Project Report on

Image Colorization, Captioning, Enhancement, Auto Background Remove

at

U.V. Patel College of Engineering

Internal Guide
Prof. Manan Thakkar

Prepared By
19012531010 Alay Patel

B. Tech Semester VI
(Computer Engineering-AI)
June 2022

Submitted to,

Department of Computer Engineering-AI
U.V. Patel College of Engineering
Ganpat University, Kherva - 384012

U.V. PATEL COLLEGE OF ENGINEERING

07/06/2022

CERTIFICATE

TO WHOM SO EVER IT MAY CONCERN

This is to certify that **Mr. Alay Patel** student of B.Tech. Semester VI (Computer Engineering-AI) has completed her/his full semester on site project work titled "**Image Colorization, Captioning, Enhancement, Auto Background Remove**" satisfactorily in partial fulfilment of the requirement of Bachelor of Technology degree of Computer Engineering-AI of Ganpat University, Kherva, Mehsana in the year 2022-2023.

College Project Guide
Sign

Prof. Manan Thakkar

Head, Computer Engineering

Dr. Paresh M. Solanki

Acknowledgement

We sincerely feel the credit of the project work could not be narrowed down to only on individual. The development of this project involves many valuable contributions. Getting the opportunity for this project of "**Image Colorization, Captioning, Enhancement, Auto Background Remove**" as fulfilment of computer engineering-AI has been brightening experience for the near future to come and a focus on excellence in this venture, we are constantly guided and encouraged by **Prof. Manan Thakkar** who is our internal guide.

We would also thank our Head of the Department **Dr. Paresh M. Solanki for** giving us such a wonderful chance to work with this interesting project and perform the project work for the entire duration of the semester and also thanks to internal guide for project technical guidance and giving inspiration in all the way during project making. Finally, we would like to thank our parents, friends and almighty for being with us to support directly or indirectly while making this project.

Thank you, sir

Abstract

Captioning images automatically is one of the heart of the human visual system. There are various advantages if there is an application which automatically caption the scenes surrounded by them and revert back the caption as a plain message. It uses various pre-trained models to perform the task of detecting objects and uses CNN and LSTM to generate the captions. It uses Transfer Learning based pre-trained models for the task of object Detection. This model can perform two operations. The first one is to detect objects in the image using Convolutional Neural Networks and the other is to caption the images using RNN based LSTM(Long Short Term Memory). Interface of the model is developed using Django, which is a web development framework of python. The main use case of this project is to help visually impaired to understand the surrounding environment and act according to that. Caption generation is one of the interesting and focussed areas of Artificial Intelligence which has many challenges to pass on. Caption generation involves various complex scenarios starting from picking the dataset, training the model, validating the model, creating pre-trained models to test the images ,detecting the images and finally generating the captions.

Colorization of grayscale or b/w images is very useful to colorized our old memories. There are various advantages.it use various pre-trained models to perform colorization.it mainly depends on CNN. Till the date there is very few complex model exist with high accuracy. Image colorization involves various complex scenarios starting from making an architecture of a model ,training model, validating model, etc.

Table of Contents

List of Figure

List of Tables

1. Introduction

Image Captioning , Colouring, Enhancement , Auto Background Removing is powerful webapp which allows user to convert their b/w images into colour image also allow to generate description from given image. This web app totally depends on deep learning and machine learning algo like LSTM, CNN. image enhancement is improving the visual appearance of an image or to provide a "better transform representation for future automated image processing. It is a web-application so it can be accessible from any mobile with browser installed on it. It is running on basic technologies like HTML, CSS, JavaScript, Python–Django framework.

1.1 Purpose

The Purpose of Image Colorization, Captioning, Enhancement, Auto Background remover. Image colorization revived old memories in images. Image captioning or generating description we can use that in lot of area in AI. Image Colorization is the process of assigning colours to a grayscale image to make it more aesthetically appealing and perceptually meaningful. These are recognized as sophisticated tasks than often require prior knowledge of image content and manual adjustments to achieve artifact-free quality. The aim of image captioning is to automatically describe an image with one or more natural language sentences. The aim of image enhancement is improving the visual appearance of an image or to provide a "better transform representation for future automated image processing. many images like medical images, satellite images, aerial images and even real life photographs suffer from poor contrast and noise. it is necessary to enhance the contrast and remove the noise to increase image quality. enhancement is techniques which improve the quality of images for human viewing, removing blurring and noise, increasing contrast and revelling details are examples of enhancement operations.

1.2 Project Scope

The Software Requirements Specification captures all the requirements in a single document. The Image Colorization System that is to be developed provides the customer facility to convert Black & White image into color image without any other input. Other than that the Image Colorization System is supposed to have the following features:

a) Single image conversion
b) Batch image files conversion
c) Video conversion

1.3 Problem Statement

Colorization has been studied by researchers in the field of computer graphics and image processing since Wilson Markle colorized pictures from the Apollo space program in 1970. Adding colour to photographs by hand is a tedious process, which requires that the artist segment the image and then assign colours to each segment. The aim of our project is to design an algorithm and interactive system that automatically colorizes a monochrome image with human guidance. The algorithm takes a grayscale image and some colour scribbles drawn by a human and produces a fully colorized image that is both consistent with the scribble and the image semantics. While colorization itself has numerous applications such as image enhancement and film restoration, the unsupervised learning techniques that allowed us to add colour to images can also enable automatically learning information such as depth from an RGB image for applications such as autonomous driving. Aim is to create a deep learning model or algorithm to generate image with an high quality. Background removal is the most frequently used photo manipulation technique in post processing tasks. Using this technique, photo editors eliminate unnecessary and unwanted items and objects from photos to make them more attractive and outstanding. Aim is to create one deep learning model which take image as input and provide an output image by eliminating unnecessary objects.

1.4 Overview

This project going to be provided a lot of automation on image. First feature of our webapp is image colorization which is basically one technique to add style to a photograph or apply a combination of styles. Additionally, image colorization can add colour to photographs that were originally taken in black and white. Application of image colorization is colouring old b/w movies into colour movies. To solve such type of problem we used a diff types of CNN architectures like unet, resnet and pick one of the best suitable models for our web app second feature of our webapp is generating description from an image.it has lot of uses in AI field. Such as by generating a caption or description of an image. We could train one model that can predict a category of an image. such as group of people then category will be predicted people, dogs playing with ball then category will be predicted dog. To solve this type of a problem we created model using LSTM, CNN. Third feature of our webapp is basically a generating an image with better resolution from lower resolution.it also has an many helpful to a lot people as well as in field of an AI. For this we used pretrained model of OpenCV. OpenCV is huge, famous library of a computer vision. fourth and last feature of an our webapp is removing background automatically by using Deep learning concept like Classification, CNN. Basically, here we use and image segmentation. overall, our webapp is useful a lot of people and also a researcher. This project will save precious time to human life.

2. Literature Survey

2.1 Literature Survey for Image Colorization

Colorization basically involves assigning realistic colours to grey-scale image. Convolutional neural networks are specifically designed to deal with image data. Many authors have done promising work on this idea.

2.1.1 Domonkos Varga [1] proposed the idea of automatic colouring of cartoon images, since they are very different from natural images, they pose a difficulty as their colours depend on artist to artist. So, the data-set was specifically trained for cartoon images, about 100000 images, 70% of which were used in training and rest for validation. But unfortunately, the colon uncertainty in cartoons is much higher than in natural images and evaluation is subjective and slow.

2.1.2 Shweta Salve [2] proposed another similar approach, employing the use of Google's image classifier, Inception ResNet V2. The system model is divided into 4 parts, Encoder, Feature extractor, Fusion layer and Decoder. The system is able to produce acceptable outputs, given enough resources, CPU, Memory, and large data-set. This is mainly proof of concept implementation.

2.1.3 Yu Chen [3] proposed a approach to mainly address the problem of colouring Chinese films from past time. They used existing data-set with their data-set of Chinese images, fine- tuning the overall model. The network makes use of multi- scale convolution kernels, combining low and middle features extracted from VGG-16.

2.2 Literature Survey for Image Captioning

2.2.1 (RNN) in order to generate captions. In the last 5 years, a large number of articles have been published on image captioning with deep machine learning being popularly used. Deep learning algorithms can handle complexities and challenges of image captioning quite well. So far, only three survey papers have been published on this research topic. Although the papers have presented a good literature survey of image captioning, they could only cover a few papers on deep learning because the bulk of them was published after the survey papers. These survey papers mainly discussed template based, retrieval based, and a very few deep learning-based novel image caption generating models. However, a large number of works have been done on deep learning-based image captioning. Moreover, the availability of large and new datasets has made the learning-based image captioning an interesting research area. To provide an abridged version of the literature, we present a survey mainly focus on the deep learning-based papers on image captioning.

