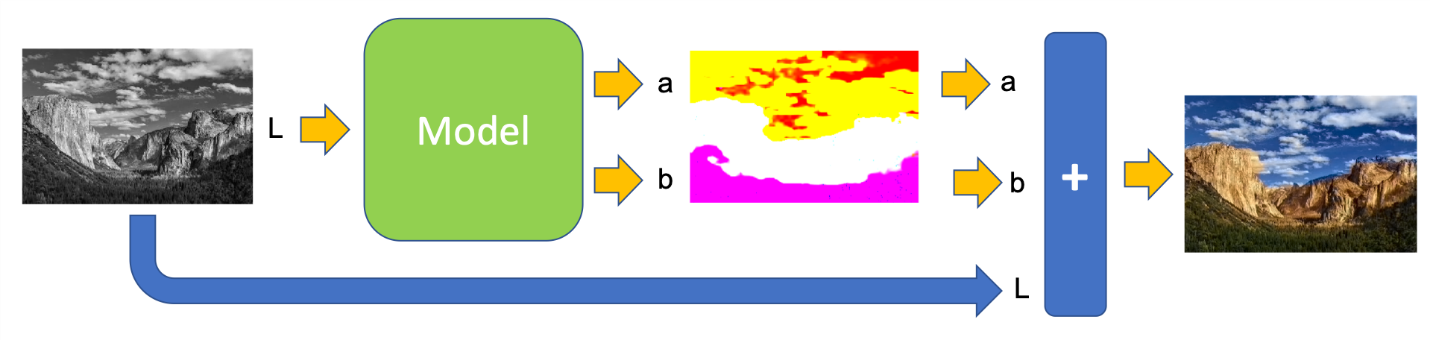
We propose a fully automatic approach that produces vibrant and realistic colorizations. This problem is clearly under constrained, so previous approaches have either relied on significant user interaction or resulted in de-saturated colorizations. The system is implemented as a feed-forward pass in a CNN at test time and is trained on over a million color images.

**3.2 Overall Description**

**3.2.1 Product Perspective**

This project is based on a research work developed at the University of California, Berkeley by Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful Image Colorization.

The idea behind this project is to develop a fully automatic approach that will generate realistic colorizations of Black & White (B&W) photos and by extension, videos.



**Figure : 1 General Diagram of System**

**3.2.2 Product Functions**

***The model that will be used on this project is the “Lab”.***

1. **Model Design** The CIELAB colour space (also known as CIE L\*a\*b\* or sometimes abbreviated as simply “Lab” colour space) is a colour space defined by the International Commission on Illumination (CIE) in 1976. It expresses colour as three numerical values, L\* for the lightness and a\* and b\* for the green–red and blue-yellow colour components. The colour space L \* a \* b \* was created after the theory of opposing colours, where two colours cannot be green and red at the same time, or yellow and blue at the same time. CIELAB was designed to be perceptually uniform with respect to human colour vision, meaning that the same amount of numerical change in these values corresponds to about the same amount of visually perceived change. Unlike the RGB colour model, Lab colour is designed to approximate human vision. It aspires to perceptual uniformity, and its L component closely matches human perception of lightness. The L component is exactly what is used as input of the AI model, that was train to estimate the remained components, “a” and “b”.
2. **AI Deep-Learning Process**

The Artificial Intelligent (AI) approach is implemented as a feed-forward pass in a CNN (“Convolutional Neural Network”) at test time and is trained on over a million colour images. In other words, millions of colour photos were decomposed using Lab model and used as an input feature (“L”) and classification labels (“a” and “b”). For simplicity split in two: “L” and “a+b”. Having the trained model, we can use it to colorize a new B&W photo, where this photo will be the input of the model or the component “L”. The output of the model will be the other components “a” and “b”, that once added to the original “L”, will return a full colorized photo.

**3.2.3 Operating Environment**

Operating System : Any

Python (version 3.6.8)

Keras(Python’s Library)

Tensorflow(Python’s Library)

OpenCV(4.0)

Therefore, all the environments having Python and required packages installed of the mentioned version or above are capable of executing the software. If any other programming component is used then the corresponding IDE will be explicitly mentioned, and should be installed for that environment.

**3.2.4 Design and Implementation Constraints**

Some internal features of the product(like in-built python constructs and packages) may depend on the version installed; hence, appropriate environment must be installed.

The user and item matrices processing  will depend on allocated memory, stack depth etc. provided by the operating system, and the runtime environment.

The inbuilt algorithms of the product work on a predefined data format, and hence the dataset must be according to the format of the algorithm.

**3.2.5 Assumptions and Dependencies**

The size of the dataset will fit in the memory provided to the software by operating system.The appropriate Python Environment (in terms of version and corresponding update) should be installed on the end user computer.

If any other explicitly specified component is used in the software, then the appropriate IDE for that system should also be present on the end user computer.

* 1. **System Features**

**3.3.1 RGB to Lab conversion**

**3.3.1.1** **Description**

Since the project uses the Lab color model but the pre-trained model is trained for RGB color space, so RGB color model need to be converted to Lab model.

* + - 1. **Stimulus**

It will be done using RGB-CIELab forward transformation model specified by the International Commission on Illumination to convert RGB model to Lab model.

## **3.3.2 CNN implementation**

**3.3.2.1** **Description**

The model trained is to be used for predicting the color combination for the given grayscale image.

**3.3.2.2** **Stimulus**

AI implementation is designed a feed forward pass CNN in order to correctly predict the color combination for the input images.

