**A Minor Project Report On**

**IMAGE TO CARICATURE**

In partial fulfilment of requirements for the degree of

**Bachelor of Technology (B. Tech.) Computer Science and Engineering**

**Submitted by:**

Manvi Vrati (190136)

Pranjal Tulsiyan (190119)

Saloni Chawla (190078)

**Under the Guidance of-**

Dr. Praneet Saurabh

*Department of Computer Science and Engineering*

**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**Mody University and Science and Technology Lakshmangarh, Distt. Sikar-332311**

15th November 2021

**ACKNOWLEDGEMENT**

I take this opportunity to express our profound gratitude and deep regards to the people who have been instrumental in the successful guidance for this project. I would like to thank my project guide **Dr Praneet Saurabh** for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessings, help, guidance and undivided attention given by him time to time shall carry us a long way in the journey of life on which we are about to embark.

I am much obliged to **Dr Vinod Maan** (HOD) and **Dr Rajeev Mathur** (Dean-SET) who has provided me with the best facilities and atmosphere for the project completion and presentation. I am always grateful towards my parents for supporting me in tough times. I would also like to thank all the team members involved in the development of this project as without their support, I would not be able to grasp everything so fast. Together we have completed the project and made it successful. Last but not least, we thank God Almighty for his choicest blessings showed on me in completing this!

**Manvi Vrati [190136]**

**Pranjal Tulsiyan [190119]**

**Saloni Chawla [190078]**

**CERTIFICATE**

This is to certify that the minor project report entitled “**Image to Caricature**”, submitted by Ms. **Manvi Vrati**, Ms. **Pranjal Tulsiyan**, and Ms. **Saloni**, as a partial fulfillment for the requirement of B. Tech. [Computer Science and Engineering], 5th Semester examination, of the School of Engineering and Technology, Mody University of Science and Technology, Lakshmangarh, for the academic session 2021-2022 is an original project work carried out under the supervision and guidance of **Dr Praneet Saurabh** has undergone the requisite duration as prescribed by the institution for the project work.

**PROJECT GUIDE: HEAD OF DEPARTMENT**

**Signature: Signature:**

**Name: Name:**

**Date: Date:**

**EXAMINER-I: EXAMINER-II**

**Signature: Signature:**

**Name: Name:**

**Date: Date:**

**ABSTRACT**

Python is a General-Purpose programming language. It is an open-source scripting language that has become the most popular introductory teaching language at top U.S. universities. It reduces manual labour by automating tasks. Python programs are shorter than equivalent programs written in any other languages as Java, because it has built-in high-level data types and dynamic typing. Renown professors believe that many universities are using Python as an introductory programming language due to its large set of highly useful libraries built over the years. They agree that it has made people feel more comfortable about exposing programming to a much broader audience.

One of the best applications of Python language is ML, i.e., machine learning. In this project, we were asked to experiment with a real-world problem, and to explore how machine learning algorithms can be used. We were expected to gain experience by using machine learning libraries.

Caricature is a comical representation of one’s image which emphasis on picturing the facial features boldly. In this minor project, we aim to transform images into cartoon. We will be using Python language on PyCharm. Conversion will be done by first uploading the image in the interface made by using Tkinter and then just by clicking on the Cartoonify Button. Hence, we get the caricature of the image ready to be saved in our devices. We will be using various ML libraries like OpenCV, Numpy, matplotlib, etc. We will put our knowledge of Machine Learning together to create this innovative program.

This application can be useful for people who are interested in comic and webtoon writing to create realistic characters. This can also be used by programmers/ hackers or other people who put their Bitmoji’s as their profile pictures as they don’t wish to reveal their faces for privacy purposes but they wish their DP to be unique. Graphic designers also use caricatures to escape their boring routine. It turns a simple hobby to lifestyle. Now a days caricatures are in trend. We can see several Snapchat and Facebook filters using similar techniques as well. People try using these techniques to see how different their friends or other famous people look like, once gone through the effect. Not only it is a great source of pleasure and entertainment for several people but also an interesting way to reach a milestone in learning technology for some. So, this application can be put to a good use!

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Topics** | **Page no.** |
|  | **Introduction** | 1-2 |
|  | Present System | 2 |
|  | Proposed system | 2 |
|  | **System Design** | 3-4 |
|  | System Flowchart | 3 |
|  | Progress Gantt Chart | 3-4 |
|  | **Hardware and Software details** | 5-8 |
|  | **Implementation Work Details** | 9 |
|  | Real-life applications | 9 |
|  | Data implementation and program execution | 9-12 |
|  | **Stimulation Code** | 13-17 |
|  | **Input/output Screens [Model Photographs]** | 18 |
|  | **System Testing** | 19 |
|  | **Individual contribution** | 20 |
|  | **Conclusion** | 21 |
|  | Limitations | 21 |
|  | Scope for future works | 21 |
|  | **Bibliography** | 22 |
|  | **Annexures** | 23 |
|  | Publication communicated | 23 |
|  | Plagiarism Report | 23 |

1. **INTRODUCTION**

Python is a General-Purpose programming language. Python programs are shorter than equivalent programs written in any other languages as Java, because it has built-in high-level data types and dynamic typing. It is an introductory programming language due to its large set of highly useful libraries built over the years. It makes people feel more comfortable about exposing programming to a much broader audience. One of the best applications of Python language is ML, i.e., machine learning. In this project, we were asked to experiment with a real-world problem, and to explore how machine learning algorithms can be used. We will be using various ML libraries like OpenCV, Numpy, matplotlib, etc. We will put our knowledge of Machine Learning together to create this innovative program “**Image to Caricature**”.

**Caricature** is a comical representation of one’s image which emphasis on picturing the facial features boldly. It can be viewed as a cartoon like image that appears two dimensional but realistic at the same time. Example of an image, converted into his caricature is as follows:



Figure 1.1. Original Image



Figure 1.2. Caricature of above image [Figure 1.1.]

* 1. **PRESENT SYSTEM**

Presently, there are several applications that people use in their mobile phones, which can provide the facility of making an image in its cartoon form. But we aim to develop a desktop application that can take input from the already present files in a system and produce caricature images in just one click and very less time. For example, Snapchat is a mobile application that can provide filters for converting our image into caricature while clicking the selfie, but that filter is actually designed in such a way that the 2-D features to be applied are already defined, which are same for everyone. But in our project, we will extract the image edges and then convert it into its caricature, so the features defined boldly will be different in every different image chosen.

* 1. **PROPOSED SYSTEM**

This project is designed for fun purpose. We can cartoonify image of people, animals, flowers or other objects, given that the image is already present in our device, on just one click. The application runs by extracting edges and smoothening the image to boldly express its unique features and creates a caricature which seems to be two dimensional but actually looks very real. This application uses machine learning libraries and creates a beautiful cartoon art of the image given in input. It is done by plotting points of images on matplotlib and also adding brighter colours to the final image.

1. **SYSTEM DESIGN**
   1. **SYSTEM FLOWCHART**

Figure 2.1. System Flow Chart

Figure 2.1. depicts the flow of data in the proposed system. How we have imported the modules n our code file, then started creating a window and used these libraries to cartoonify image.

