

|   |   |  |  |
|---|---|--|--|
| <b>Topic</b>  | <b>IMAGE PROCESSING</b>   |  |  |
| <b>Class Description</b>  | Students will learn how to implement the image processing activities like image modifications using python with the help of web based UI.   |  |  |
| <b>Class</b>  | <b>ADV-C266</b>   |  |  |
| <b>Class Time</b>   | <b>60 mins</b>  |  |  |
| <b>Goals</b><br> | <ol style="list-style-type: none"> <li>1. What is Image processing</li> <li>2. Image processing using python</li> <li>3. Create logic and UI for processing an image into a black and white and sketch image.</li> </ol>  |  |  |
| <b>Resources Required</b>   | <ul style="list-style-type: none"> <li>• Teacher Resources: <ul style="list-style-type: none"> <li>○ Use Gmail login credentials.</li> <li>○ Laptop with camera</li> <li>○ Earphone with mic</li> <li>○ Notepad and Pen</li> </ul> </li> <li>• Student Resources: <ul style="list-style-type: none"> <li>○ Use Gmail login credentials.</li> <li>○ Laptop with camera</li> <li>○ Earphone with mic (optional)</li> <li>○ Notepad and Pen</li> </ul> </li> </ul> |  |  |
| <b>Class Structure</b>  | <b>Warm Up</b><br><b>Teacher-Led Activity</b><br><b>Student-Led Activity</b><br><b>Wrap Up</b><br><b>Project Pointers and Cues</b>  |  | <b>5 Mins</b><br><b>15 Mins</b><br><b>30 Mins</b><br><b>5 Mins</b><br><b>5 Mins</b>                |
| <b>Class Steps</b>  | <b>Say</b><br>   |  | <b>Do</b><br> |

**Step 1:**  
**Warm up**  
**(5 mins)**

In the previous class 265, we learnt the concept of image processing code which is used to convert the Color image into Black and white and deploy the same image processing web app on heroku which will act as IAAS.

**Q** Why have we mentioned the method as POST?

**A** We have mentioned the method as POST because we have set the same method to our form.

**Q** Why have we used the **save()** function and which parameters have we passed to it?

**A** We have used **save()** function to upload the image in a folder.

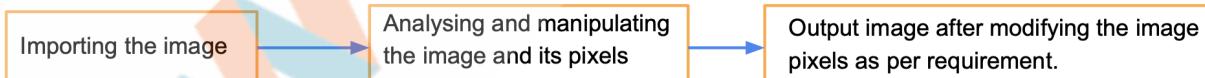
|   |  |   |
|---|--|---|
|   |  |   |
| <b>Teacher Initiates Screen Share</b>   |  |   |
| <p><b>Step 2:<br/>Teacher-Led<br/>Activity<br/>(15 mins)</b></p>   |  | <p><b>NOTE: Course outcomes are the activities which are giving the idea of concept to be implemented so precisely convey it to learners before the Teacher Activity delivery</b></p> <p><b>SAY: Access the data from the local device which will emulate as the data is taken from the cloud environment, so they can modify and restore it on the local device.</b></p> |
| <p>In the previous class we created an image processing application and hosted it on cloud. That can be used as Infrastructure as service where people will be using the Infrastructure for converting their image to balck and white image.</p> <p>In today's class also we will continue working on image processing in which we will convert the image into sketch</p> <p>Let' Start:</p> <p><b>Image Processing</b></p> <p>Image processing is a method to perform some operations on a given image, in order to enhance the image quality or to get the useful image pixel value as information from it.</p> |  |   |



Lets see the following steps required to make such conversion of images.

**Image processing basically includes the following three steps:**

1. Importing the image
2. Analysing and manipulating the image and its pixels.
3. Output image after modifying the image pixels as per requirement.



#### **1. Importing the image:**

- In this, we do import the required image on which we want to make any changes or want to convert it into some different type.
- Then getting the image pre-processed by converting the image into an array of pixels which will be further used to add the required feature of image conversion.

## 2. Analysing and manipulating the image and its pixels.

- Here, we do apply the function of image processing as per the requirement. For example, a function for converting colored images into black and white images.

## 3. Output image after modifying the image pixels as per requirement.

- Now as the image pixels are converted and set as per the requirement i.e from black and white to Color image. Another image processing tool helps to convert the converted pixels of image into a full dimensional image as output which will be a modified colored image as per the requirement.