2.3 Literature Survey for Image Enhancement

2.3.1 One paper proposes to use adaptive region-based histogram equalization to improve the contrast of grayscale images . The paper uses the “Yale B” database, the “Extended Yale B” database and the “CMU-PIE” dataset, which contains a large number of face images with single-sided illumination and low intensity. The paper uses the “Euclidean distance nearest-neighbour” classifier to classify images. Experiments show that using an adaptive region-based histogram equalization algorithm can improve the accuracy of image classification to 100%. However, the “Euclidean distance nearest-neighbour” classifier has lower accuracy for image classification than the CNN model. This thesis also tried to use the datasets to experiment, but the CNN model performed very well on the

original dataset and can achieve 100% accuracy, so it can't be proved that image enhancement can help CNN to perform better on these datasets

2.3.2 Studies have shown that using bimodal image enhancement can greatly improve the accuracy of recognition when CNN is natural scene character recognition . Different bimodal image enhancements will impact on the accuracy of image recognition. The proposed bimodal image enhancement algorithm performs better than the traditional bimodal image enhancement algorithm on ICDAR 2003 Robust OCR dataset . The recognition accuracy rate from about 70% has increased to over 80%. Although the conclusion of the paper is obvious, the paper does not compare the recognition rate on the original data set. In addition, the ICDAR 2003 Robust OCR dataset contains a large number of low-quality images, while normal clear images account for only one-third of the data set. Under normal circumstances, the dataset of the CNN model is most the clear images. The accuracy of CNN with the bimodal image enhancement algorithm proposed in the paper is lower than the accuracy of CNN with traditional bimodal image enhancement. Therefore, the paper does not explain whether the image enhancement algorithm will affect the recognition rate of CNN on the normal image dataset.

2.3.3 In addition, one paper has shown that using image enhancement function with the Laplace operator can improve the performance of R-CNN and fast R-CNN in pedestrian detection task . The experiment in this paper compares the original metro crowd image dataset with the enhanced image dataset with Laplace operator. The result is Laplace operator can increase the detection rate of 2% and 1% in the two R-CNN models respectively. The paper's experiments used transfer learning, fine-tuning pre-trained R-CNN model. However, the experiment did not use cross validation and did not use hypothesis testing, which indicates that there is a high risk in the conclusion of the experiment. In addition, there is also a difference between target detection and image recognition.

2.4 Existing systems description

2.4.1 Imagecolorizer.com

Image Colorizer is our basic AI feature that could add colour to black and white pictures naturally. Sometimes, you wished you could get back old damaged photos or see them in colour; the unique features of the Image colorizer online tool can come to your rescue. These features are straightforward to use and save your time. A 95% satisfaction rate with more than 4000 plus visitors at the site implies that people like this online tool's features very much.

2.4.2 Hotpot.ai

Colorize pictures with AI, turning black and white photos to colour in seconds. Reimagine the past by colorizing pictures of ancestors and historic figures. Our technology is currently based on OpenCV Model and proprietary enhancements.

2.4.3 Colorizer.cc

Colorize.cc - is AI service which colorize black and white photo and video. In few clicks you can touch the magic of machine learning technologies. Colorize hundreds of your family photos in few minutes.

2.5 Problems with existing systems

- Paid Features.
- Watermarks in free version.
- Login required
- Limited times

2.6 Solution applied in our project

- Totally Free
- No Watermarks
- No login required
- No limited times

3. Methodology

We have trained image colorization using Keras library. Because of performance issue here we used pretrained model for image colorization and image captioning both. For image enhancement we have used the OpenCV super-resolution model. The models are integrated to the backend using a Django. Django is a web framework. This means Django provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website. Every time the client performs an action, it is done in the form of a request. All the requests managed by Django application, which also handles the frontend.

3.1 Tools and Technology

In this project we have used Python Programming Language along with CNN to work with Image Data.

We used the following tools to implement the project -

HTML + CSS (for frontend)
Python + libraries such as Django (for backend)
Keras
TensorFlow
JavaScript
OpenCV

- **Django**

Web framework defined for developing a web application which can be implemented on python.

- **JavaScript**

JavaScript is a lightweight, cross-platform, and interpreted compiled programming language which is also known as the scripting language for webpages. It is well-known for the development of web pages; many non-browser environments also use it.

- **Python**

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.

- **Open CV**

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

4. Software Requirements Specification (SRS)

4.1 Feasibility Study

Feasibility study can help you determine whether or not you should proceed with your project. It is essential to evaluate cost and benefit. It is essential to evaluate cost and benefit of the proposed system. Five types of feasibility study are taken into consideration.

- **Technical feasibility:** It includes finding out technologies for the project, both hardware and software. For our project, system needs internet connection. While using our webapp, make sure you have a steady internet connection. It is also not an issue in this era where almost every home or office has Wi-Fi.
- **Operational feasibility:** It is the ease and simplicity of operation of proposed system. System does not require any special skill set for users to operate it. In fact, it is designed to be used by almost everyone.
- **Economic feasibility:** Here, we find the total cost and benefit of the proposed system over current system. For this project, the main cost is documentation cost. As far as maintenance is concerned, free of cost
- **Organizational feasibility:** This shows the management and organizational structure of the project. That won't create any management issues and will increase the feasibility of the project.

4.2 Hardware Requirements

The software is designed to be light-weighted so that it doesn't be a burden on the machine running it. This system is being build keeping in mind the generally available hardware and software compatibility. Here are the minimum hardware and software requirement for our project.

DEVELOPER-Side requirement:

Hardware:

- Intel core i5 7th Generation.
- RAM 4GB or more.

Software:

- Windows 10(32-bit) or above.
- Python 3 or later.
- Django
- TensorFlow GPU
- OpenCV

CLIENT-Side requirement:

- Laptop or mobile phone with internet Connection

4.3 Software Requirements

4.3.1 User

4.3.1.1 Sign Up

Input: Field required to register are full name, last name, e-mail address, password, confirm password.

Process: As you will click on the signup button, it will take to a new page and in that you have to fill your full name, email address, password and all these details. And after click on signup button.

Output: Once register successfully through filling correct information it will take you to login page.

4.3.1.2 Login

Input: Field required to register are full name, last name, e-mail address, password, confirm password.

Process: As you will click on the signup button, it will take to a new page and in that you have to fill your full name, email address, password and all these details. And after click on signup button.

Output: Once register successfully through filling correct information it will take you to login page.

4.3.1.3 Home

Input: click on any one of the Feature you wanted to use

Process: After click on any one of the features of our webapp you will be redirected

Upload image page where user need to upload image.

Output: Automatically whatever user selected feature going to execute and give an output image

4.3.1.4 Upload Image

Input: User need to upload an image

Process: After uploading an image model will execute on that image and output will be shown on webapp.