Preprocessing of the model is require in order to achieve a prefect model.

**3.3.2.3 Functional Requirement**

A fast computer is required with PYTHON 3.6 in order to train model.

**3.3.3 Video Extension**

**3.3.3.1 Description**

Using the model trained for image on videos

**3.3.3.2 Stimulus**

Extending the image model to work for videos. Model trained for image colorization is used on video frame by frame. Each frame of video is colorized separately and colorized video is stored.

**3.4 Other Nonfunctional Requirements**

**3.4.1 Performance Requirements**

The system should be able to handle large dataset about a million records). The output should be available for a user in a short time (3-4 seconds) after training and pre-processing.

**3.4.2 Safety Requirements**

While the software processes or executes any particular task, the user must wait for the existing task to complete.

**3.4.3 Software Quality Attributes**

The software cannot afford any unexpected failures. In order to avoid any failures occurrence the specifications have been respected and followed diligently. The following attributes have been taken into considerations:

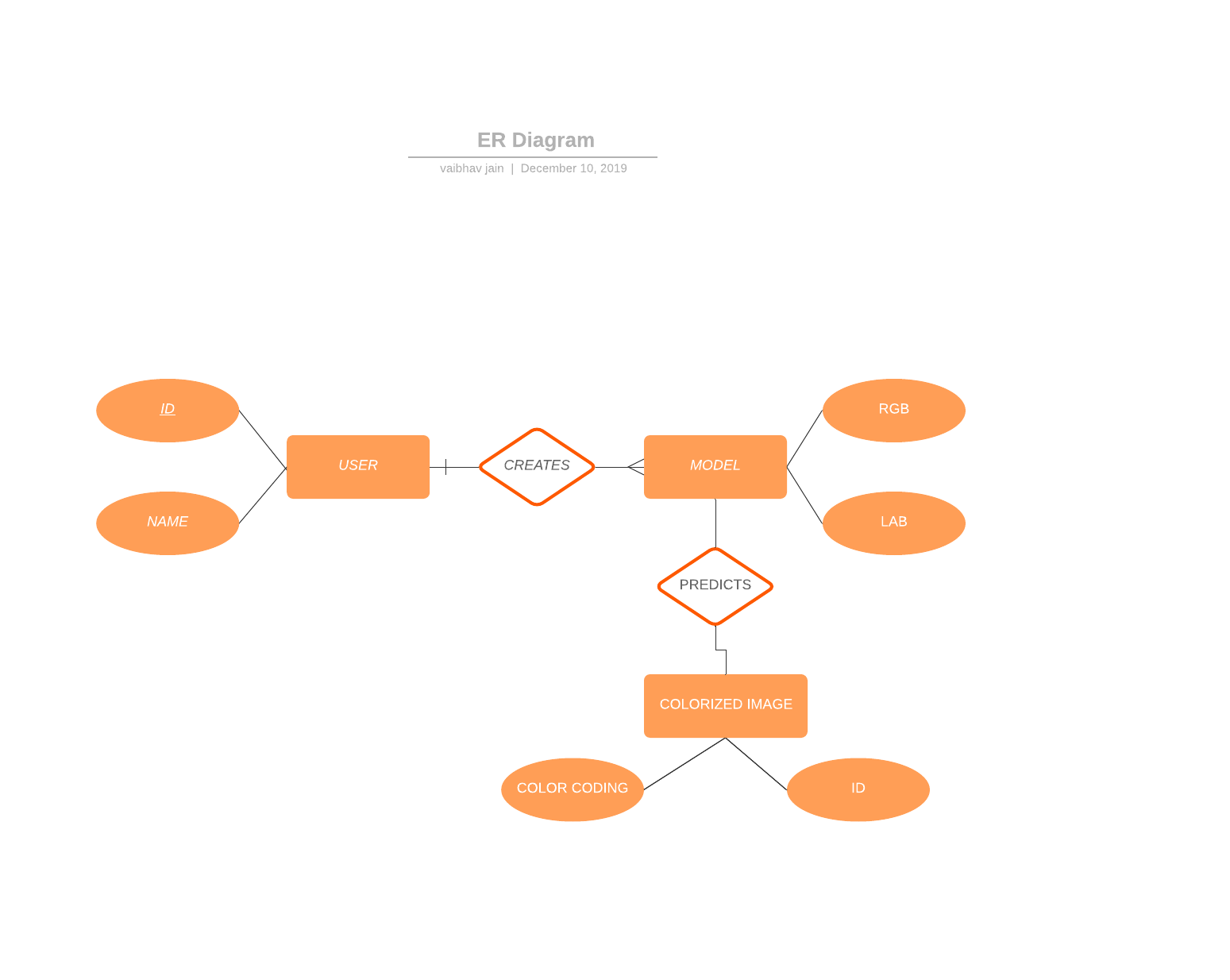
**Maintainability**: the code has been written following the standard coding practices so as to ensure that any changes and/or improvements can be made easily and when required

**Reliability**: the system has been designed to provide maximum reliability.

**Flexibility**: The layout/architecture of the system will be flexible enough for some later requirements change or application enhancement.

**Usability**: usability is really simple, the user only needs to upload the image and click on predict in order to get the verification result.

**3.5 ER- Diagram**



**Figure 2 : ER-Diagram of System**

**4. Design Specifications**

**4.1 Design Considerations**

**4.1.1 Assumptions and Dependencies**

The project is intended to process only black and white images

The training images are created using images from internet.