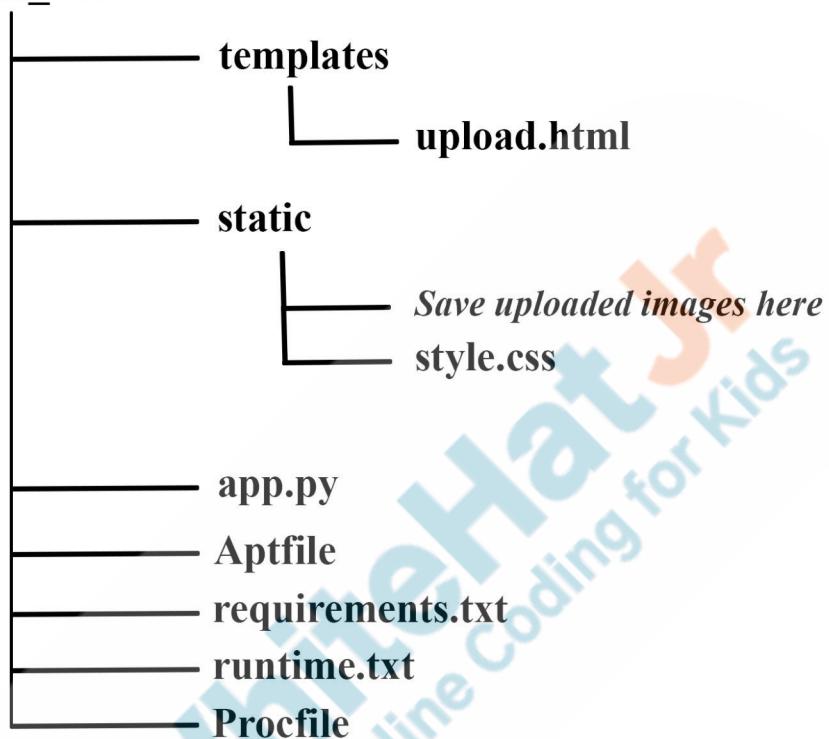
So here in this class we are going to write code and build a web app to upload images and convert it into Grayscale or Sketch image as per choosing the option from the drop down and display image on a localhost url in the web browser.

So let's start.

**NOTE - Download the complete code from [Teacher-Activity-1](#) and unzip the folder and move it to the Document\cloud folder. Then explain the folder structure :**

**File structure:**

## Class\_266



In this folder we have :



In which folder is the **HTML** i.e **upload.html** file is saved?



The **HTML** i.e **upload.html** file is saved in the **Templates folder** which will hold the UI of our application.



In which folder is the **style.css** file saved?

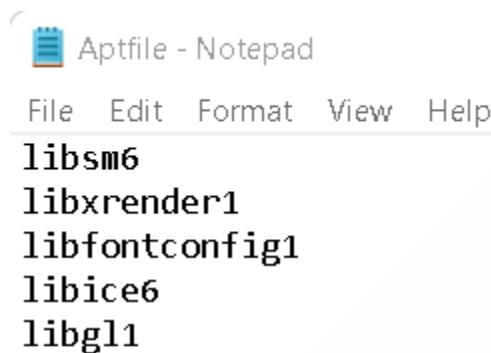


The **style.css** file is saved in the **static** folder.

- **app.py** file which will have the main logic of the web app

In addition to this we have:

- **Aptfile** : This file is required to build the connection with heroku and required building packages. As we are using cv2 library hence when we want to host it on heroku we need to mention the required packages for cv2 library works well with heroku properly. This file is already given to you predefined.



Aptfile - Notepad

File Edit Format View Help

```
libsdl2-2.0
libxrender1
libfontconfig1
libice6
libgl1
```

- **Procfile** - This file is used to deploy web apps on heroku. In this file we mention which file should run when a web app is deployed on heroku. This file tells heroku which file to execute in order to run the app on the heroku platform.