Output: Result of model

4.4 Project Planning

Table 4.1.1 Project Planning

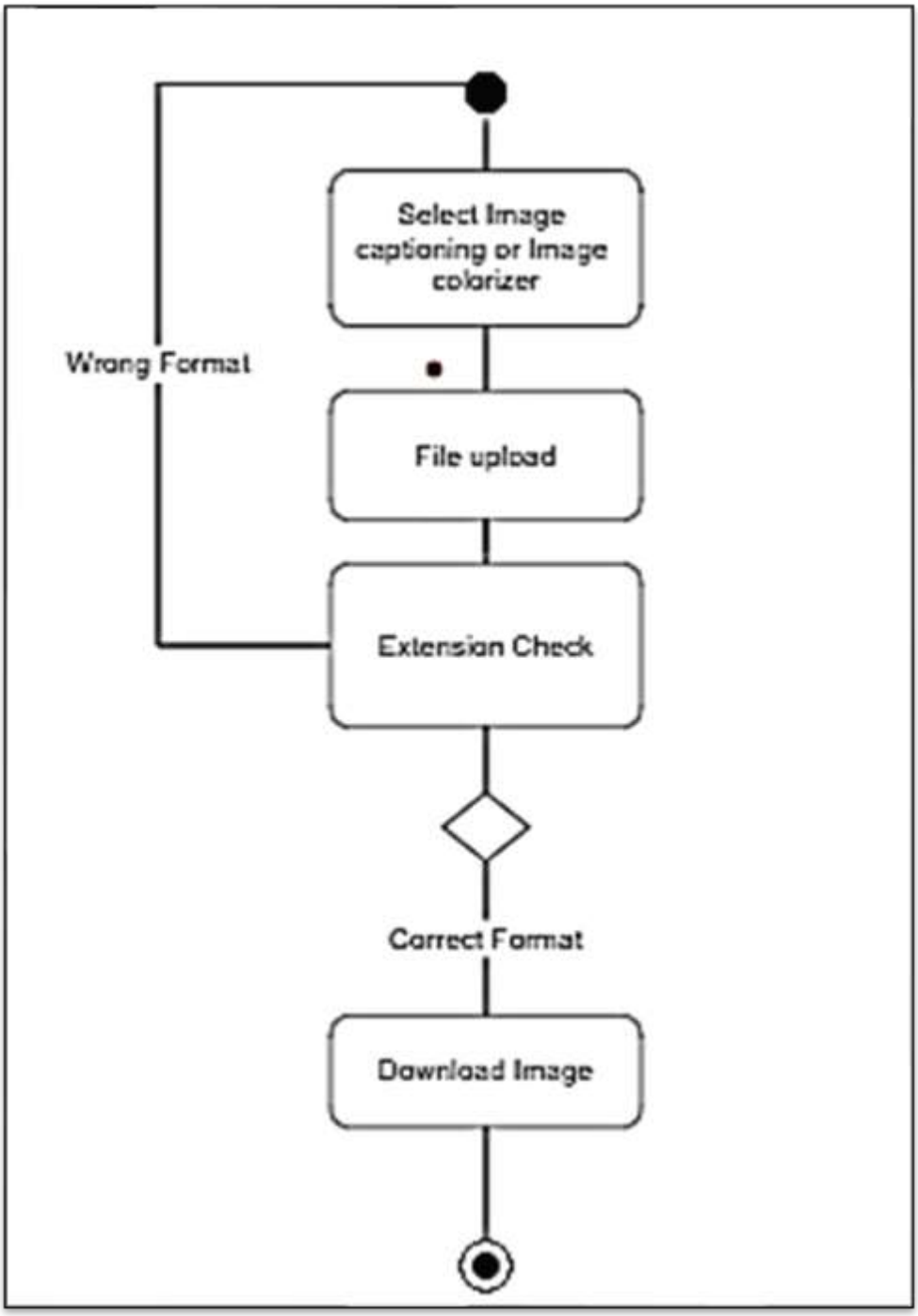
Sr. No	Task To Complete	Estimated Time
1	Project Initialization	4th Aug – 18th Aug
2	Documentation	18th Aug – 9th Sep
3	Requirements gathering	9th Sep – 23rd Sep

4	Estimation and Scheduling	23rd Sep
5	Design Analysis	5th Feb-19th Feb
6	Validate and Update	20th Feb – 16th March
7	Development	17th March – 30th May
8	Testing and Validation	1st June – 5th June

5. System Design

5.1 Activity Diagram

Figure 5.1 Activity Diagram



5.2 Class Diagram

Figure 5.2 Class Diagram

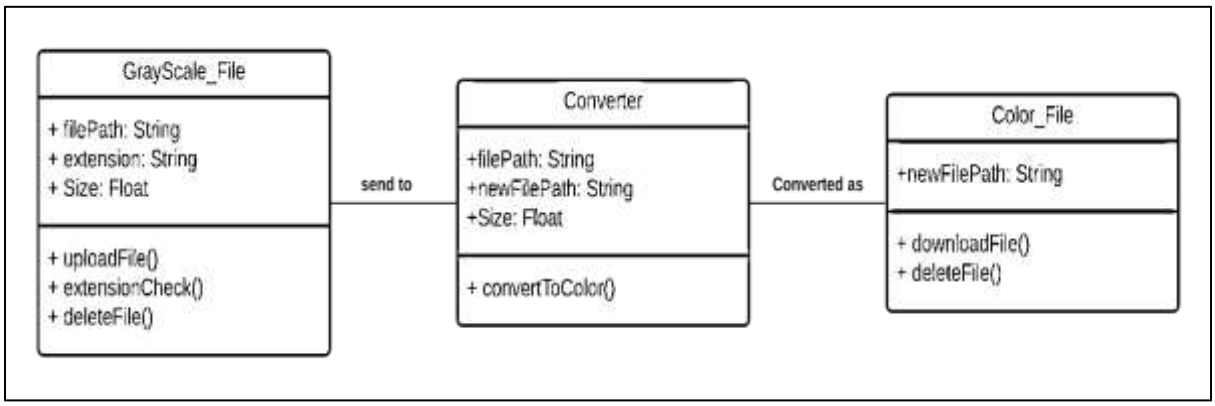
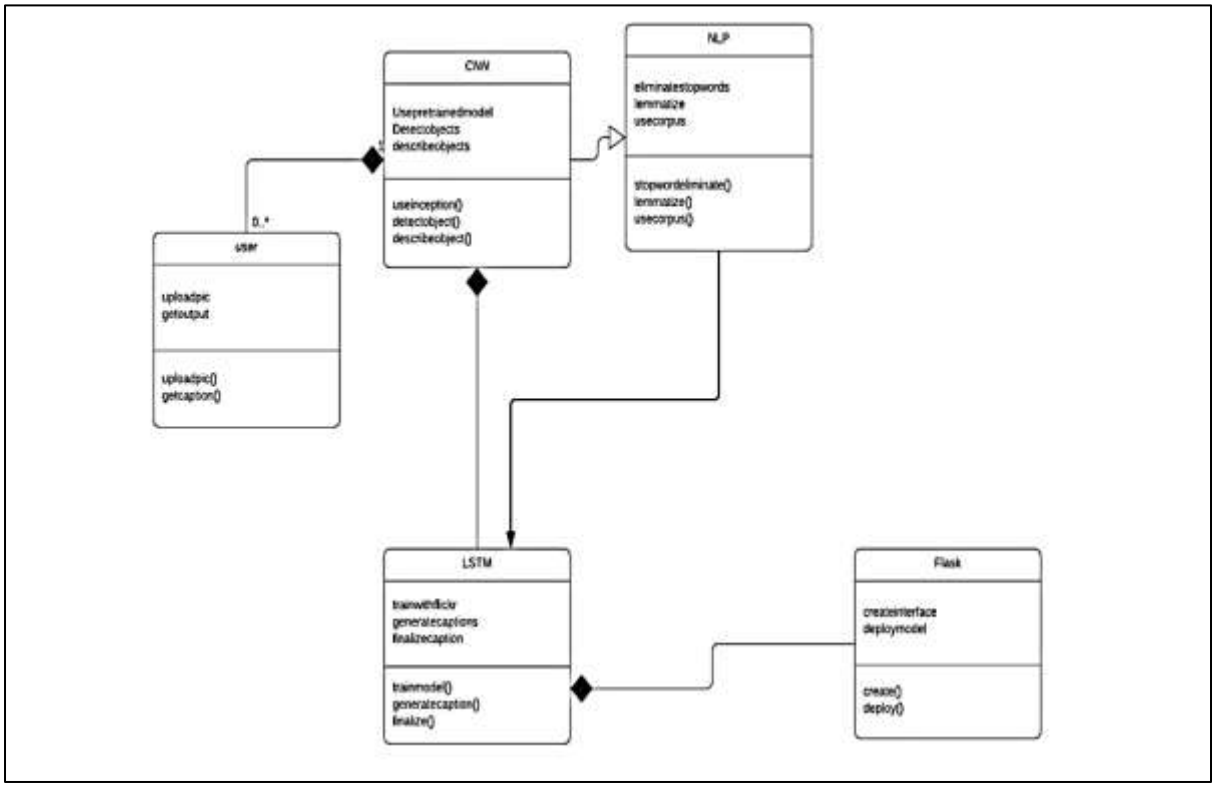