* 1. **PROJECT TIMELINE**

Figure 2.2. depicts that this project was created in the time allotted to us by university. This image is the Gantt chart of our code implementation. Following are the heads:

* **Initialising the topic:** we came across various topic for creating minor project but since machine learning is a demand in the university, we opted for developing a unique project in it.
* **Research**: for making this concept work we did a lot of research work on different modules present in machine learning [Python] which we can use in our project. Many libraries were rejected out till the end.
* **Project design:** after gaining information about what work the libraries in machine learning may do, we created a design, a patter in which our project will work [say, the flow of steps]
* **Project synopsis:** since synopsis are to be submitted, we prepared it in a very short time and asked suggestions from our mentor.
* **Implementation phase I:** in this phase, we created the functions to use file from system and convert image to cartoon**.**
* **Implementation phase II:** in this phase, after our mid-term examinations, we worked on grey scaling, smoothening and other aspects of images to be used for making its cartoon. Some people started putting the code together and system testing while others were engaged in documentation work at this time.
* **Documentation:** this phase includes creating reports, extracting research papers etc. this ran parallel to phase 2 of implementation.

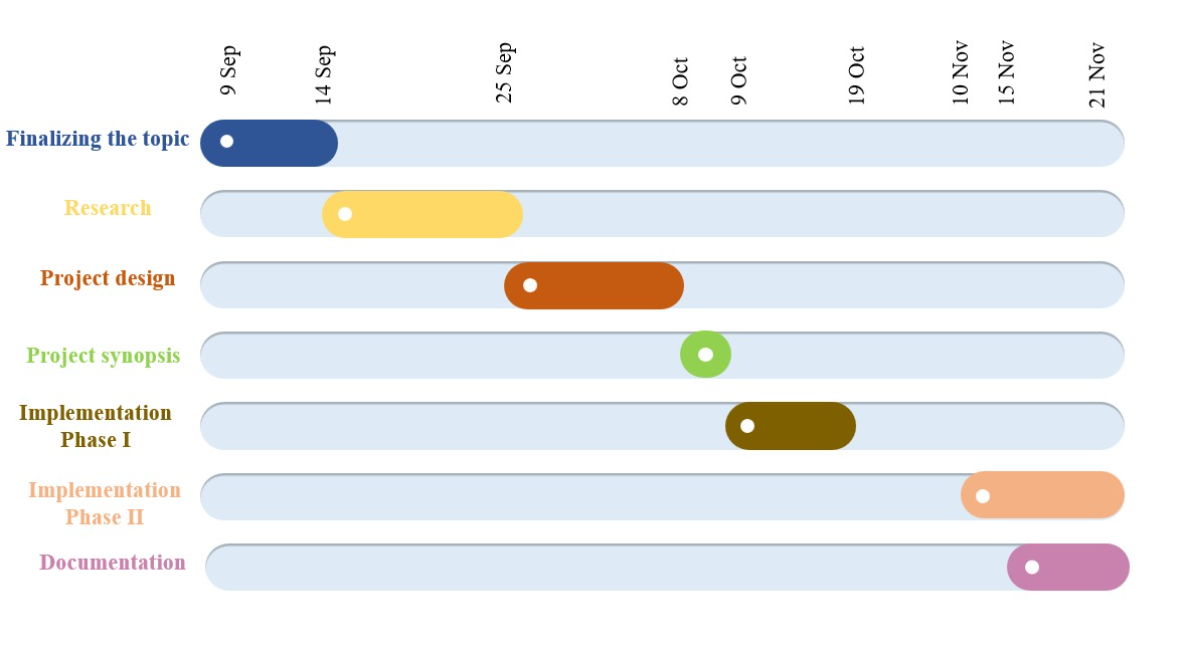


Figure 2.2. Gantt Chart of project timeline

1. **HARDWARE AND SOFTWARE REQUIREMENTS**
   1. **CPU AND MEMORY ANALYSIS**

For This project a 4 GB RAM is sufficient. The CPU and Memory usage is shown in the following figures. The following figure depicts the usage of CPU and Memory before and after the app is started for the project to run successfully.

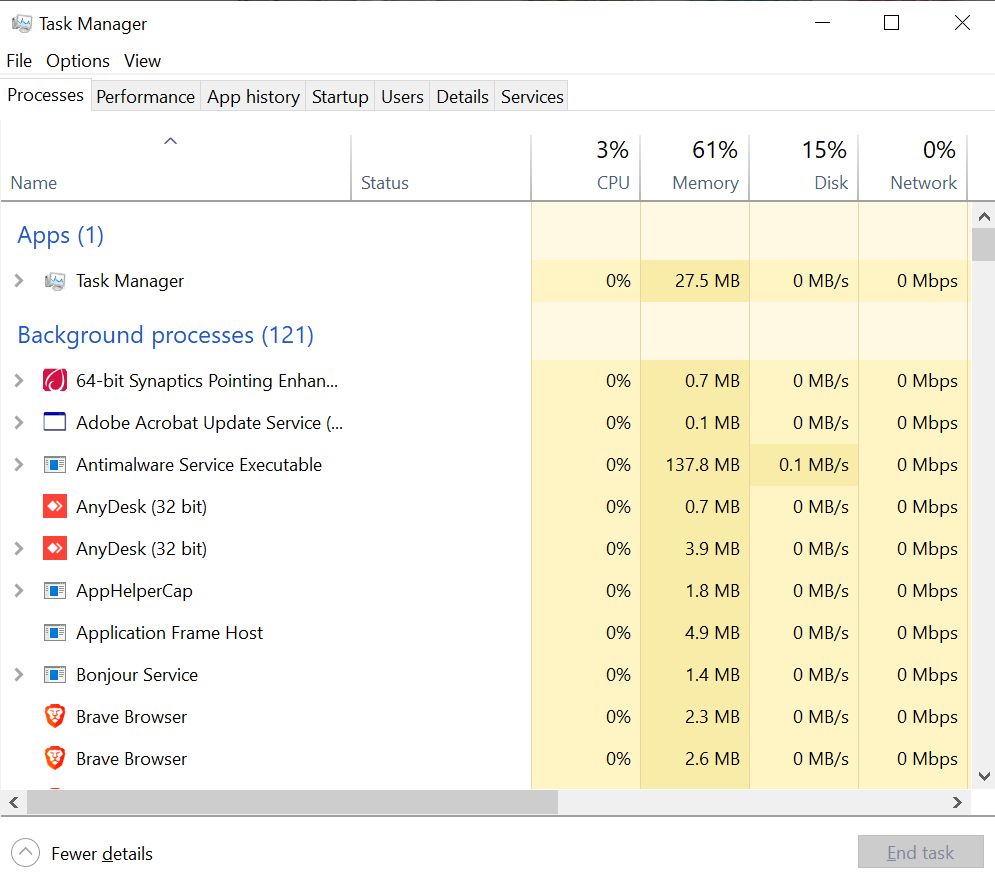
****

Figure 3.1. CPU and Memory usage before starting project

Figure 3.1. shows the CPU and memory usage before any applications are started. Initially it shows that the CPU usage is 3% and Memory usage is 61%. Here 27.5 MB is occupied by the Task Manager and remaining memory by the background processes.