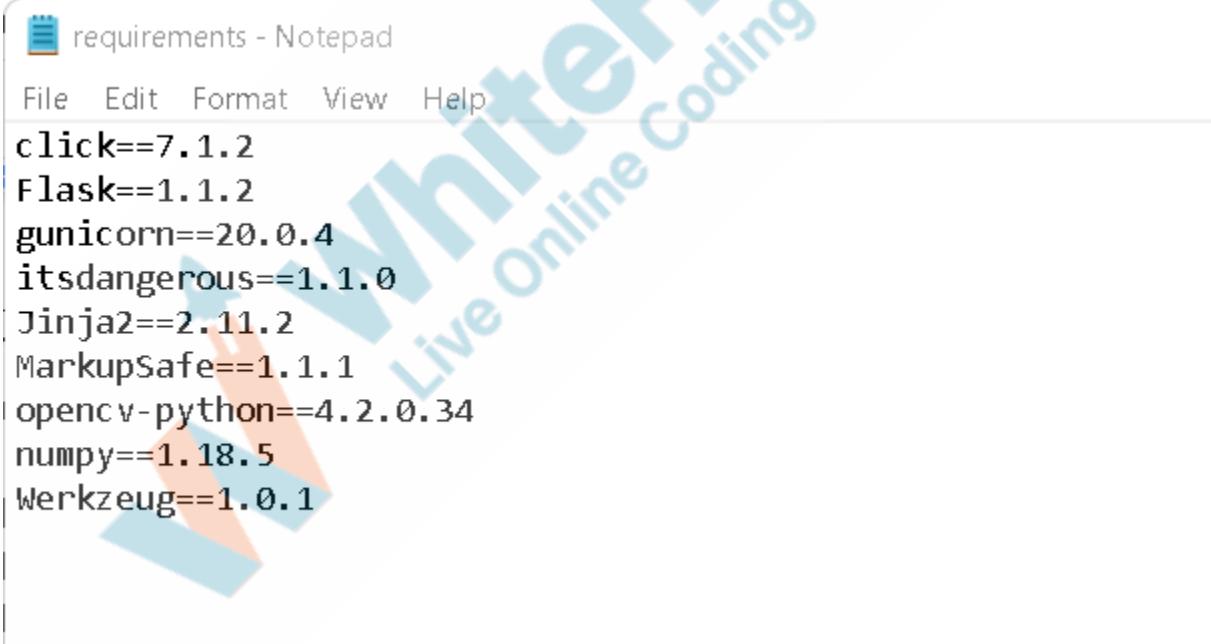
Procfile - Notepad

File Edit Format View Help

```
web: gunicorn app:app
```

Here :

- **web** defines a web application.
- **gunicorn** is a small python server which is required to deploy app on heroku
  - The **app** is nothing but our python file name where all our python code is stored. If we had some other file name then that name would be here.
- **Requirements.txt** - As name suggests in this file we mention the libraries which will be used for deployment of the webapp. So if you want to use a new library then you need to mention it in this file.
  - As we wish to deploy the web app on cloud we need to mention all the used libraries in the requirements.txt file.
  - So when you deploy the web app, the cloud platform will look into this **requirements.txt** file and install all the mentioned libraries as per the versions of each library.

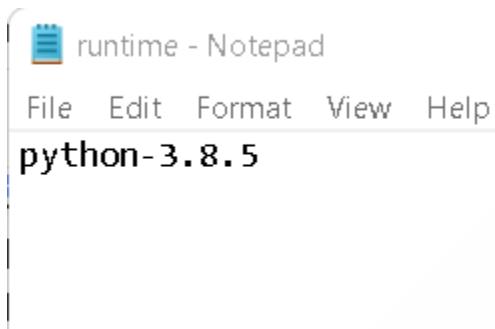


```
requirements - Notepad
File Edit Format View Help
click==7.1.2
Flask==1.1.2
gunicorn==20.0.4
itsdangerous==1.1.0
Jinja2==2.11.2
MarkupSafe==1.1.1
opencv-python==4.2.0.34
numpy==1.18.5
Werkzeug==1.0.1
```

- Here you can see we are using click, flask, gunicorn, itsdangerous, jinja2, markupsafe and werkzeug libraries.
- In our project we are using the above libraries. So you can say that requirements.txt file holds the libraries name and its version required to run the

program.

- **runtime.txt** : This file includes the version of python on which we want to run our web application, Like in this file we define the python version which we need to install on heroku and run our flask application as shown below.