Figure 5.3 Class Diagram



5.3 Sequence Diagram

Figure 5.4 Sequence Diagram

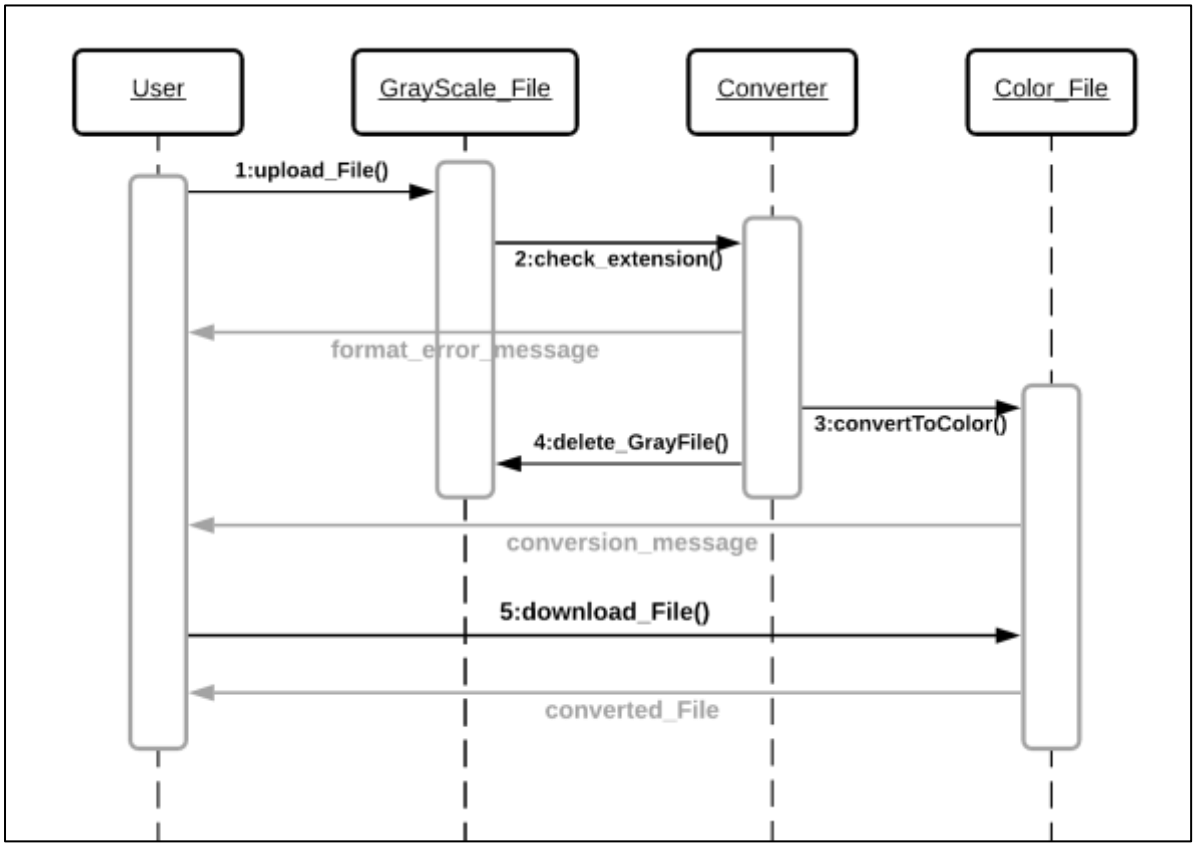
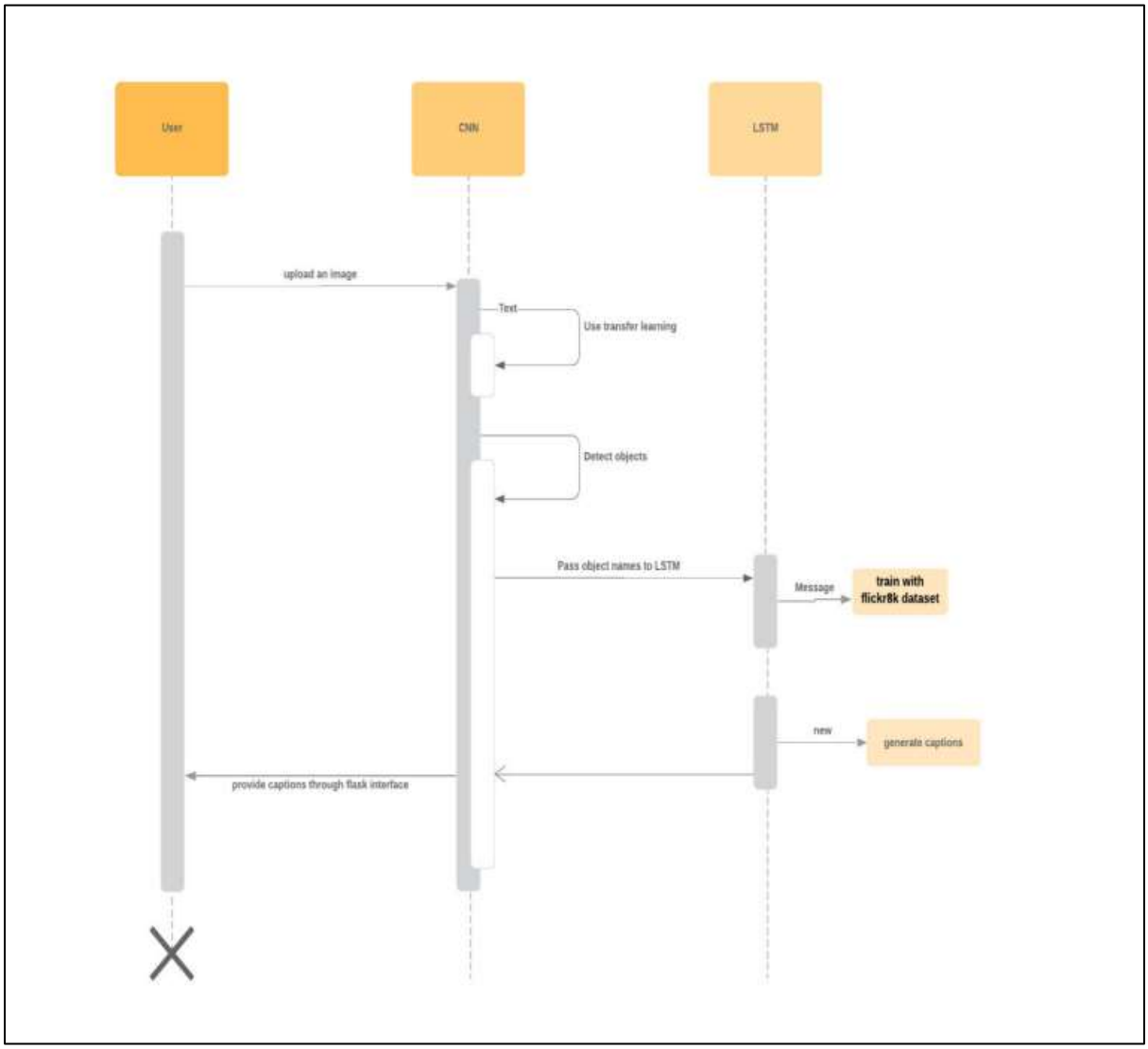
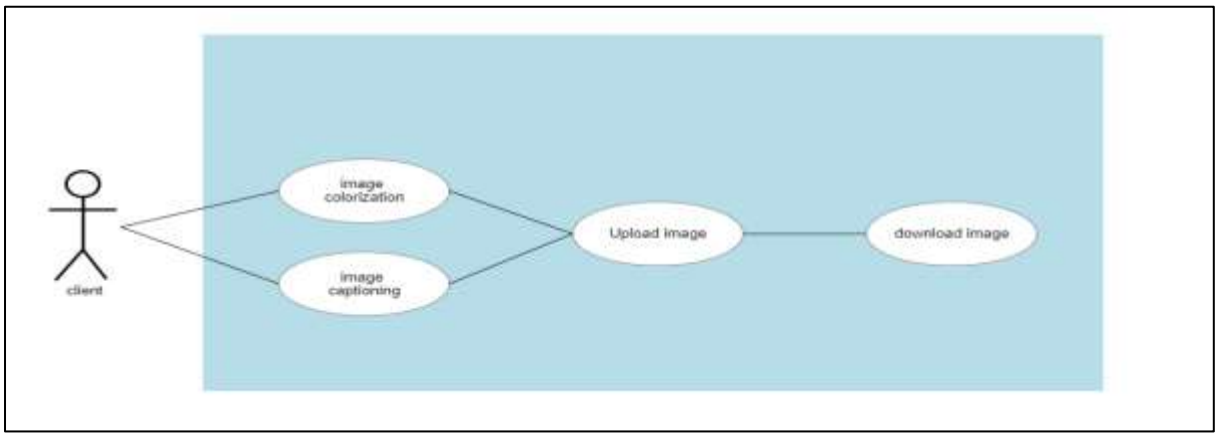


Figure 5. 5 Sequence Diagram



5.4 Use Case Diagram

Figure 5.6 Use case Diagram



5.5 Data Dictionary

Table 5.5.1 Data Dictionary

name	datatype	length	constraint	description
user_id	INT	10	Primary key	User id Auto Generate
username	VARCHAR	20	Not Null	Name of user
password	VARCHAR2	30	Not Null	Login password for user