If the image resolution format is changed then the model needs to be retrained

**4.1.2 General Constraints**

The application can only run on a system that supports python 3.6.8 and above

The system must also have SciKit and Keras packages installed

System must also have OpenCV 4.0 and above.

**4.1.3 Goals and Guidelines**

The product has two major components in the model, that are model training and colorization.

Each image is expected to be processed and colorized in less than 10 seconds

**4.1.4 System Environment**

The application is intended to work on Windows/Linux operating system

**4.1.5 Risks and Volatile Areas**

None that we are aware of

**4.2 Architectural Strategies**

Python is used to develop the model with Keras ,Tensorflow and OpenCV.

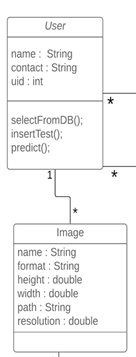
Convolutional Neural Network (CNN) are the basis of developing this computer vision model to verify the signature

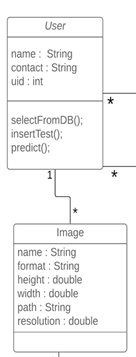
* Python is used to develop the model with keras ,tensorflow and OpenCV.
* Convolutional Neural Network (CNN) are the basis of developing this computer vision model to predict the color combination.
* The model is developed using CEILAB color space instead of RGB color space because LAB color space includes the brightness factor and therefore it resembles the human eyesight more accurately than an RGB model.
* The model is planned to be extended on video colorization.

**4.3 Data Model and Description**

**4.3.1 Data Description**

**4.3.1.1 Data Objects**





**Figure 3: Class Diagram of System**

This subsection of the document explains system's classes and their relations with each other. Most of the system functionalities are represented in Figure 1.

**User**: This class represents the user entity and it keeps a uid, name, contact. User selects image from database using selectFromDb(), insert image for testing using insertTest() and clicks on predict() to get the verification result.

**Image**: This class represents the image entity and it has the fields namely name, format, height, width, path and resolution.

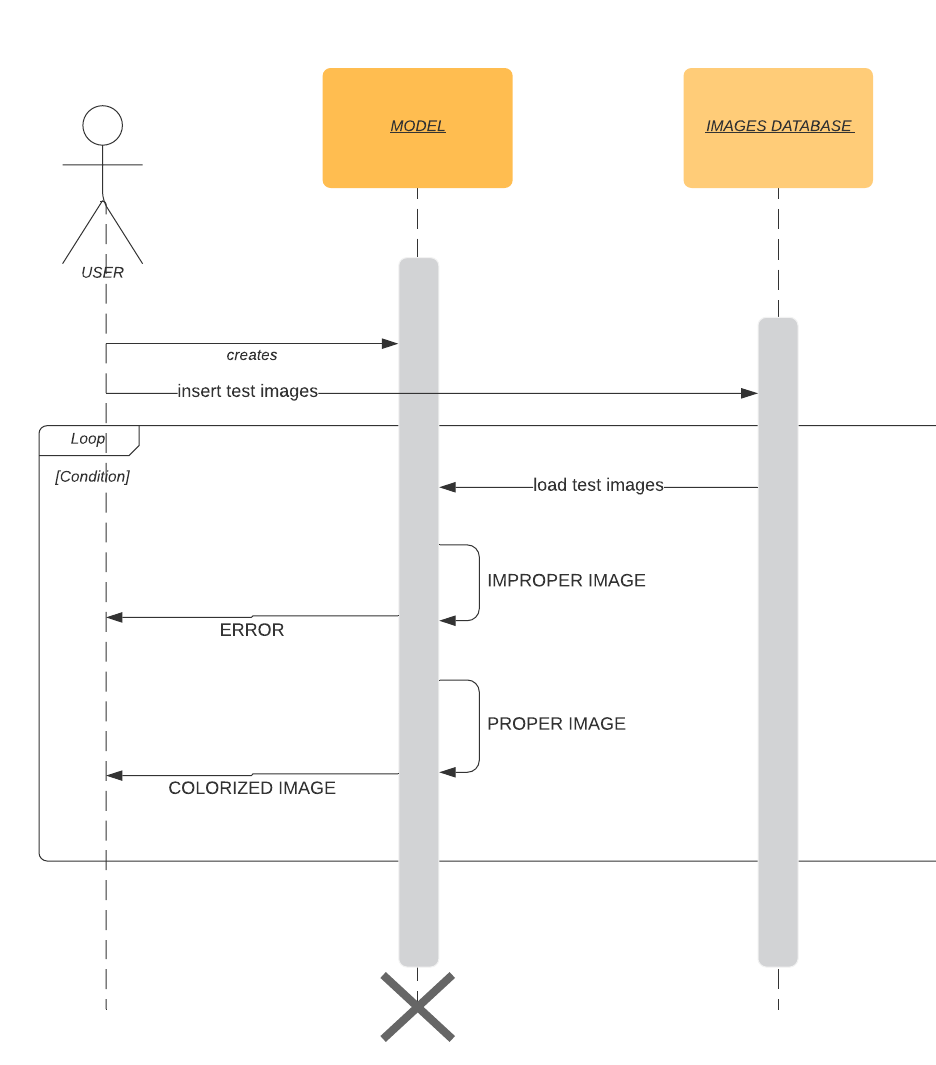
**4.3.1.2 Data Dictionary**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| id | This is unique id for every user |
| Img/Video | Black and white Image/Video |
| path | The path of image present in database |
| format | The format of image present in database. For e.g. .png, .jpeg , .jpg etc |
| resolution | Pixel count of the image present in database. Eg. 255 x 255 , 500 x 500 etc. |