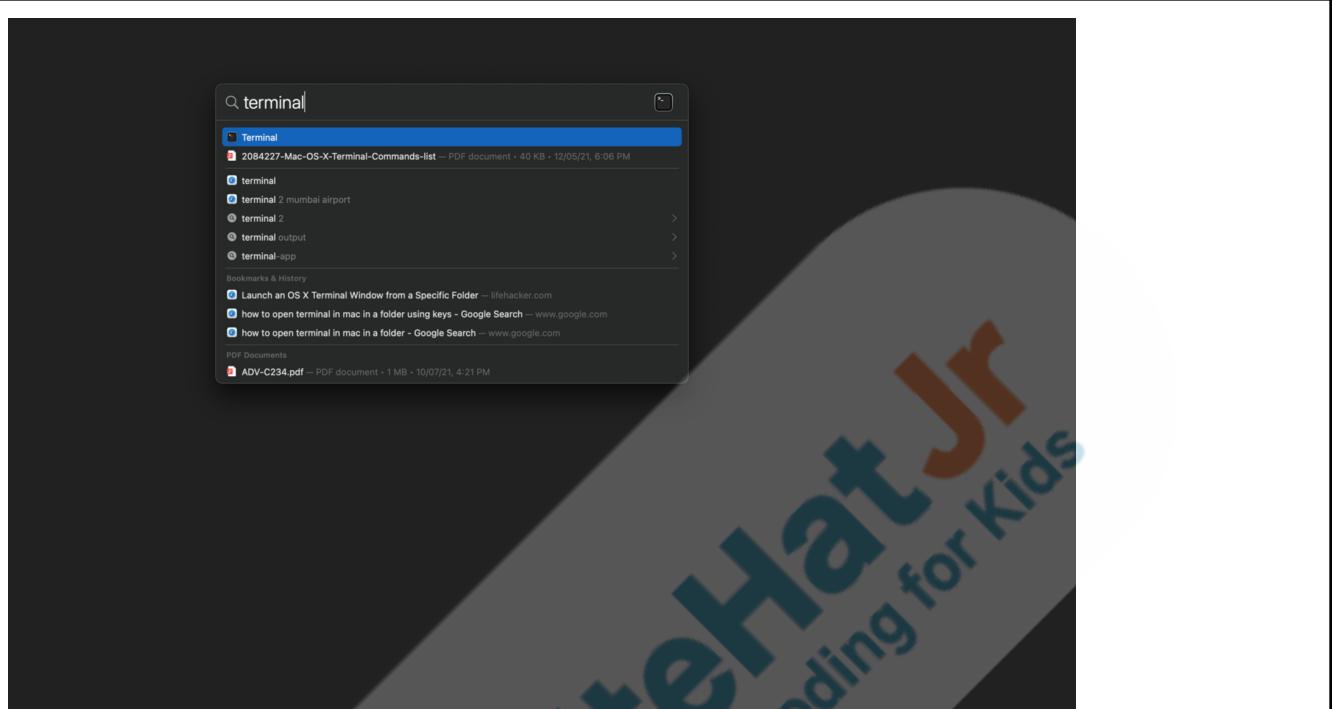
runtime - Notepad

File Edit Format View Help

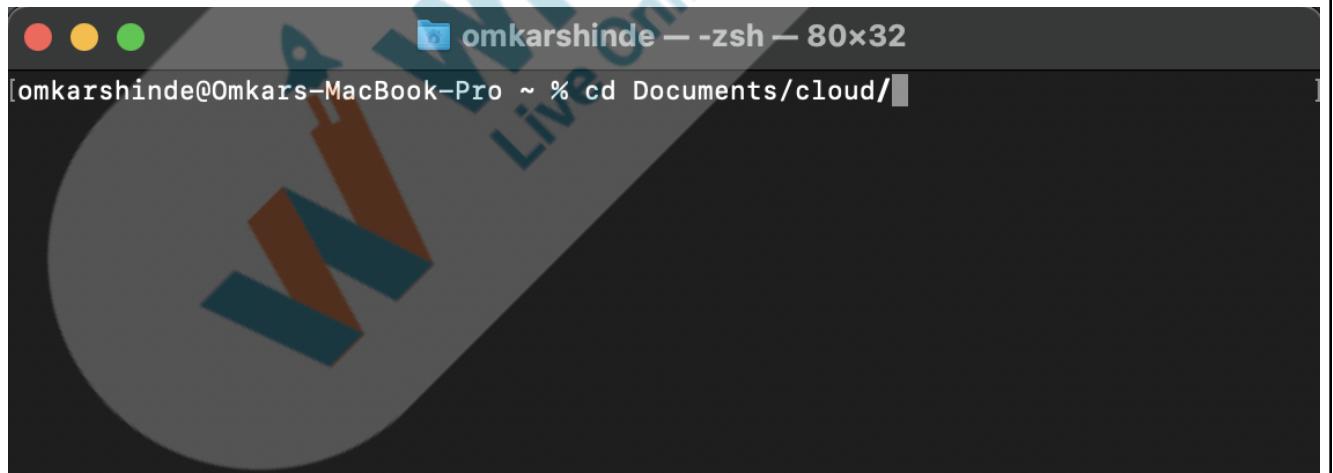
**python-3.8.5**

#### Testing the code For Mac:

1. Go to “cloud” folder in Documents directory and open the cloud folder
2. Now to run file in terminal first open terminal by pressing command + SPACEBAR key and type ‘terminal’