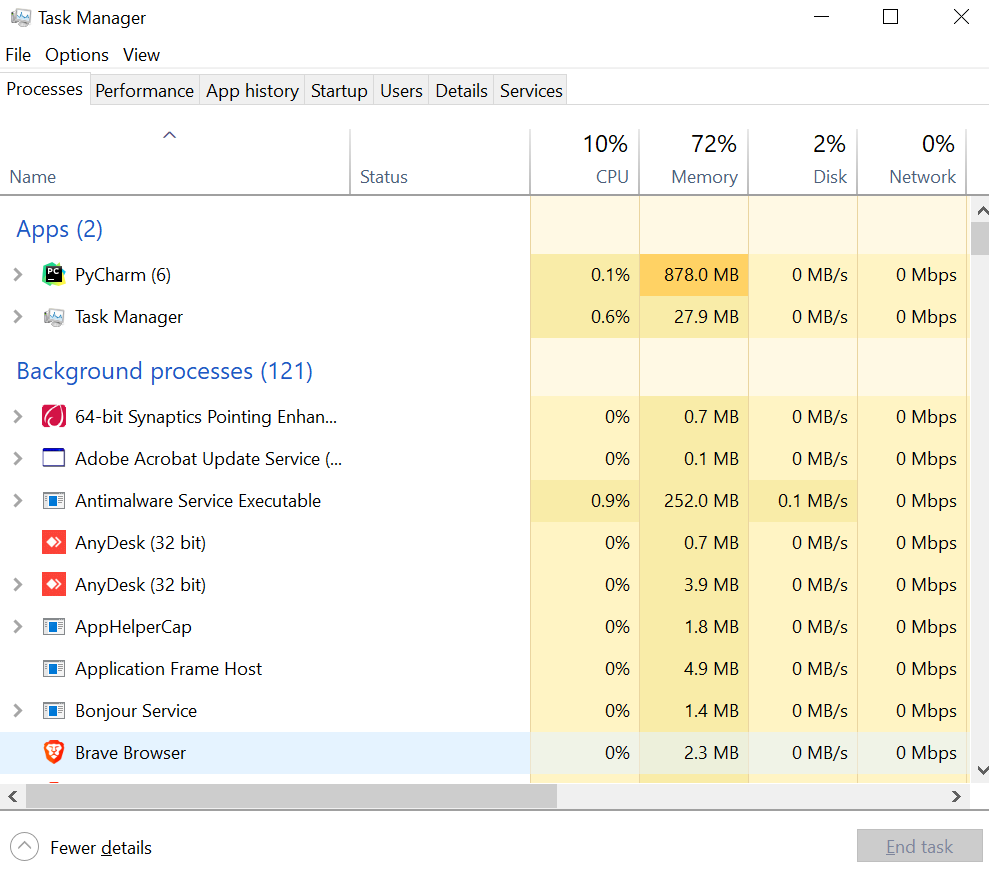
****

Figure 3.2. CPU and Memory usage after starting XAMPP

Figure 3.2. Shows the CPU and Memory usage after Pycharm is started for database connectivity. The CPU usage is 0.1% and the Memory needed for Pycharm is 878.0 MB.

The above analysis shows that a RAM with 4 GB is sufficient for the Hardware of the project. And as it is not a web-based application so, we need not have an Internet connectivity to run the project.

* 1. **PYCHARM**

PyCharm is a dedicated Python Integrated Development Environment (IDE) that provides a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, Web, and Data Science development. It is a cross-platform IDE that provides consistent experience on the Windows, macOS, and Linux operating systems. It supports only Python Language. We can even change themes according to our preferences. It auto completes maximum functions for smooth working environment.

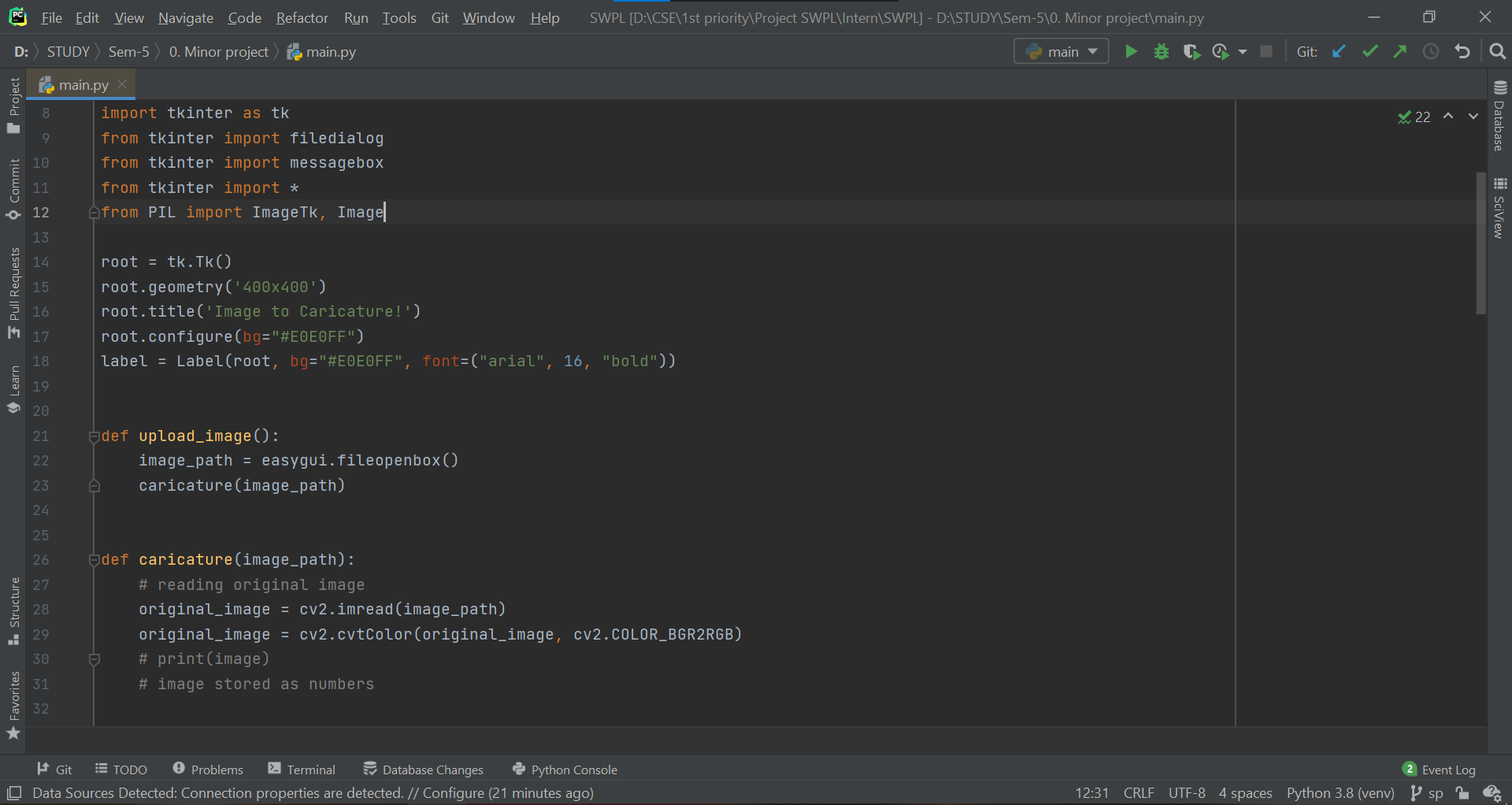


Figure 3.3. PyCharm Environment

* 1. **PYTHON PROGRAMMING**

Python is the fastest growing language in terms of developers, libraries and applications it can be used for. It could be used in Machine Learning, Artificial Intelligence, Web Development; anything we can think of, can be developed with python. Python is a general-purpose programming language which is high-level, easy to learn and dynamically initialized.