6. Prototype

6.1 Login Page

Description: This is login page. Here user can login into their account

Figure 6.1 Login page

6.2 Signup Page

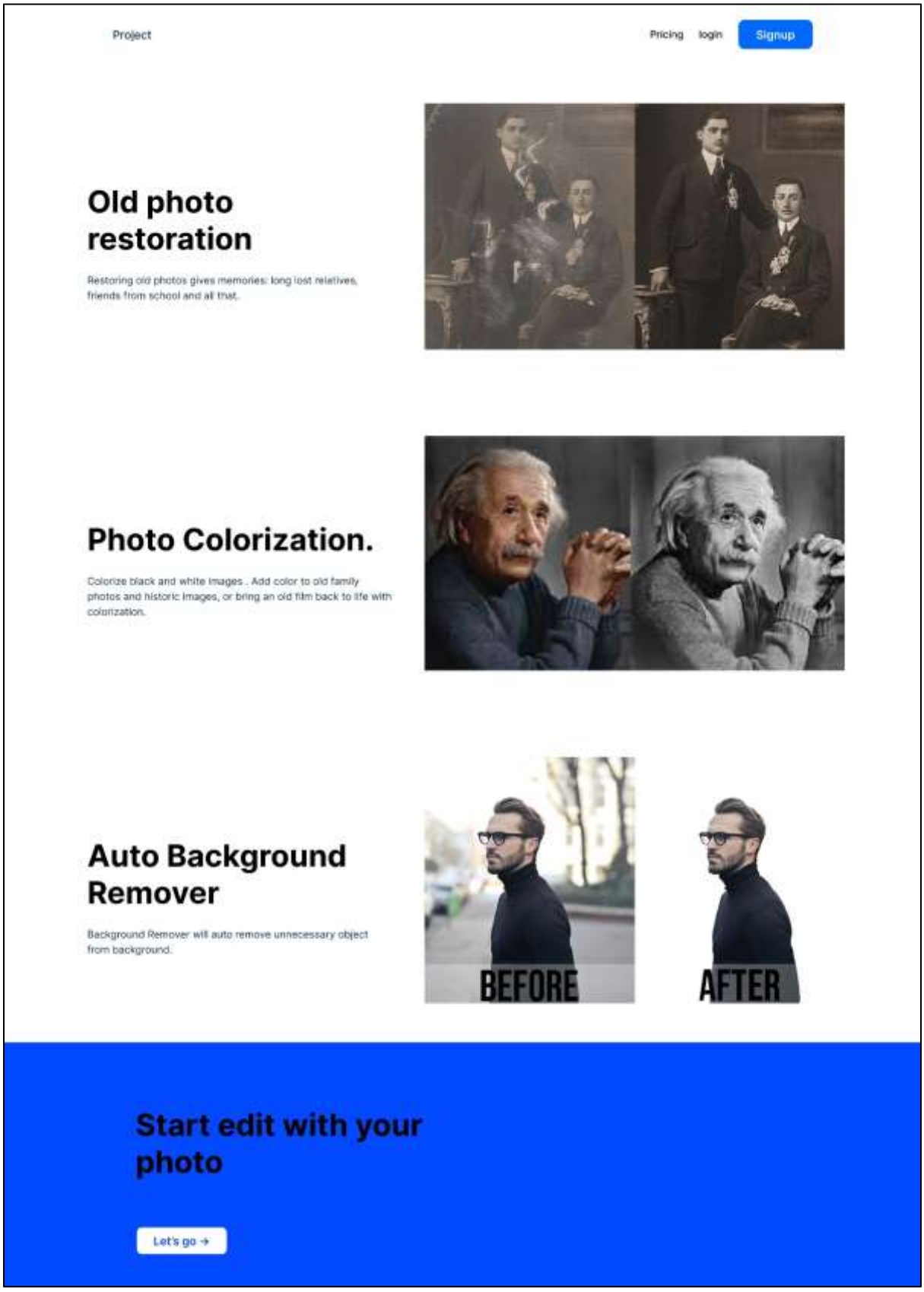
Description: This is signup page. Here user can register or create into their account

Figure 6.2 Signup page

6.3 Home

Description: This is home page. Here user can see all available features and select one of them

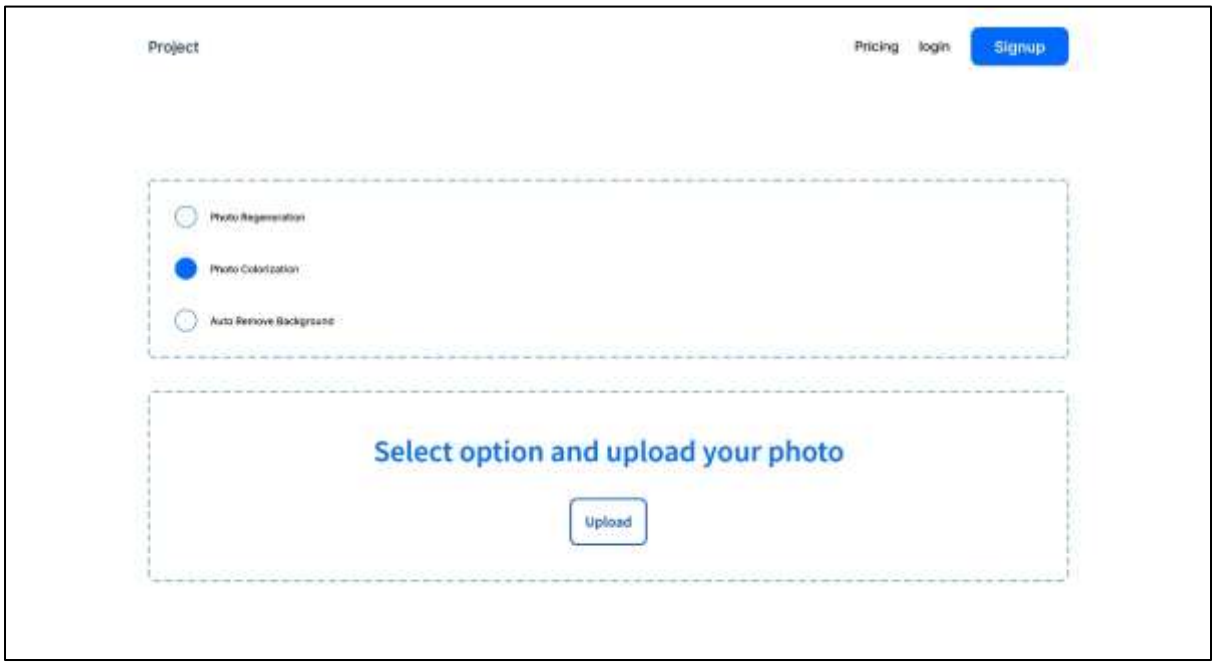
Figure 6.3 Home page



6.4 Upload Image

Description: This is next page after selecting feature. User can Upload image.

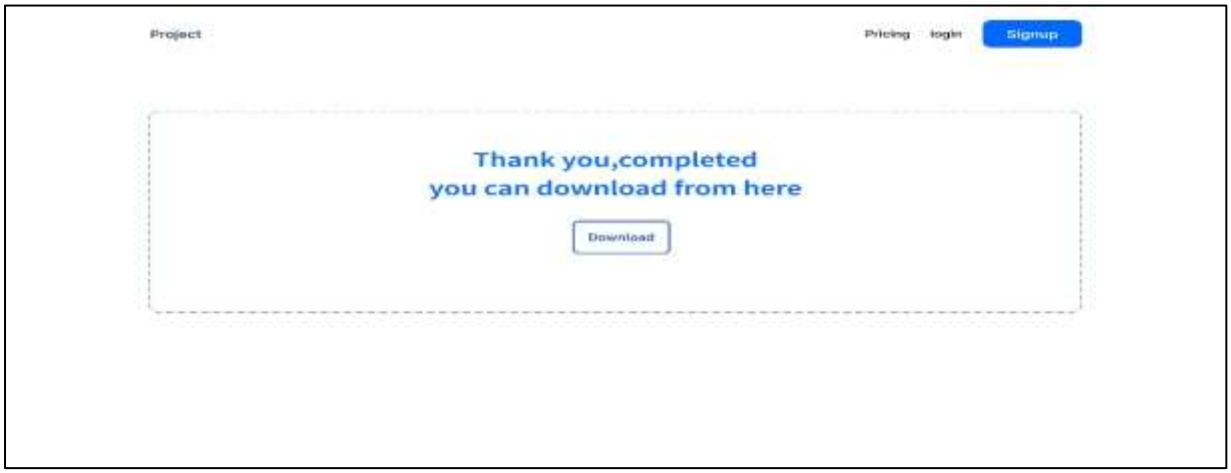
Figure 6. 4 Upload Image



6.5 Download Image

Description: This is final page where use can download output image.

Figure 6. 5 Download image



7. Implementation

This chapter include snapshot of working project created by me. Our webapp you can use without login into account. Consider user selected a photo colorization and click on that. After clicking “try this” button, user will be redirected to upload image page where user need to give permission and after that user need to upload image and after uploading image model will run in backend and then after sometime output will be shown in upload image page for now.