3. Then locate Documents folder using cd command as  
**cd Documents/cloud/**

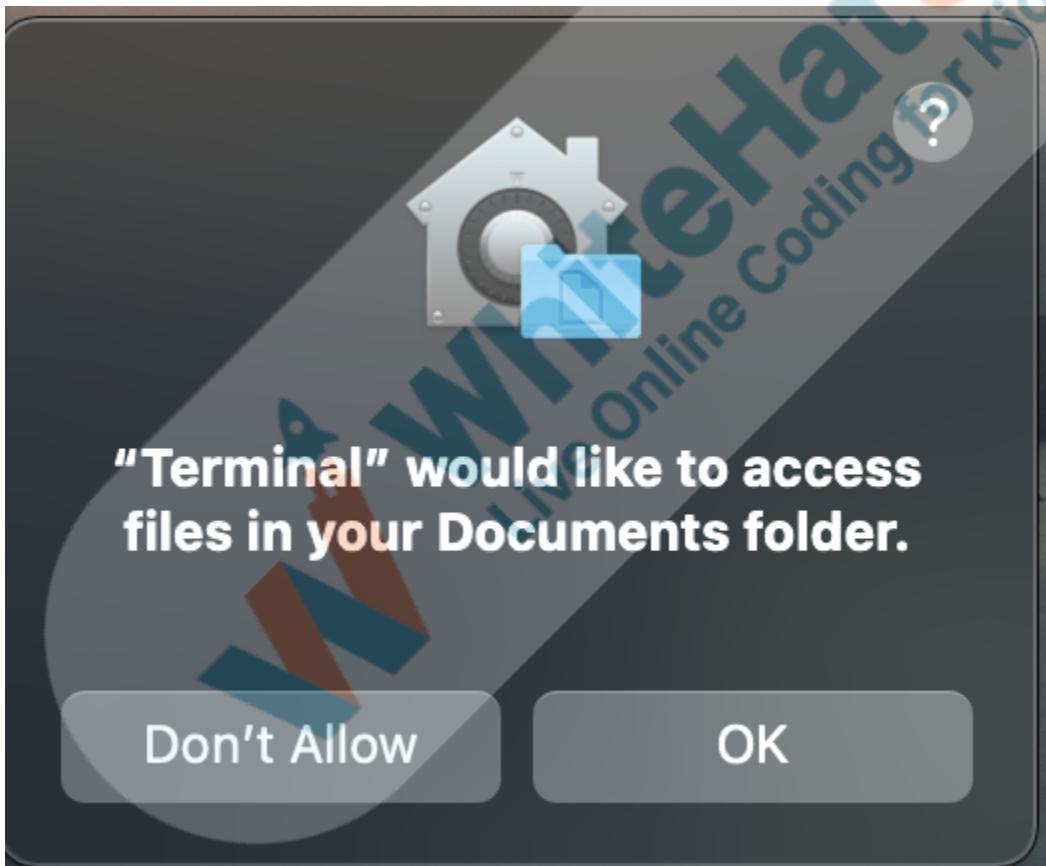


4. Now activate blockchain environment as **conda activate cloud**



```
cloud -- zsh -- 80x32
[omkarshinde@Omkars-MacBook-Pro ~ % cd Documents/cloud
[omkarshinde@Omkars-MacBook-Pro cloud % conda activate cloud
(cloud) omkarshinde@Omkars-MacBook-Pro cloud %
```