In python we used indentations instead of braces to display a block of code. Python is an easy and open-source language, it is free for everyone to use. Python can be used to make web application, mobile application, server-side scripting, AI, ML etc. There is a huge community of people who came together to develop libraries or modules that can be used to optimize solutions. It has huge Libraries for user support.

Here, as we are using PyCharm and we have included Numpy module, we are viewing results plotted into numpy calculator interface as shown below.

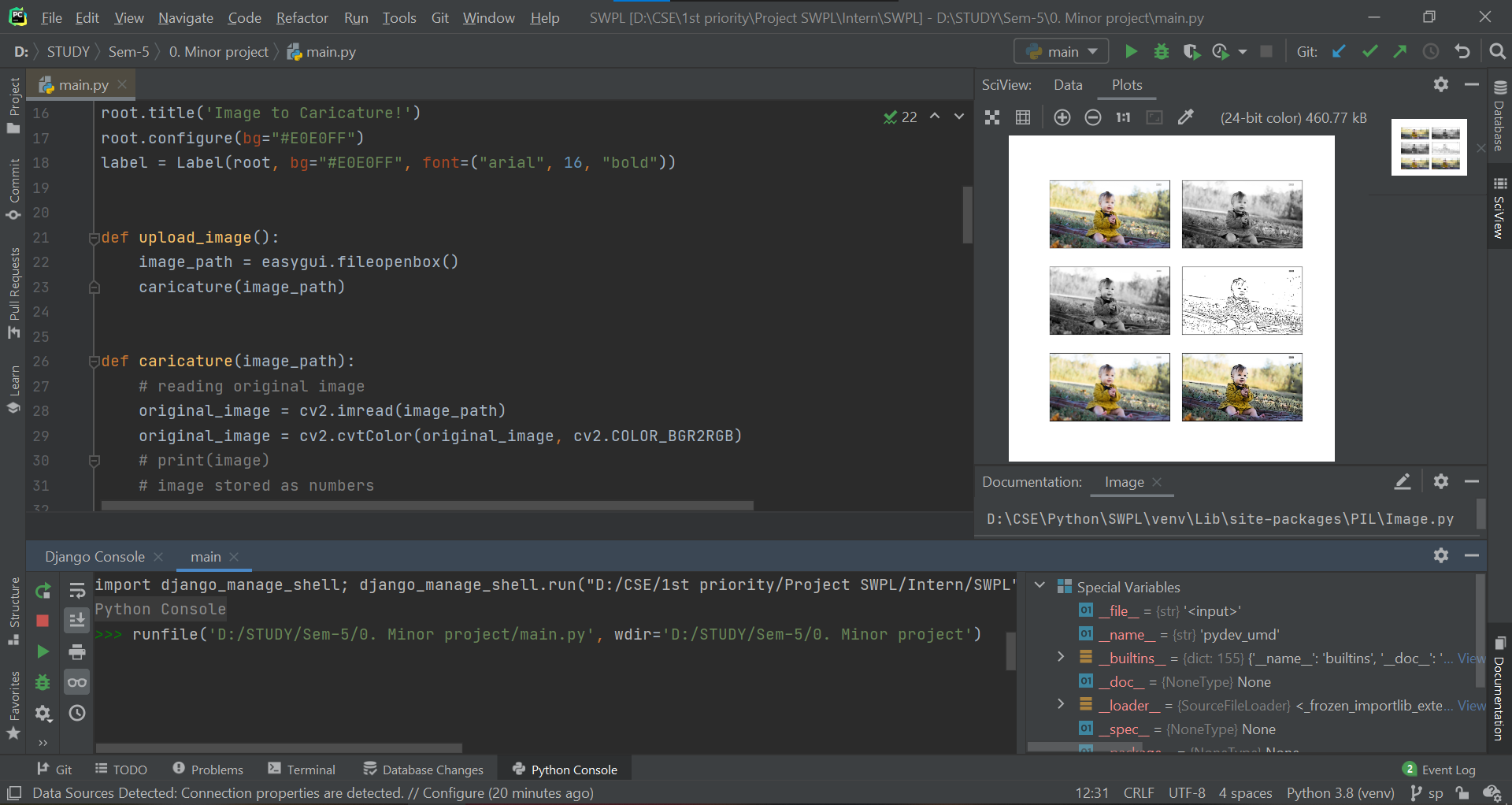


Figure 3.4. Python Programming, Project Code Snippet

1. **IMPLEMENTATION WORK DETAILS**
   1. **REAL LIFE APPLICATIONS**

* This application can be useful for people who are interested in comic and webtoon writing to create realistic characters.
* This can also be used by programmers/ hackers or other people who put their Bitmoji’s as their profile pictures as they don’t wish to reveal their faces for privacy purposes but they wish their DP to be unique.
* Graphic designers also use caricatures to escape their boring routine. It turns a simple hobby to lifestyle.
* Now a days caricatures are in trend. We can see several Snapchat and Facebook filters using similar techniques as well. People try using these techniques to see how different their friends or other famous people look like, once gone through the effect.
* Not only it is a great source of pleasure and entertainment for several people but also an interesting way to reach a milestone in learning technology for some. So, this application can be put to a good use!
  1. **IMPLEMENTATION AND PROGRAM EXECUTION**

Following are the libraries and modules used in our project for creating a caricature image of a given image:

* + 1. **Module cv2**

OpenCV is a python open-source library used for image processing, ML and computer vision.

cv2.imread(path, flag) is used to load image from specified file where the parameter path is the path of image to be read and flag is the way to read the image. Flags are also of 3 types:

* cv2.imread\_COLOR
* cv2.imread\_UNCHANGED
* cv2.imread\_GRAYSCALE
  + 1. **Module easygui**

Easygui is a module for very simple and easy GUI programming. It provides a user-friendly interface for simple GUI interaction. In this programmer does not need to know anything about framer, tkinter, widgets or lambda. EasyGUI is different from other GUI generators. It is not event-driven. All GUI interactions are invoked by simple function calls. This is used here to open a file box. It allows us to select any file from our system.

* + 1. **Module numpy**

NumPy is a fundamental package and a python library that provides a multidimensional array object, derived objects like marked array and matrices. It is also used for operations on arrays, mathematical logical shape, shape manipulation, selecting, sorting. It provides N- dimensional array , sophisticated functions, tools for integrating C/C++, useful linear algebra.

* + 1. **Module imageio**

Imageio is a library used in python programming with easy interfacing to read and write a wide range of image data like animated images, volumetric data and scientific formats.