**Table 1 : Data Dictionary Table of the System**

**4.4 Sequence Diagram**

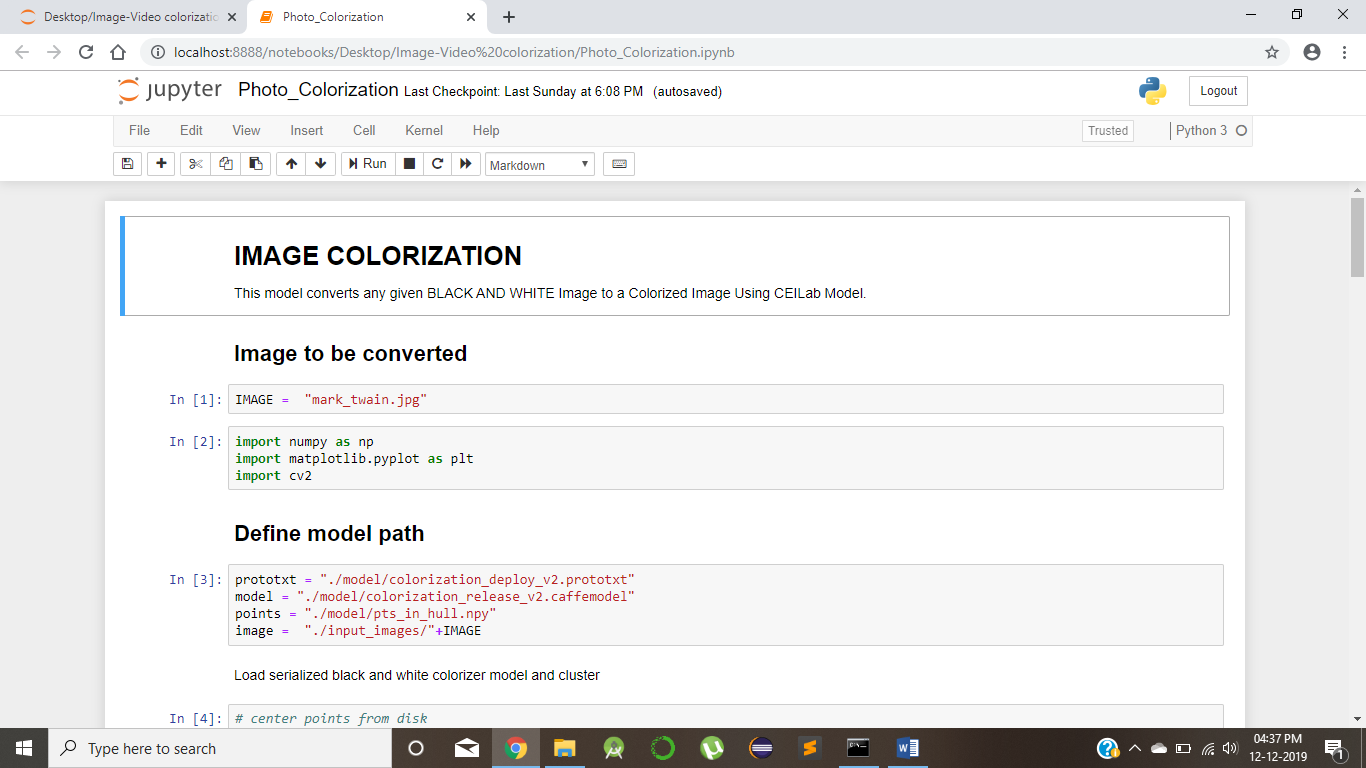
A **sequence diagram** simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms eventdiagrams or event scenarios to refer to a sequencediagram. Sequencediagrams describe how and in what order the objects in a system function.