5. If you see this window please click on OK



6. And then go to class\_266 folder by **cd class\_266**

```
Class_264 — -zsh — 80x24
[(notepad) omkarshinde@Omkars-MacBook-Pro Downloads % cd Class_264
(notepad) omkarshinde@Omkars-MacBook-Pro Class_264 % ]
```

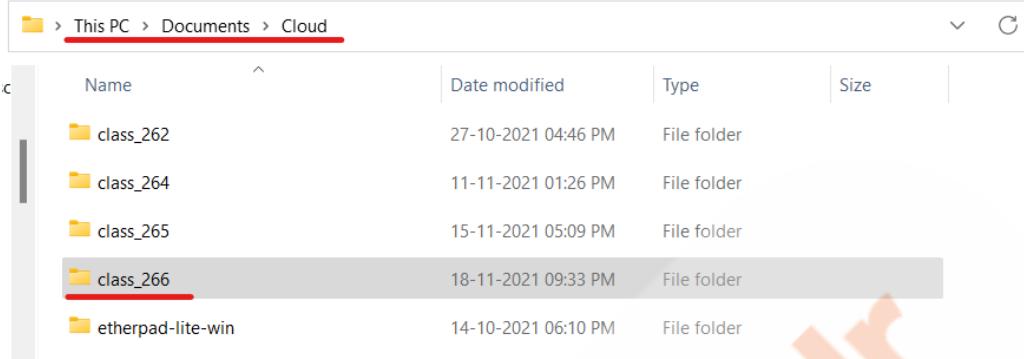
7. Now run python file as  
**python app.py**

8. It will look like this,

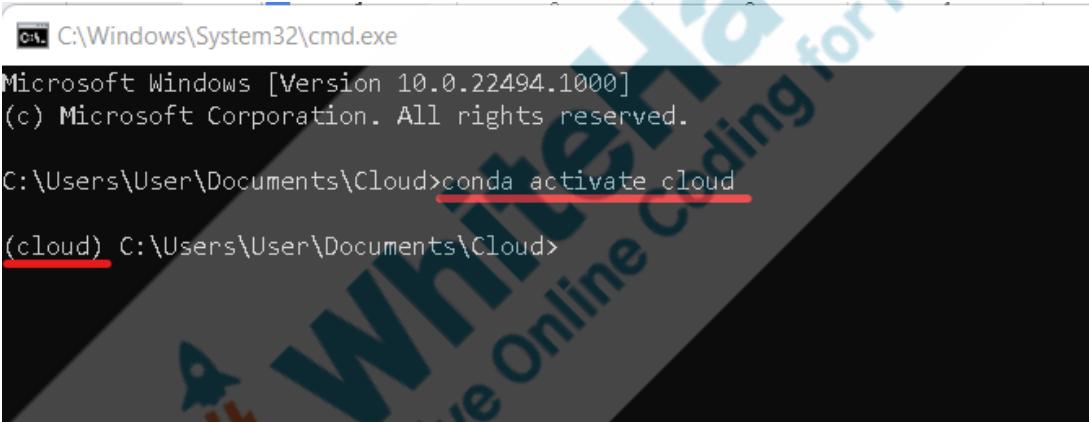
```
Class_264 — python app.py — 80x24
[(notepad) omkarshinde@Omkars-MacBook-Pro Downloads % cd Class_264
[(notepad) omkarshinde@Omkars-MacBook-Pro Class_264 % python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 ]]
```

### Testing the code For Windows :

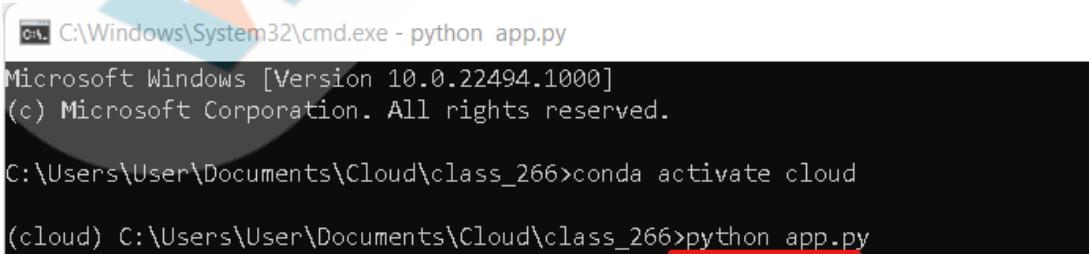
1. Now open the **CMD** in **class\_266** folder which is inside the **Cloud** folder which is located in **Document/Cloud** directory.