Figure 7.1 Home page

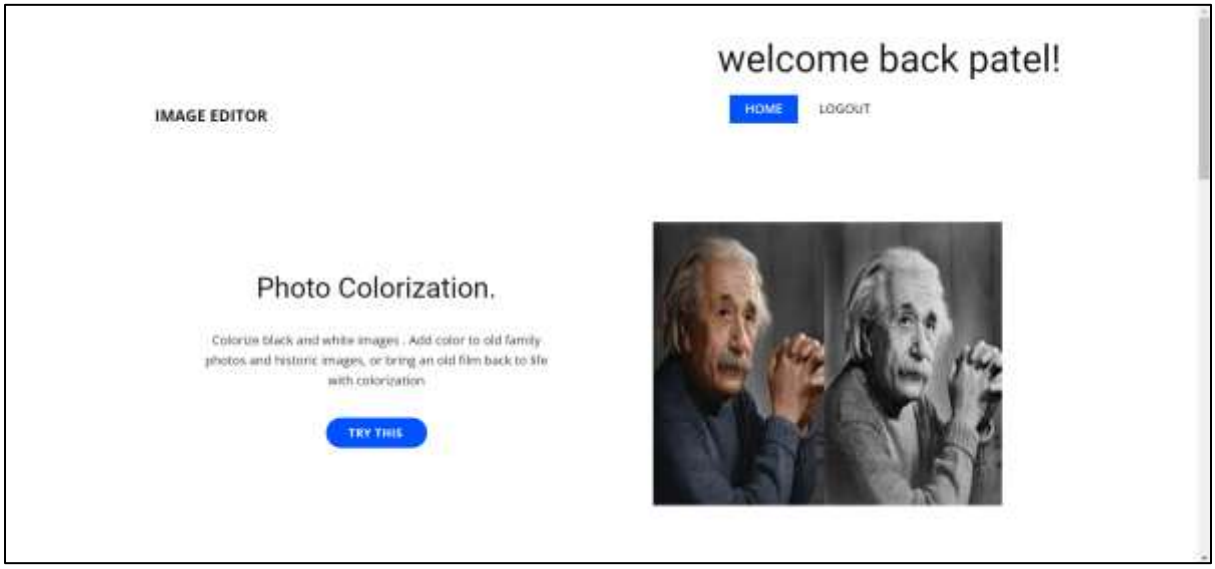


Figure 7. 2 Upload image

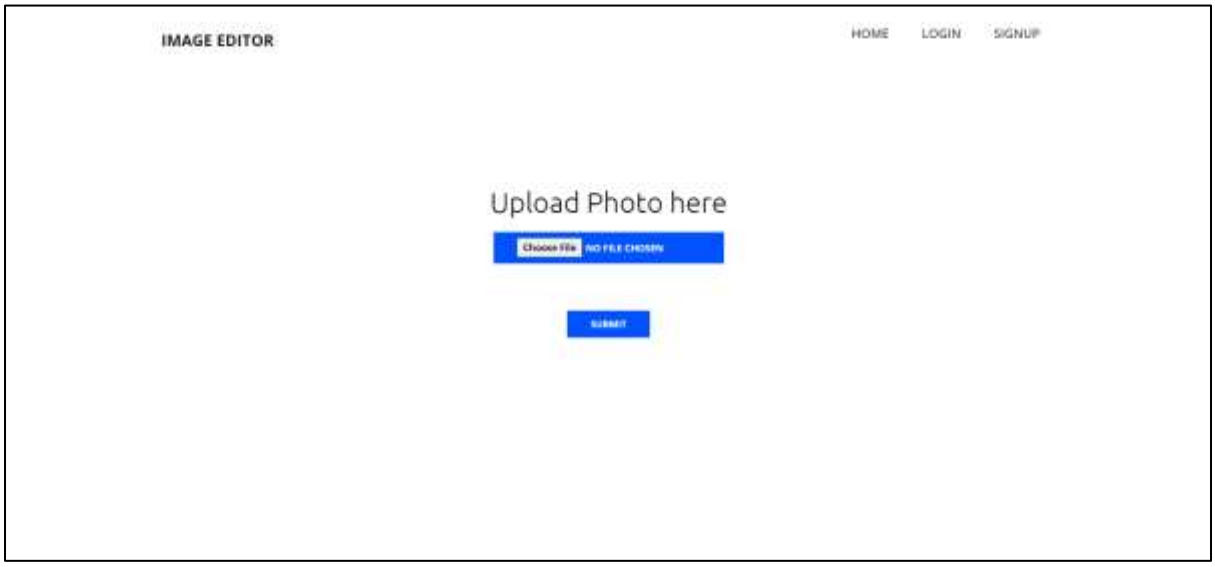


Figure 7. 3 Upload image

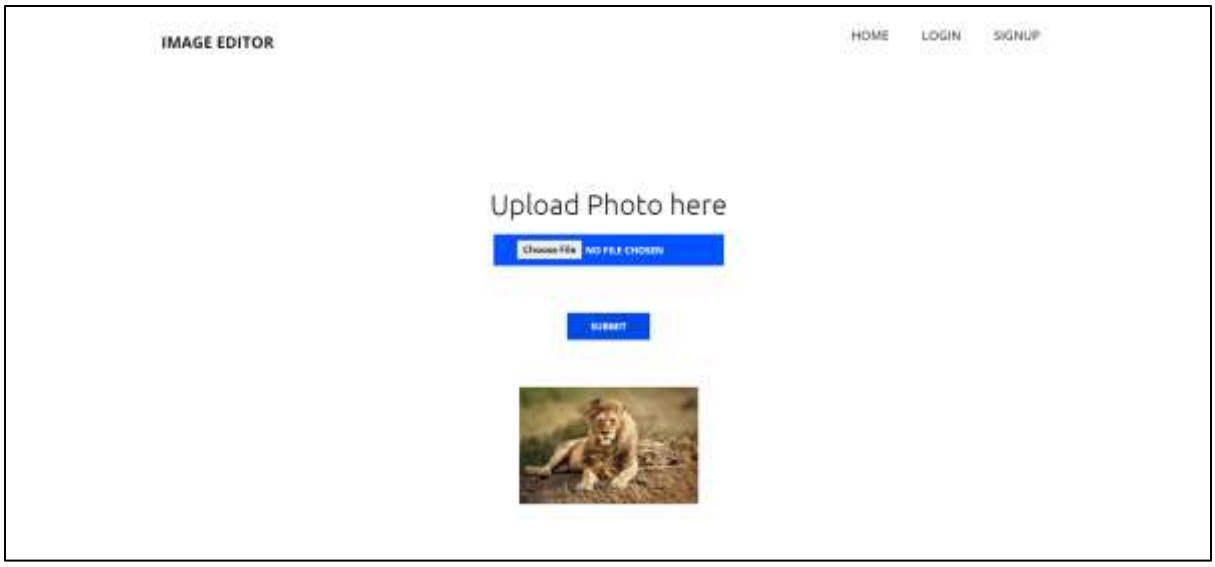
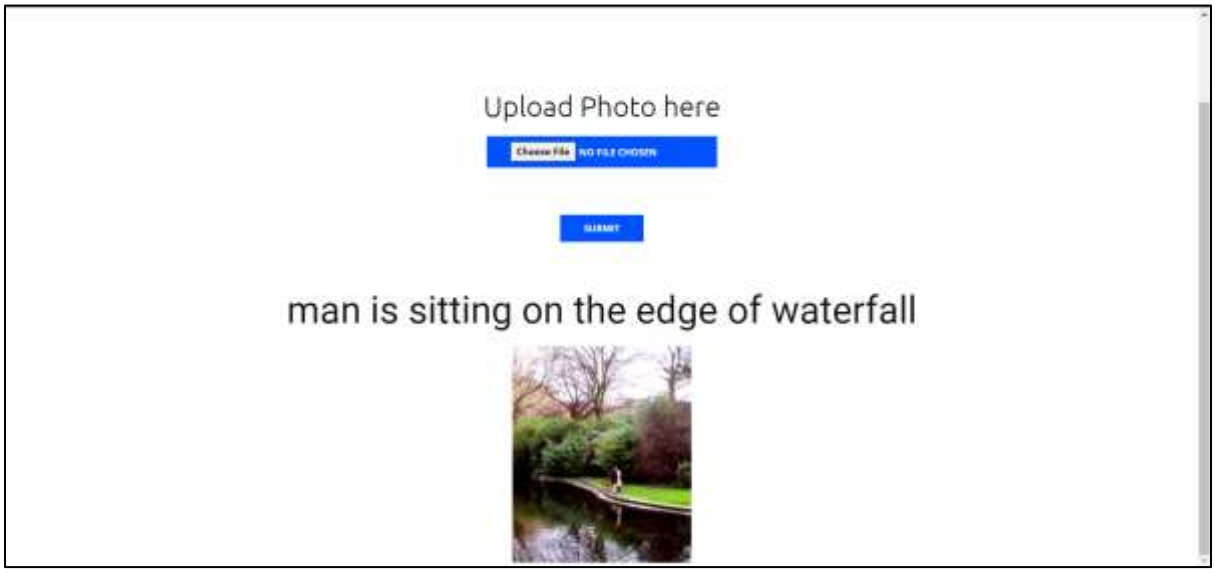


Figure 7. 4 Output image



8. Software Testing

Software testing is a process of running with intent of finding errors in software. Software testing assures the quality of software and represents final review of other phases of software like specification, design, code generation etc

8.1 Testing Plan

A Test Plan is a detailed document that describes the test strategy, objectives, schedule, estimation, deliverables, and resources required to perform testing for a software product. Test Plan helps us determine the effort needed to validate the quality of the application under test. The test plan serves as a blueprint to conduct software testing activities as a defined process, which is minutely monitored and controlled by the test manager. This system follows a coding and testing strategy for every small change for those particular changes and after completion of coding the system gets tested for the last time

8.2 Testing Method Testing Method

In testing methods, we are going with unit testing in which system is tested unified. That means each and every component added to the system is going to be tested whenever it will be added to the system for the first time before final release.