* For reading an image imageio.imread() method is used.
* For reading gif file imageio.get\_reader() method is used.
* For creating image file imageio.imwrite() method is used.
  + 1. **Module sys**

In python programming sys module is used to manipulate different parts of python runtime environment by various functions and variables. With this we can perform operations on interpreter as it gives access to those variables and functions that interact strongly with the interpreter. Also, it provides variables for controlling input/output in a better way. This is done using 3 variables:

* stdin- used to get input from cmd directly and also automatically adds '\n' in the end of every sentence
* stdout- used to display output in any form directly in screen console
* stderr- sys.stderr is written whenever an exception occurs in Python
  + 1. **Module matplotlib**

matplotlib.pyplot is a plotting library used to interface to matplotlib. It provides a way of plotting just like MATLAB. Also, it opens figures on the screen and acts as figure GUI manager.

pyplot is majorly used for interactive plots like Line plot, Histogram, Polar, 3D plot etc. and simple cases of programmatic plot generation. Matplotlib is generally used for user interface toolkits like Tkinter, awxPython etc.

* + 1. **Module os**

As the name suggests OS module in python is used for interacting with operating system through functions. OS is one of the python's standard utility modules which provides a portable way of using OS dependent functionality.

Its functions are:

* os.getcwd- confirms returns the presently working directory.
* os.mkdir- to create a new directory.
* os.chfir- to change current working directory to a newly created one.
* os.rmdir()- to remove any specified directory
* os.listdir()- to list all files and directories in the specified directory
  + 1. **Module tkinter**

Tkinter is standard graphical user interface (GUI) library for python. It provides various controls that are used to build GUI applications. Some of them are:

* Button- In python application button is used to add different kinds of button.
* Canvas- This widget is used to draw the canvas on window
* Entry- This is used to display single line text field.
* Label- It is a text field used to display information about other widgets.
* Menu- used to add menu items

Tkinter modules we used here:

* Messagebox widgets: used to display messageboxes for applications in python.
* Filedialogbox: It is intended to provide unique dialogues to be used for dealing files.

There are several predefined methods present in tkinter that makes window making easy for any application we develop.

* + 1. **Module PIL**

PIL stands for python imaging library which is a free and open-source library in Python intended to support for opening, controlling and saving various image file formats. It provides powerful image processing capability and efficient internal representation. It is designed for fast access to data stored in a basic pixel format. It provides a strong foundation for any image processing tool.

1. **STIMULATION CODE**
   1. **IMPORTING REQUIRED MODULES**

Following modules are imported in the code:

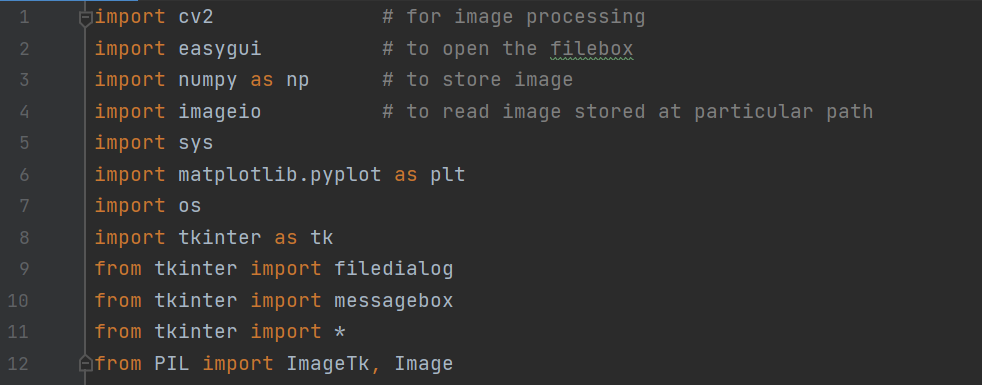


Figure 4.1. Imported modules

* 1. **BUILDING A FILE BOX TO CHOOSE A PARTICULAR FILE**

In this step we have written the function ‘upload\_image’ to open the file box, i.e., the pop-up box to choose the file from the device, which opens every time we run the code. fileopenbox() is the method in easyGUI module which returns the path of the chosen file as a string.

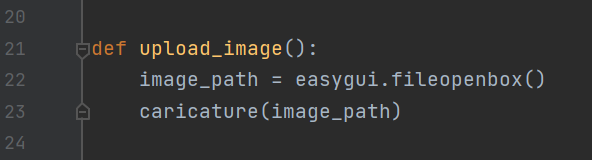


Figure 5.2. Function upload\_image()

* 1. **READING AND RESIZING IMAGE**

In **caricature** function 1st we are reading and resizing the image. **imread** is a method in cv2 which is used to store images in the form of numbers. The image is read as a numpy array, in which cell values depict R, G, and B values of a pixel. **cvtColor(image, flag)** is a method in cv2 which is used to transform an image into the colour-space mentioned as ‘flag’. We will resize the image after each transformation to display all the images on a similar scale at last.

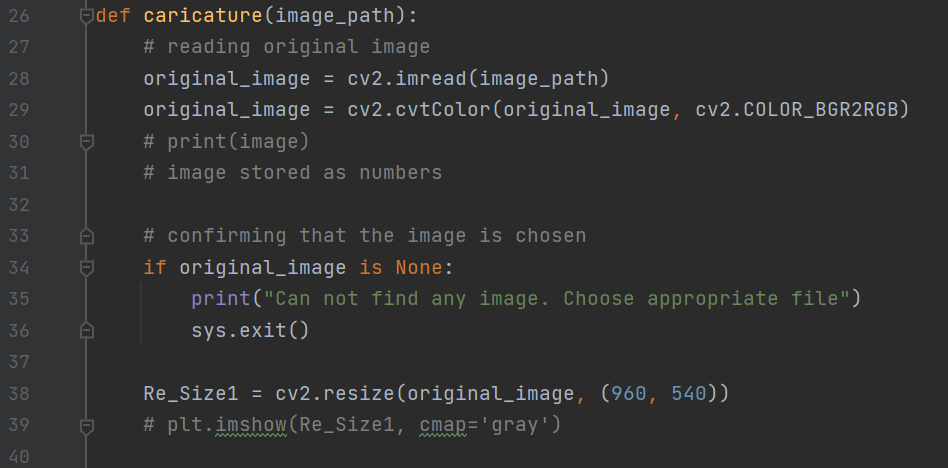


Figure 5.3. Function caricature()

* 1. **TRANSFORMING AN IMAGE TO GRAYSCALE AND SMOOTHENING IT**

To convert an image to a cartoon, several transformations are done. The image is converted to a Grayscale image. Then, this Grayscale image is smoothened. After this we try to extract the edges in the image. Finally, we form a color image and mask it with edges. This creates a beautiful cartoon image with edges and lightened color of the original image.

Here, to convert the image in grayscale, we have used the BGR2GRAY flag. This returns the image in grayscale. To smoothen an image, we simply apply a blur effect. This is done using **medianBlur()** function. The center pixel is assigned a mean value of all the pixels. This in turn, creates a blur effect.