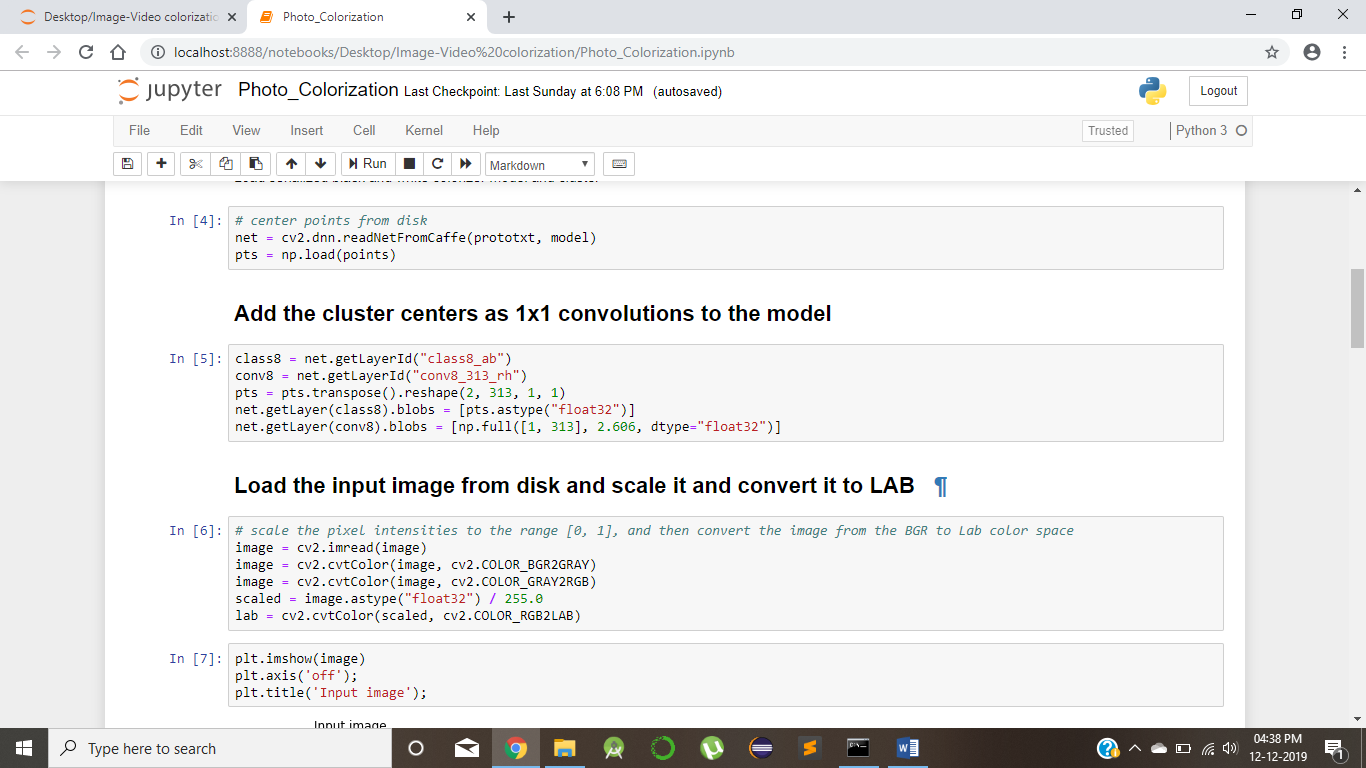


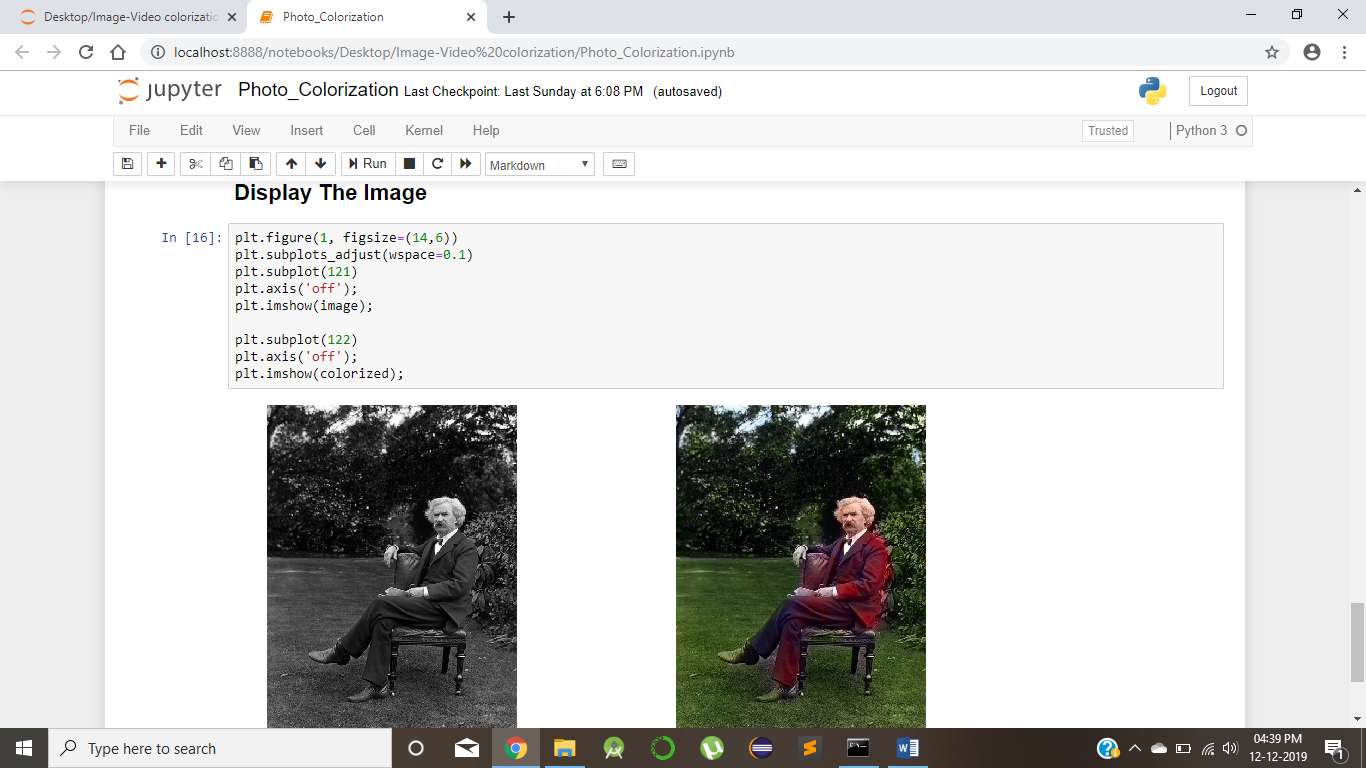