8.3 Testing Cases

Here is the list of test cases which is covered by unified testing models.

8.3.1 User

Figure 8.3. 1 Test cases

Purpose	**Result**
Verify if a user will be able to login with a valid username and valid password.	Pass

Verify if a user cannot login with a valid username and an invalid password	Pass
Verify if a user can logout	Pass
Verify the messages for invalid login	Pass
Verify if the data in password field is either visible as asterisk or bullet signs	Pass
Verify if error page is shown	Fail
Verify if home pages shown after successful login	Pass
Verify if user can click on all the “try this” button	Pass
Verify if user validated by an email	Fail
Verify if user can upload image	Pass
Verify if user upload image with wrong format get error message	Fail
Verify if user able to download image	Pass
Verify if user able to get desire output	Pass
Verify if user use Image colorization	Pass
Verify if user use Image Enhancement	Fail
Verify if user use Image Auto image background removing	Fail
Verify if user use Image Captioning	Pass

9. Conclusion & Future work

9.1 Conclusion

Image captioning and Colorization has many advantages in almost every complex area of Artificial Intelligence. The main use case of our model is to help visually impaired to understand the environment and made them easy to act according to the environment. As, this is a complex. task to do, with the help of pre trained models and powerful deep learning frameworks like TensorFlow and Keras, we made it possible. This is completely a Deep Learning project, which makes use of multiple Neural Networks like Convolutional Neural Network and Long Short-Term Memory to detect objects and captioning the images. To deploy our model as a web application, we have used Django, which is a powerful Python's web framework.

9.2 Future Work

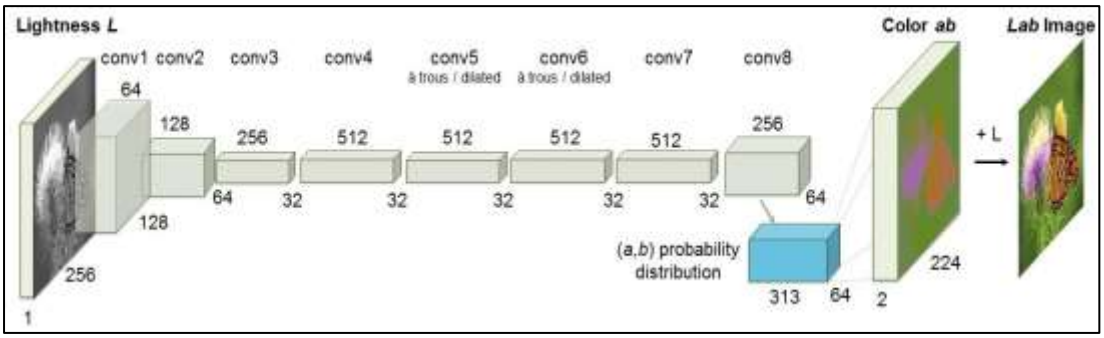
In future we'll add multiple future like object detection, image enhancement auto smile maker on people faces. Also, user can store their data and download image , see their history. Also, we try to build a mobile app same as our webapp so people can download on their phones. We try to build offline mobile application. this web app is for limited user we'll try to solve this problem also.

10. Detail About Models

10.1 Image colorization

10.1.1 Architecture

Figure 10.1 Image colorization model



10.1.2 Convolutional Neural Network

A CNN is made up of multiple layers of neurons, each of which is a nonlinear operation on a linear transformation of the preceding layer's outputs. The layers mainly include convolutional layers and pooling layers. The convolutional layers have weights that need to be trained, while the pooling layers transform the activation using a fixed function.

10.1.3 Conv Layer

A convolutional layer is the main building block of a CNN. It contains a set of filters (or kernels), parameters of which are to be learned throughout the training. The size of the filters is usually smaller than the actual image. Each filter convolves with the image and creates an activation map.

10.1.4 LAB color space

In LAB color space L stands for Lightness, A stands for red/green value, B stands for Blue/yellow value. The LAB color space is particularly useful for boosting colors and definition in images due to the way it handles colors when compared to RGB and CMYK. Rather than describing how colors should appear on a screen or in print, LAB is designed to approximate human vision. It's comprised of three axes: L represents

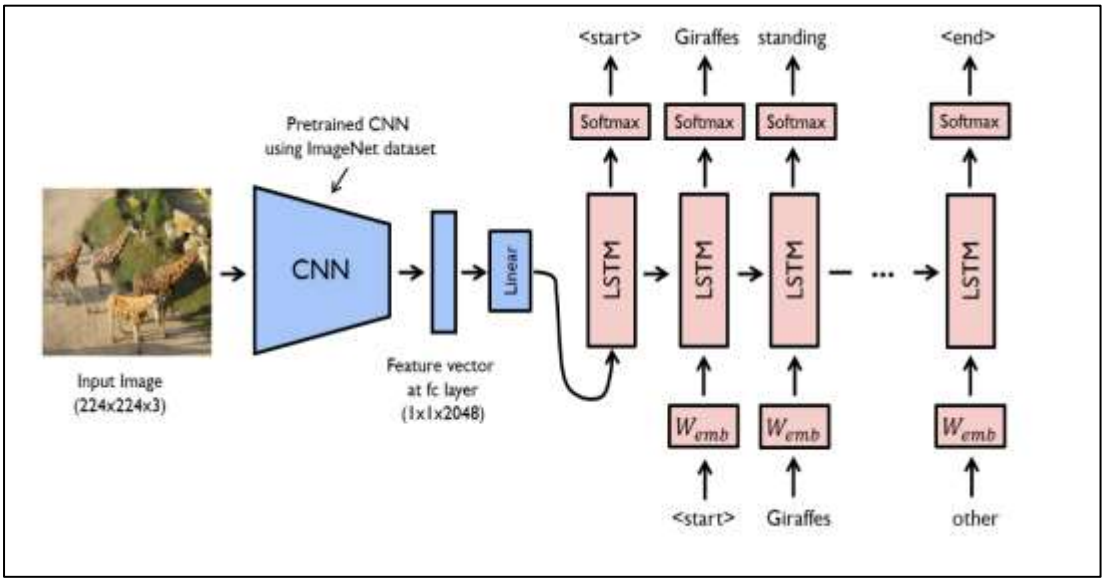
darkness to lightness, with values ranging from 0 to 100; a represents greenness to redness with values of -128 to +127; and b represents blueness to yellowness also with values from -128 to +127.

10.2 Image Captioning

So, to make our image caption generator model, we will be merging these architectures. It is also called a CNN-RNN model. CNN is used for extracting features from the image. We will use the pre-trained model Xception. LSTM will use the information from CNN to help generate a description of the image.

10.2.1 Architecture

Figure 10. 2 Image captioning model



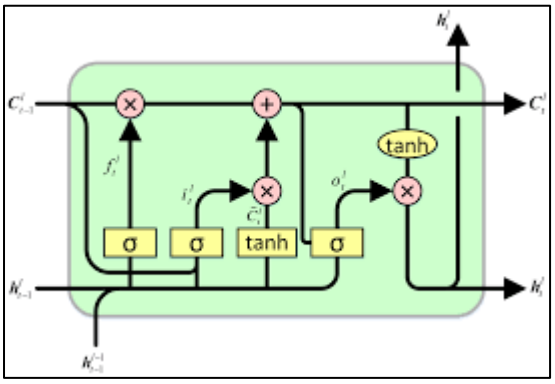
10.2.2 Convolutional Neural Network

A CNN is made up of multiple layers of neurons, each of which is a nonlinear operation on a linear transformation of the preceding layer's outputs. The layers mainly include convolutional layers and pooling layers. The convolutional layers have weights that need to be trained, while the pooling layers transform the activation using a fixed function.

10.2.3LSTM

LSTM stands for Long short-term memory; they are a type of RNN (recurrent neural network) which is well suited for sequence prediction problems. Based on the previous text, we can predict what the next word will be. It has proven itself effective from the traditional RNN by overcoming the limitations of RNN which had short term memory. LSTM can carry out relevant information throughout the processing of inputs and with a forget gate, it discards non-relevant information.