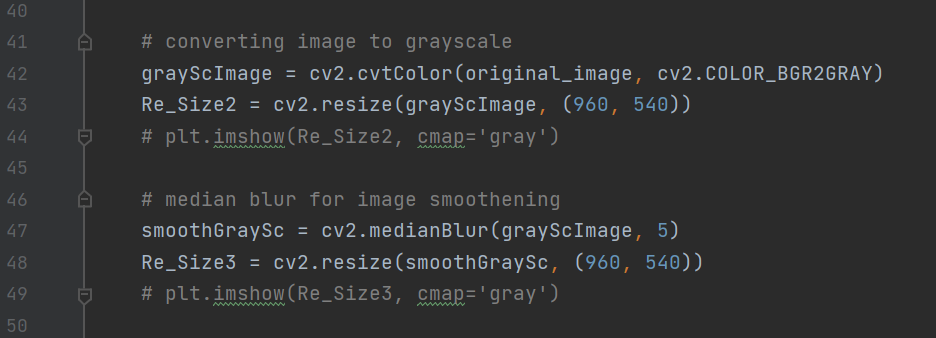


Figure 5.4. Converting image to grayscale and smoothening

* 1. **RETREIVING THE EDGES OF AN IMAGE AND TO PREPARE A MASK IMAGE**

Caricature has 2 specialities: its highlighted edges and smooth colours. So, we need to bring out the edges and make them bold so that they highlight even more. For this we take mean of neighbourhood pixel values minus the constant. The type of threshold applied is THRESH\_BINARY. For masking the image, the bilateral filter is used that removes the noise from image. This is smoothening of an image. Similar to beautify image option we have in mobile cameras.

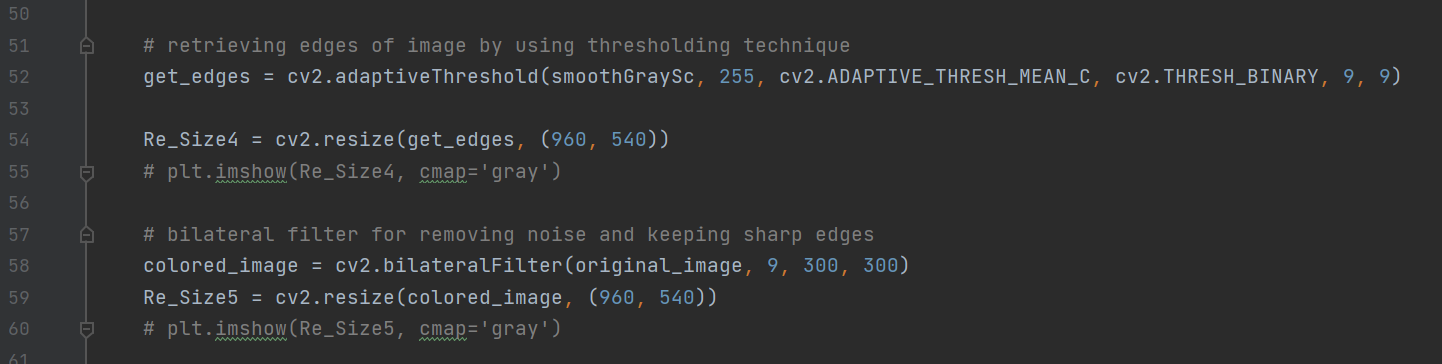


Figure 5.5. Edges retreive and masking of image

* 1. **GIVING A CARTOON EFFECT**

We need to combine the beautified image and the bold retrieved edges together to obtain a caricature image. Bitwise AND operation is performed on two images and this is called Masking. Plotting the transitions together using matplotlib, we first create axes and display images in each block.

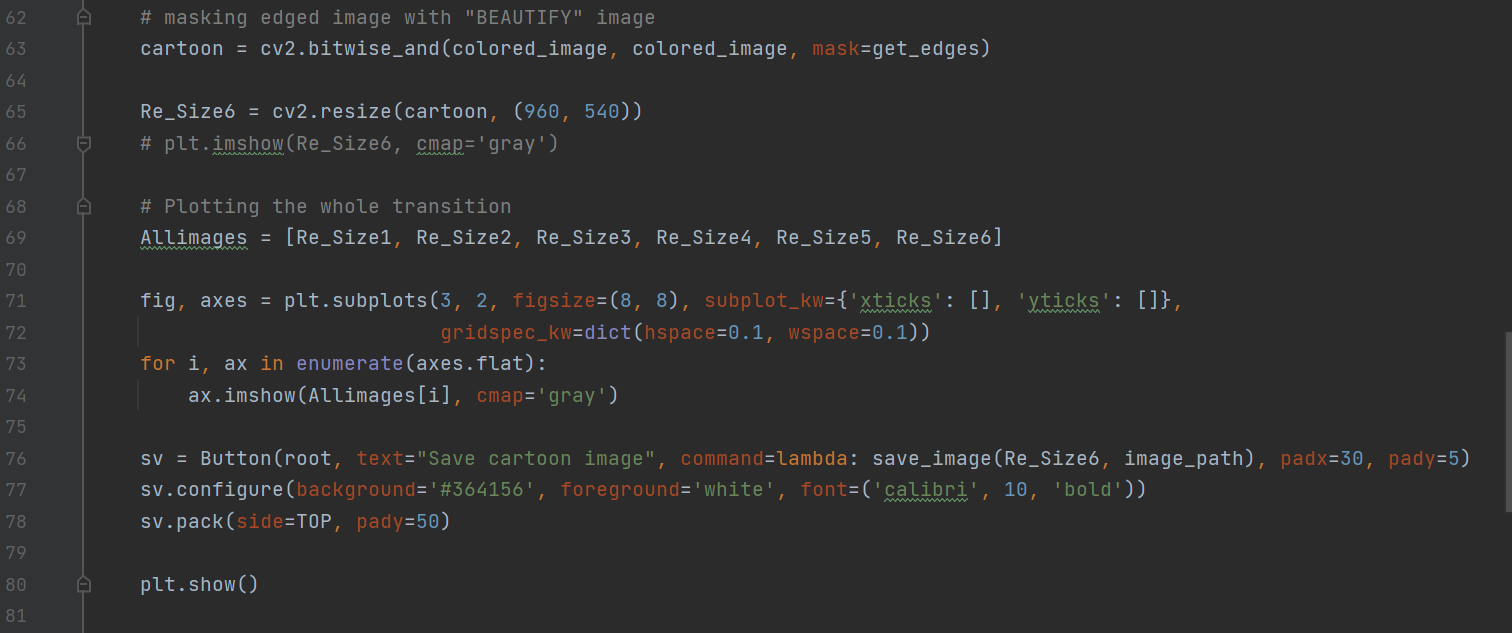


Figure 5.6. Caricature effect and plotting points

* 1. **FUNCTIONALITY OF SAVE BUTTON**

To save the image we have created, we take the old image path to save image in the same folder and have predefined the name: “cartoonified\_image”. For taking the image path using the library OS.