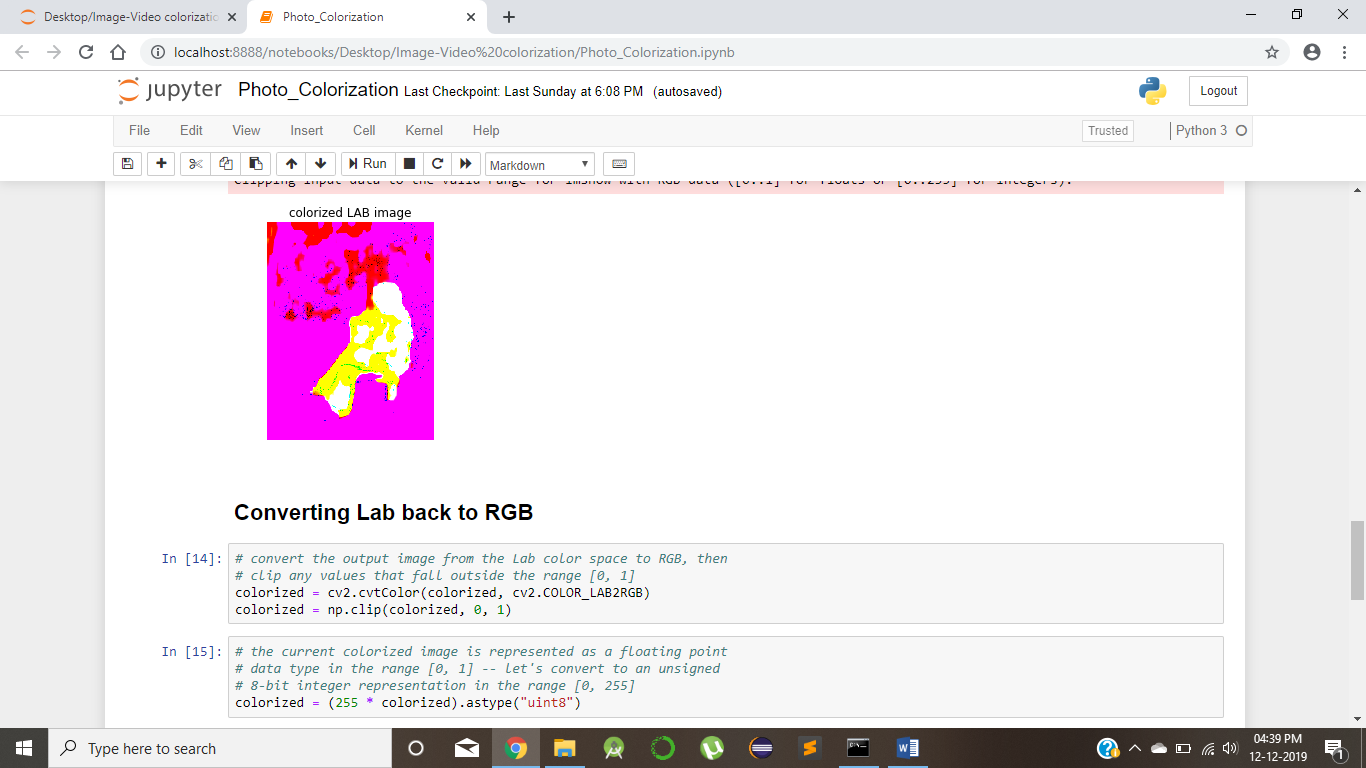
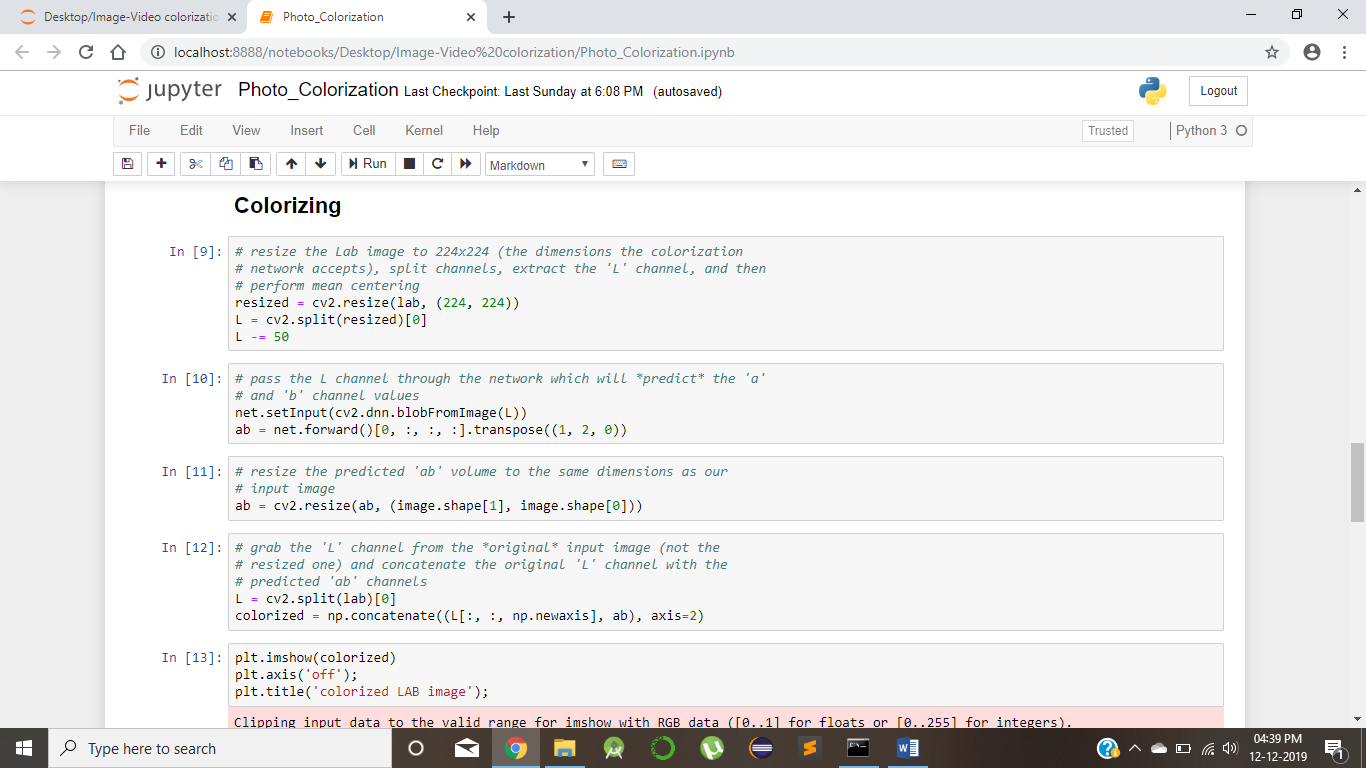