Figure 10. 3 LSTM cell



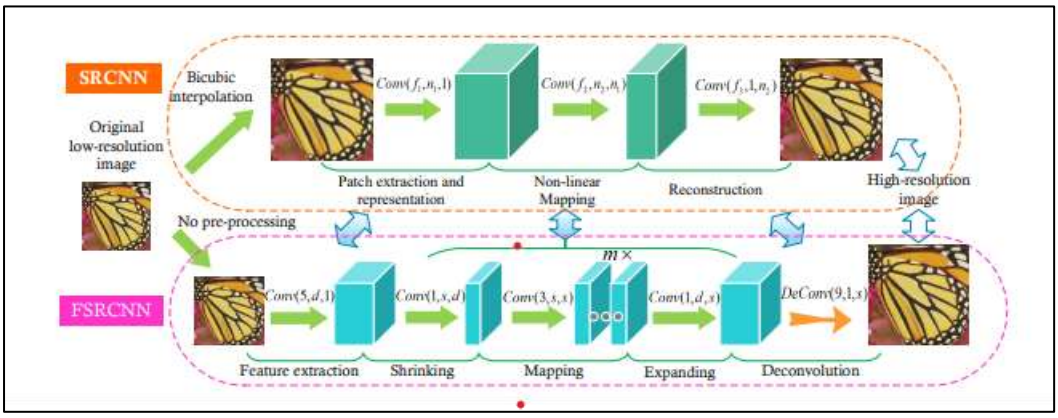
10.2.4SoftMax Activation Function

The SoftMax function is used as the activation function in the output layer of neural network models that predict a multinomial probability distribution. That is, SoftMax is used as the activation function for multi-class classification problems where class membership is required on more than two class labels.

10.3 Image Enhancement

10.3.1 Architecture

Figure 10.4 Image enhancement model



10.3.2 Convolutional Neural Network

A CNN is made up of multiple layers of neurons, each of which is a nonlinear operation on a linear transformation of the preceding layer's outputs. The layers mainly include convolutional layers and pooling layers. The convolutional layers have weights that need to be trained, while the pooling layers transform the activation using a fixed function.

10.3.3 FSRCNN

Fast Super-Resolution Convolutional Neural Network (FSRCNN) is an imaging technique that upscales the resolution quality of a digital image. It is based on a shallow network design that is faster and clearer than its predecessors in reproducing the image

SRCNN uses the following steps:

- Bicubic Interpolation to up sample the pixels to desired resolution.
- Convolutions are performed that improves the quality of the image. In this case, we have 9x9, 1x1 and 5x5 convolutions.

FSCRNN involves the following steps:

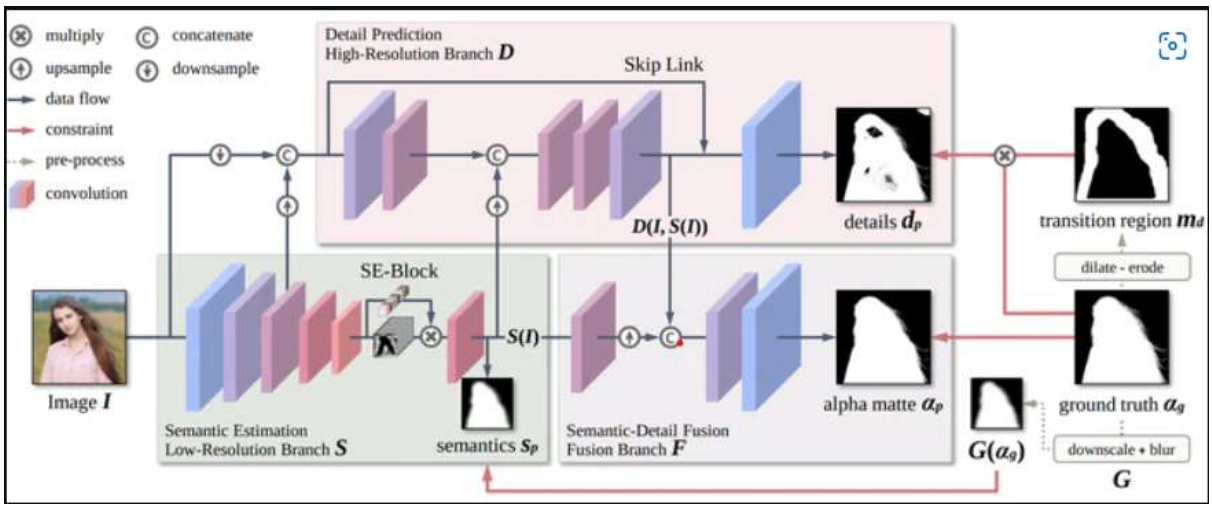
- Feature extraction — Replaces bicubic interpolation with 5x5 convolutions.

- Feature extraction — Replaces bicubic interpolation with 5x5 convolutions.
- Non-Linear Mapping — Multiple layers are applied 3x3.
- Expanding — The feature map is now increased by 1x1 convolutions.
- Deconvolution — High resolution image is reconstructed using 9x9 filter.

10.4 Image auto background removing

10.4.1 Architecture

Figure 10. 5 Image auto background removing (MODnet)



On its backbone, MODNet is built on the idea that a trimap-free matting as a segmentation step plus a trimap-based matting step can achieve better performance. MODNet extends this idea by further dividing the trimap-free image matting into semantic estimation, detail prediction, and semantic-detail fusion. The semantic estimation outputs a coarse foreground mask, whereas the detail prediction outputs a fine foreground mask. The semantic-detail fusion blends the features from both the first two layers.

MODNet presents a simple, fast, and effective architecture to avoid using green screens in real-time portrait matting. By only looking at RGB images, the model enables the prediction of alpha matte under varying

scenarios. Although having many upsides, the downside of the MODNet architecture is that it cannot handle strange costumes and strong motion blurs, which are not covered by the training set the model hasn't been trained on.

10.4.2 Semantic Estimation

The first step of semantic estimation is to locate the human/person in the image. This difference from the original method is that the extraction of the high-level features is only done by an encoder I-e the low-level branch. This turns the estimation more efficient since it is no longer done by a separate model that contains the decoder, in addition to the encoder.

10.4.3 Detail Prediction

The region around the foreground portrait is processed with a high-resolution branch which takes the image, the low-level branch output, and the low-level features from the low-level branch as inputs. The purpose of reusing the low-level features is to reduce the computational overheads of detail prediction.

10.4.4 Semantic Detail-fusion

The fusion branch of MODNet is convolutional neural network architecture. To learn about convolutional neural networks, visit a comprehensive guide to convolutional neural networks. The values from the previous two branches are concatenated together to predict the final alpha matte.

11. Reference

1. Hwang J, Zhou Y. Image colorization with deep convolutional neural networks. InStanford University, Tech. Rep. 2016.

2. Anwar, Saeed, Muhammad Tahir, Chongyi Li, Ajmal Mian, Fahad Shahbaz Khan, and Abdul Wahab Muzaffar. "Image colorization: A survey and dataset." *arXiv preprint arXiv:2008.10774* (2020).

3. Baldassarre, F., Morín, D.G. and Rodés-Guirao, L., 2017. Deep koalarization: Image colorization using cnns and inception-resnet-v2. *arXiv preprint arXiv:1712.03400*.

4. Wang, Zhihao, Jian Chen, and Steven CH Hoi. "Deep learning for image super-resolution: A survey." *IEEE transactions on pattern analysis and machine intelligence* 43.10 (2020): 3365-3387.

5. Dong, C., Loy, C. C., He, K., & Tang, X. (2015). Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, *38*(2), 295-307.

6. Minaee, Shervin, et al. "Image segmentation using deep learning: A survey." *IEEE transactions on pattern analysis and machine intelligence* (2021).

7. Hossain, M.Z., Sohel, F., Shiratuddin, M.F. and Laga, H., 2019. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CsUR)*, *51*(6), pp.1-36.

8. Wang, C., Yang, H., Bartz, C. and Meinel, C., 2016, October. Image captioning with deep bidirectional LSTMs. In *Proceedings of the 24th ACM international conference on Multimedia* (pp. 988-997).

9. Colorizing Old B&W Photos and Videos With the Help of AI | by Marcelo Rovai | Towards Data Science

10. Black and white image colorization with OpenCV and Deep Learning - PyImageSearch

11. Python based Project - Learn to Build Image Caption Generator with CNN & LSTM - DataFlair (data-flair.training)

12. Deep Learning based Super Resolution with OpenCV | by Xavier Weber | Towards Data Science

13. OpenCV Super Resolution with Deep Learning - PyImageSearch

14. Single image super-resolution with deep neural networks - Martin Krasser's Blog (krasserm.github.io)

15. Remove the background from images using AI and Python (livecodestream.dev)