2. Now active the cloud environment by using the command as  
**conda activate cloud**



3. Now run python file as  
**python app.py**



4. It will look like this,

```
(cloud) C:\Users\User\Documents\Cloud\class_266>python app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

### Output Common for both MAC and WINDOWS

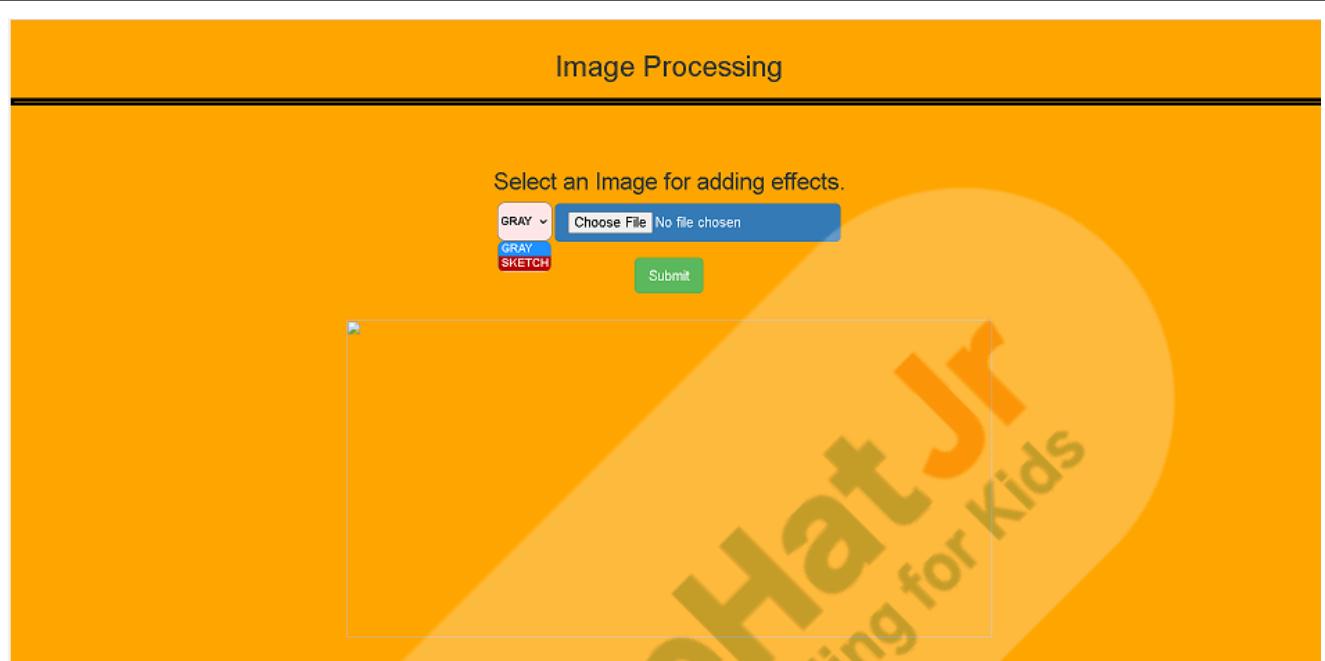
Now open your browser and paste this address : [127.0.0.1:5000](http://127.0.0.1:5000)

Here 127.0.0.1 is our host and 5000 is our port number to run our flask application

Now select the image, and click on the submit button. The image will get uploaded and displayed in the same web page as shown below and the uploaded images will get stored in the **static** folder inside **class\_c266** folder.

#### Output:

**(Note: While converting the image format, only upload jpg image not jpeg image format).**



- Converted Image into Black and White by choosing and applying the GRAY option.



- Converted Image into Sketch by choosing and applying the SKETCH option.



**(Note: The uploaded images will get stored in the static folder inside class\_c266 folder).**

| This PC > Documents > Cloud > class_266 |                     |               |      |
|---|---------------------|---------------|------|
| Name                                    | Date modified       | Type          | Size |
| static                                  | 19-11-2021 11:11 AM | File folder   |      |
| templates                               | 18-11-2021 09:33 PM | File folder   |      |
| .DS_Store                               | 18-11-2021 09:25 PM | DS_STORE File | 7 KB |
| app                                     | 19-11-2021 11:18 AM | PY File       | 2 KB |
| Aptfile                                 | 18-11-2021 09:25 PM | File          | 1 KB |
| Procfile                                | 18-11-2021 09:25 PM | File          | 1 KB |
| requirements                            | 18-11-2021 09:25 PM | Text Document | 1 KB |
| runtime                                 | 18-11-2021 09:25 PM | Text Document | 1 KB |

Now let me explain to you the logic of the code.

### Code & Explanation

Today we will be performing the below mentioned tasks:

1. Complete CSS code is given predefined for you.
2. Some of the HTML code is predefined for you, you need to complete the code.
3. Some of the python code is predefined for you, you need to complete the code.

Predefined CSS code.

**Documents\cloud\class\_266\templates\style.css**

## 1. Css Code:

- This css code is already added in the style.css file and will provide the styling to the html file.