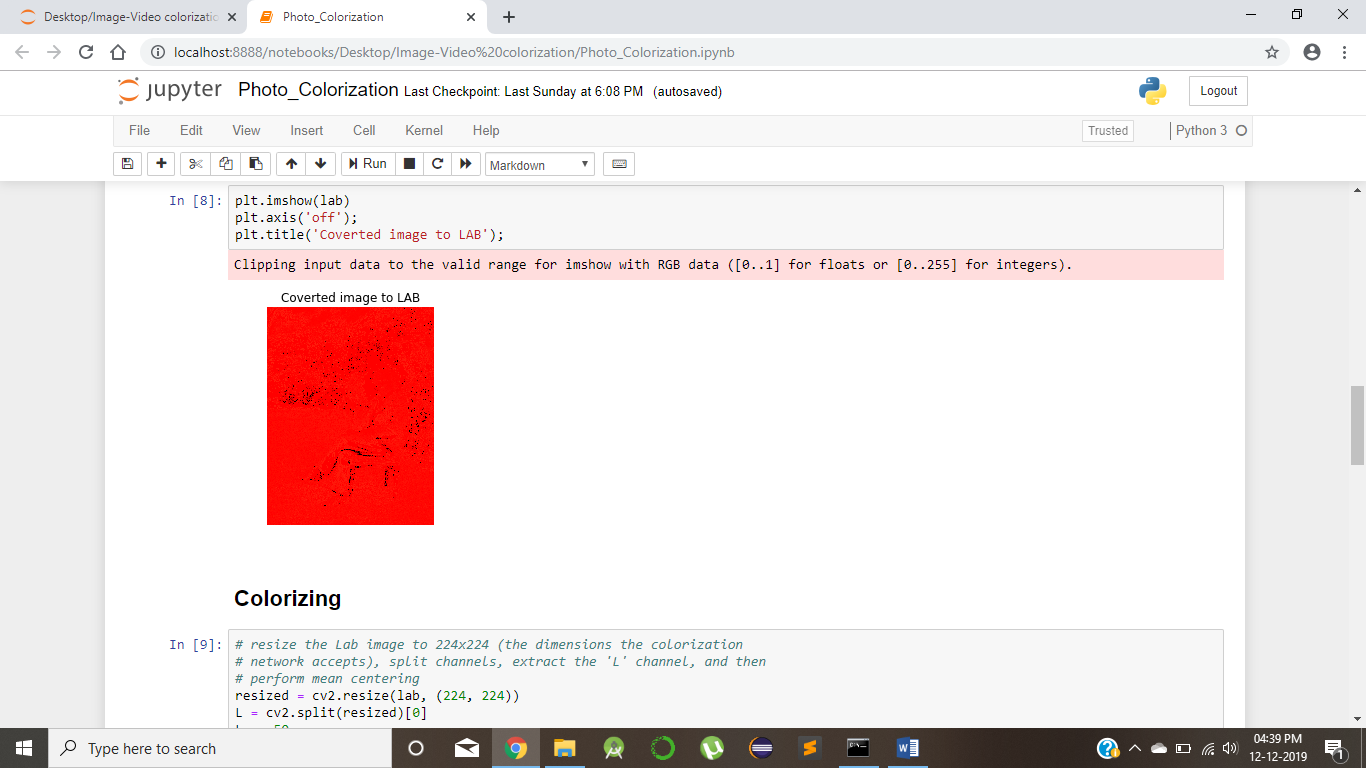
**Figure 4 : Sequence Diagram of System**