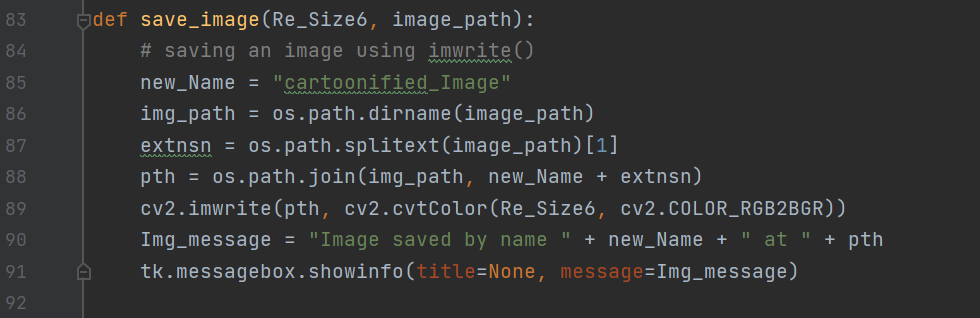


Figure 5.7. Saving image

* 1. **MAKING THE MAIN WINDOW BY TKINTER**

Making the window using tkinter is upon the user, it can be done according to our preferences. At the end of the file we need to write **root.mainloop()** so that the tkinter window appears as soon as the main program is executed.

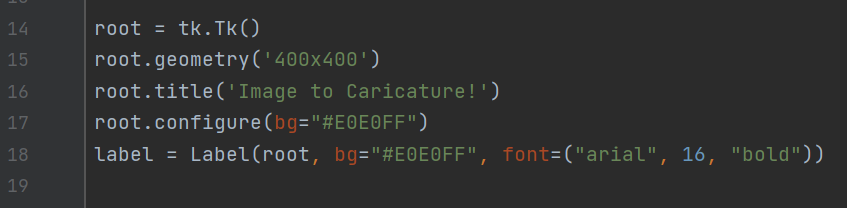


Figure 5.8. Tkinter window making

* 1. **MAKING THE CARTOONIFY AND SAVE BUTTON IN THE MAIN WINDOW**

Following are the snippets to make the save button and cartoonify button.

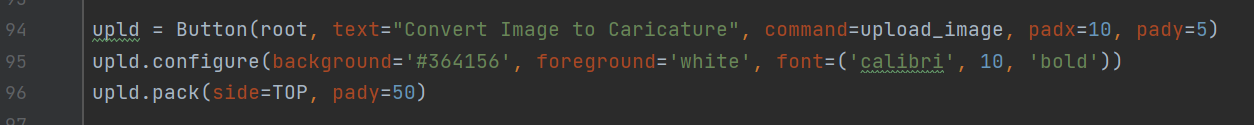


Figure 5.9. Upload button

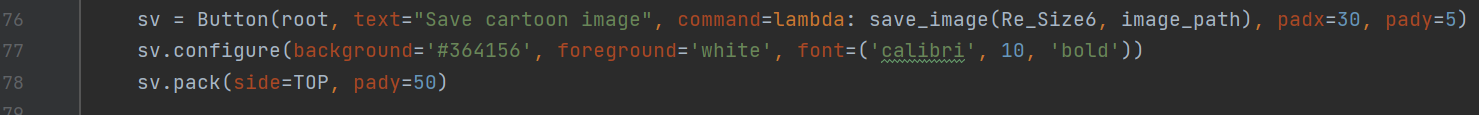


Figure 5.10. Save button

1. **INPUT/ OUTPUT SCREENS**

Some of the model images that are processed into caricatures using our application are present below:

|  |  |  |
| --- | --- | --- |
|  | 🡪 |  |
|  | 🡪 |  |
|  | 🡪 |  |
|  | 🡪 |  |

1. **SYSTEM TESTING**

In this project, we have done the manual testing to test the application. In **Manual testing**, test-cases are executed normally by all the team members, i.e., without any support from tools or scripts. We have to ensure that the application is working properly based on the multiple model images. We need to evaluate numerous components like dimension, quality of image, and extent to which the image is cartoonified, by uploading different images in the tkinter window. There can be a possibility that not all errors are fixed, as the process is manual and not machine operated. For these circumstances, we can work on suggestions provided by users and implement fixing those bugs in the next version of the application.

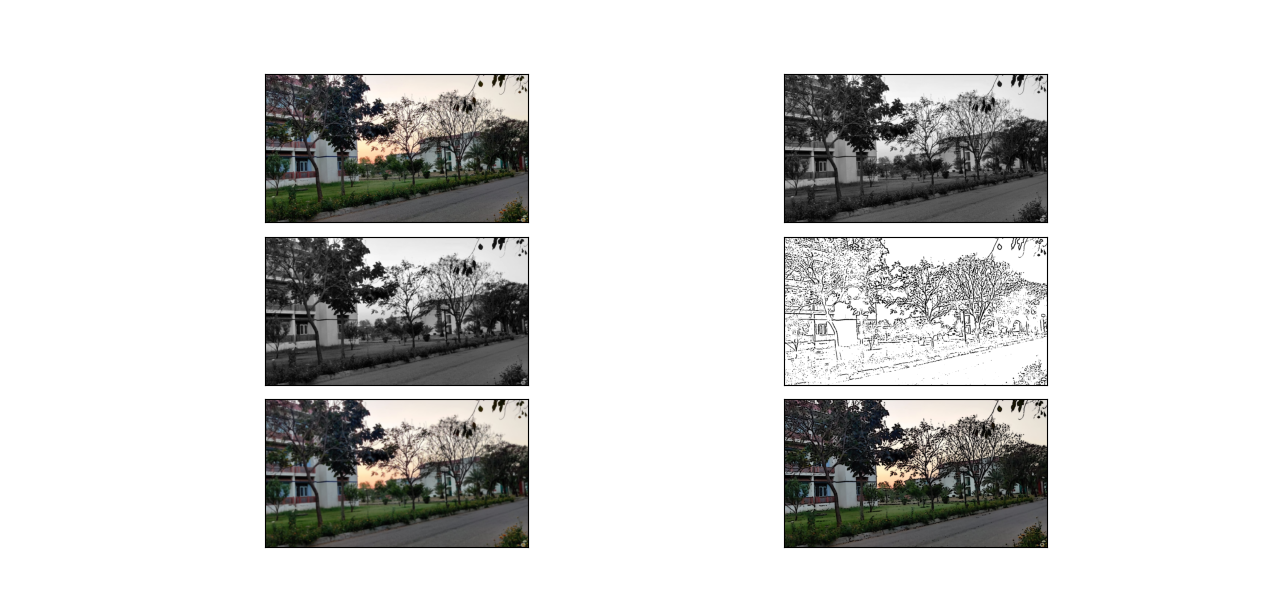


Figure 7.1. System testing done manually at all phases of image conversion

1. **INDIVIDUAL CONTRIBUTION**

Since this project was constructed during the times of COVID pandemic, all the members in the team had to work from home. Hence the project was completed when we were all connected through google meet. We read about ML together, explored libraries and functions and completed the code together. Even the documentation was done in parts by everyone. Therefore, the project was completed with equal efforts made by all the team members.

* Manvi Vrati [190136]:

I have particularly contributed in the part of code where we are starting by uploading the image and extracting the edges. I have also done the report formatting and explaining the code in detail in our report. In the research paper, I have contributed by searching existing models and their limitations so that we can see how our model works better.

* Pranjal Tulsiyan [190119]:

I have particularly contributed in the part of code where we are converting image to grayscale and masking it. I have also done the part in report where we are showing the output through various images and identifying the scope. In research paper, I have contributed by precisely plotting the output image on graphs and deriving the results.

* Saloni Chawla [190078]:

I have particularly contributed in the part of code where we are returning the cartoonified image by compiling the mask effect with highlighted edges and saving the image into our devices. I have also done the part in report where we have explained about all the libraries we have used in our project in detail. In research paper, I have contributed in searching matter and putting forward our algorithm in clear steps.

Since we have done manual testing, everyone performed hit and trial to obtain limitations and outcomes of our application. Everyone equally contributed in searching matter for documentation.

Due to the teamwork, together we have completed the project successfully.

1. **CONCLUSION**

In this project with the help of OpenCV, we have transformed images (snapshots) to the finest caricature of them. Also, with the help of CV2 which is known as Computer vision, we have transformed images in such a way that every image will have its own unique features on being cartoonified as, the features used to cartoonify them are not predefined. They are retrieved from the qualities of the original image itself.

* 1. **LIMITATIONS**

In this project, we can see that dimensions are fixed, i.e., when we take an image of any size [dimension wise] we get the caricature image in the dimensions that we have already given in the project. The dimensions are static so, some times we get an image that is slightly compressed. Also, image quality decreases a bit after the task is performed. Moreover, caricature image can only be clearly identified, if the original image quality is high, otherwise sometimes the difference may not be visible to bare eyes. This process is a general one and will not give the best result for different types of images.

* 1. **SCOPE FOR FUTURE WORK**

We can find a proper way to save images by user defined names. We can work on the point that caricature can be obtained from any size of image, considering all dimensions possible. The library cv2 can be explored and the quality of produced image can be improved. If the program is further converted into application, we can link this to SQLite or other databases to store all information.

1. **BIBLIOGRAPHY**
2. Isabelle Belato, *Caricature*, Sept 13, 2021. Accessed on: Oct 8, 2021. [Online].

Available: <https://en.wikipedia.org/wiki/Caricature>

1. “*Cartooning an Image using OpenCV-Python*,” Dec 22, 2020. Accessed on: Sept 25, 2021. [Online].

Available: <https://www.geeksforgeeks.org/cartooning-an-image-using-opencv-python/>

1. Tazki Anida Asrul, *Turn Photos into Cartoons Using Python*, Jan 2, 2021. Accessed on: Sept 14, 2021. [Online].

Available: <https://towardsdatascience.com/turn-photos-into-cartoons-using-python-bb1a9f578a7e>

1. Nikita Jain, “*Understanding cartoon emotion using integrated deep neural network on large dataset*”, April 21, 2021. Accessed on: Nov 19, 2021. [Online].

Available: <https://link.springer.com/article/10.1007/s00521-021-06003-9>

1. Aljaž Jeromel & Borut Žalik, “*An efficient lossy cartoon image*”, August 29, 2019. Accessed on: Nov 24, 2021. [Online].

Available: <https://link.springer.com/article/10.1007/s11042-019-08126-7>

1. Wenqing Chu, “*Learning to caricate via semantic shape transform*”, July 9, 2021. Accessed on: Nov 20, 2021. [Online].

Available: <https://link.springer.com/article/10.1007/s11263-021-01489-1>

1. Vaishali Sundarshan, Amritest Singh, “*Cartoonifying an image using OpenCV and Python”*, January 24, 2021, Accessed on: Nov 19, 2021. [Online].

Available:<https://www.academia.edu/44964756/Cartooning_an_Image_Using_Opencv_and_Python>

1. Mohommad Razi Ul Haq, “*What is OpenCV in Python*”, March 05, 2021. Accessed on: November 20, 2021. [Online].

Available:[What is OpenCV in Python? (educative.io)](https://www.educative.io/edpresso/what-is-opencv-in-python?utm_term=&utm_campaign=%5BTest%5D+Dynamic+Verticals&utm_source=adwords&utm_medium=ppc&hsa_acc=5451446008&hsa_cam=14045073269&hsa_grp=128822123241&hsa_ad=535845844735&hsa_src=g&hsa_tgt=dsa-1394252596758&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gclid=Cj0KCQiAy4eNBhCaARIsAFDVtI2F1xzLOEc2HCQ_IA7oWwlreyGx3uG9MNmc2mJRRTx64VkEZxW9-_YaAvcUEALw_wcB)

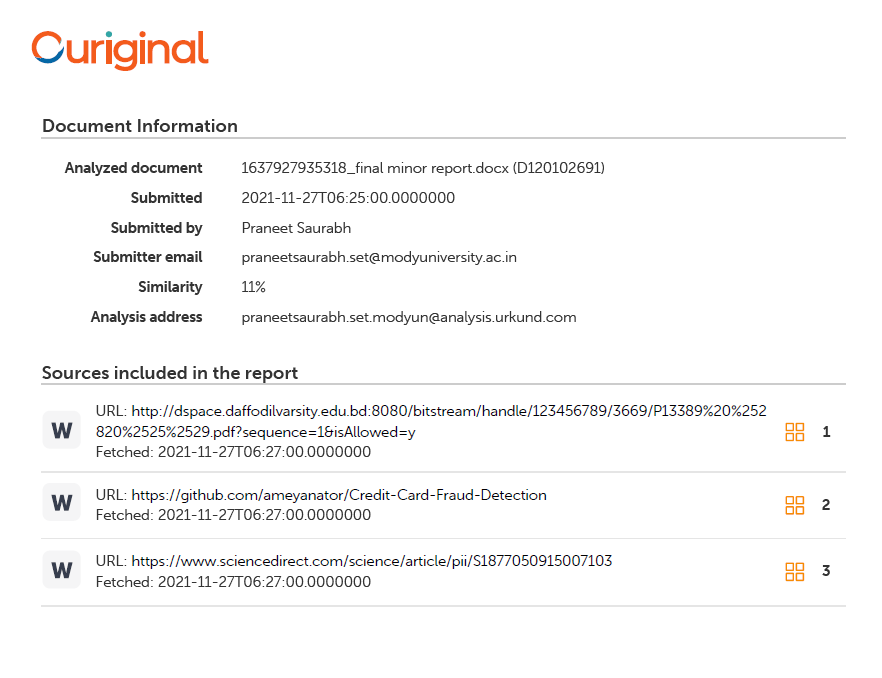
1. “*Pyplot in Matplotlib*”, Accessed on: November 20, 2021. [Online].

Available: [Pyplot in Matplotlib - GeeksforGeeks (ampproject.org)](https://www-geeksforgeeks-org.cdn.ampproject.org/v/s/www.geeksforgeeks.org/pyplot-in-matplotlib/amp/?amp_js_v=a6&amp_gsa=1&usqp=mq331AQKKAFQArABIIACAw%3D%3D#aoh=16380218424052&referrer=https%3A%2F%2Fwww.google.com&amp_tf=From%20%251%24s&ampshare=https%3A%2F%2Fwww.geeksforgeeks.org%2Fpyplot-in-matplotlib%2F)

1. Image references, Accessed on: November 15, 2021. [Online].

Available: [500+ Wild Animal Pictures [HD] | Download Free Images on Unsplash](https://unsplash.com/s/photos/wild-animal?modal=%7B%22tag%22%3A%22Filters%22%7D)

1. **ANNEXURES**
   1. **Research Paper** link [published]**:**
   2. **Plagiarism report:**

****