**5. Implementation**

* The application is based on ***Computer Vision*** and uses ***Convolutional Neural Network*** as its basic structure.
* First we created the data for the model.
* Images for model creation were taken from the internet.
* After Model creation the trained model is used for predicting color combinations for black and white image.
* Same model is used for colorizing video by loading video frame by frame and colorize each frame individually.

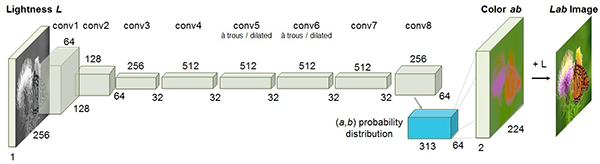
****

****

****

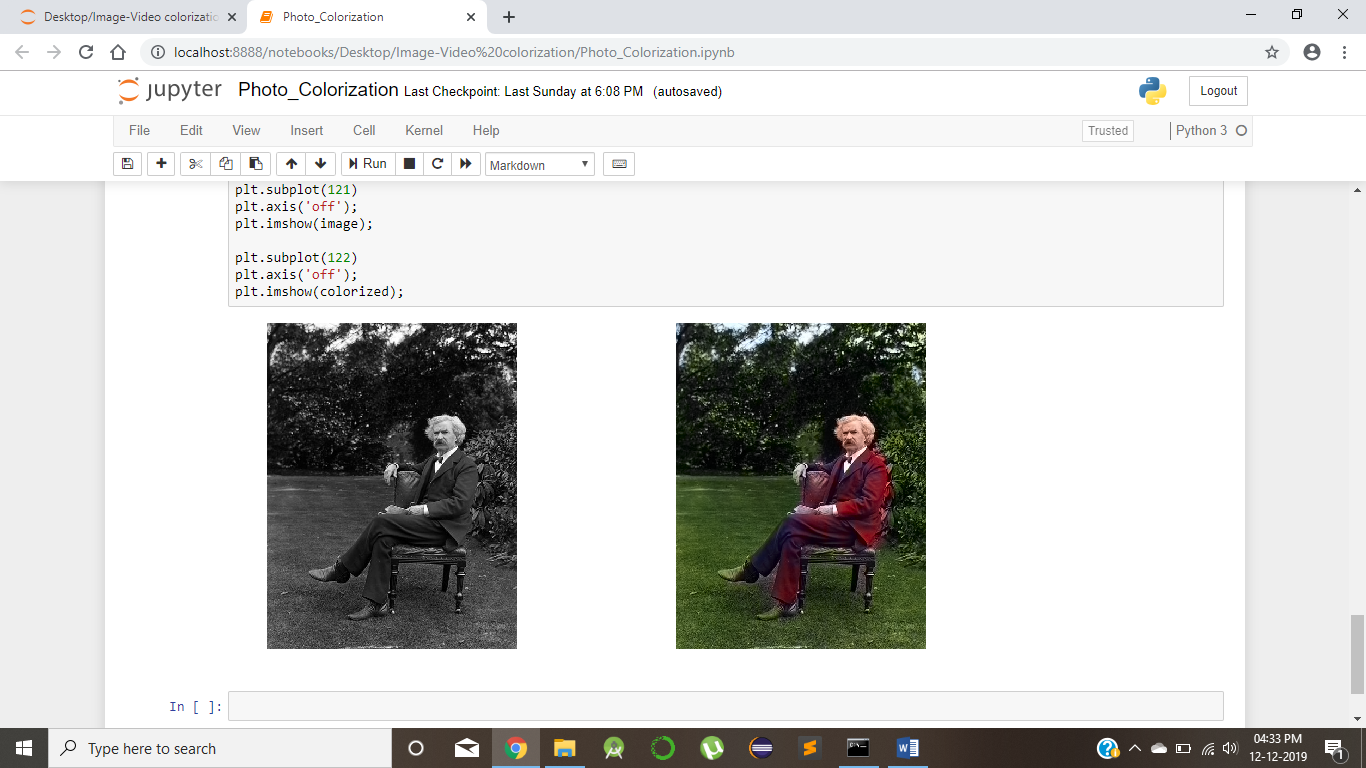
**6. Figures**

**6.1 Network Diagram**

**Figure 5 : Network Diagram of System**

**7. Results Obtained**

* The model of RGB and LAB color space were combined to produce an end to end model.
* The training data consisted of 47020 image pairs and validation data consisted of 3000 image pairs.
* The model is able to predict color combinations and is able to colorize the given black and white images.



**8. Challenges**

The main challenges faced during the course of this project are as follows :

**Data Creation** - The first challenege was to find the data of colorized images that would best the model proposed by the CEILAB color space. This problem was overcomed by creating our own dataset by including images of light colors that were absent in RGB model.

**Computational Need** - As it is required to use a GPU in order to train such a deep model with large dataset, and without it , the task is pretty much impractical. Therefore, we used Google Colab notebook for the purpose of using a GPU service.

**Video Extension** – Next challenging task was to extend the model trained for image to be capable of colorizing videos. For this we took video frame by frame and colorize each frame and store the result.

**9. Future Scope**

Our work therefore lays a solid foundation for future work. Moving forward, we have identified several avenues for improving our current system. To address the issue of color inconsistency, we can consider incorporating segmentation to enforce uniformity in color within segments. We can also utilize post-processing schemes such as total variation minimization and conditional random fields to achieve a similar end. Finally, redesigning the system around an adversarial network may yield improved results, since instead of focusing on minimizing the cross-entropy loss on a perpixel basis, the system would learn to generate pictures that compare well with real-world images. Based on the quality of results we have produced, the network we have designed and built would be a prime candidate for being the generator in such an adversarial network

**10. Conclusion**

Through our experiments, we have demonstrated the efficacy and potential of using deep convolutional neural networks to colorize black and white images. In particular, we have empirically shown that formulating the task as a classification problem can yield colorized images that are arguably much more aesthetically-pleasing than those generated by a baseline regression-based model, and thus shows much promise for further development

**11. References**

[1] Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: Proceedings of the IEEE International Conference on Computer Vision. 415–423

[2] R. Dahl. Automatic colorization. http://tinyclouds.org/colorize/, 2016.

[3] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In International conference on artificial intelligence and statistics, pages 249–256, 2010.

[4] Dahl, R.: Automatic colorization. In: http://tinyclouds.org/colorize/. (2016)

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015.

[6] M. J. Huiskes and M. S. Lew. The mir flickr retrieval evaluation. In Proceedings of the 1st ACM international con ference on Multimedia information retrieval, pages 39–43. ACM, 2008.

[7] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[8] D. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.