```
1
2  body{
3      background-color: orange;
4      text-align: center;
5  }
6
7  input[type=file] {
8      display:inline-block;
9  }
10
11 img {
12     width:70%;
13 }
14
15 hr{
16     border: 5px solid black;
17 }
18
19 select {
20     height: 50px;
21     width: 70px;
22     border-radius: 10px;
23     font-weight: bold !important;
24     color: #000 !important;
25     background: #ffe6e6 !important;
26     text-transform: uppercase;
27 }
28
29 .option {
30     font-weight: bold !important;
31     color: #fff !important;
32     background: #bc0000 !important;
33     text-transform: uppercase;
34 }
```

Let's write following html code in upload.html file which is stored in **Documents\cloud\class\_266\templates\upload.html**

## 2. HTML code:

### Complete Code:

```

1  <!-- frontend code for image processing -->
2  <html>
3  <title> Image Processing </title>
4  <meta name="viewport" content="width=device-width, initial-scale=1">
5  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
6  <link rel="stylesheet" href="{{ url_for('static', filename='style.css')}}">
7
8  <body>
9  <center>
10 <br>
11  <h1>Image Processing</h1><br>
12  <form method="post" action="/" enctype="multipart/form-data"><br><br>
13  <h2>Select an Image for adding effects.</h2>
14
15  <select name="image_type_selection">
16
17      <option class="option" value="gray">Gray</option>
18      <option class="option" value="sketch">Sketch</option>
19
20  </select>
21
22  <input class="btn btn-primary btn-lg" type="file" name="file" ><br><br>
23  <input class="btn btn-success btn-lg" type="submit" value="Submit"><br><br>
24
25  </form>
26
27  <br><br><br><br>
28
29 </center>
30 </body>
31 </html>
32
33
34

```

Annotations on the code:

- Line 1-7: Title and Bootstrap added
- Line 8-13: Added center, form and few more tags inside body
- Line 14-23: Define select tag to add dropdown option for selection of image conversion and Adding Input buttons
- Line 24-27: Passing uploaded image

Predefined Code  
 Code need to perform

### Predefined code:

- In predefined code we have defined html tag , title tag, bootstrap links, style.css link and body tag.
- Inside the body tag,
  - We have defined <center> tag
  - Within the <center> tag, We have defined a <br> tag which basically breaks one line and creates space.

- Then we have h1 tag which has text as “**Image Processing**” which will get displayed on the top of the webpage.
- We had defined a form tag

**Q** Why do we use **<form>** tags in html?

**A** We use **<form>** tags to take the input from the user in various formats i.e text, file, etc.

**Q** Inside the **<form>** tag why we have used the **POST** method?

**A** **POST** method is used to take data from the form and pass it to the python code

- Then define **enctype="multipart/form-data "** . Which is the HTML function which is used when we are taking the file from the user.
- Then Inside the form tag, add an input tag that will be used by the user to choose the image.
  - Within the input tag define the class as **class="btn btn-success btn-lg"** which will provide the style to this input tag.
    - Here **btn** provides the padding and margin to the input tag.
    - **btn-success** will add dark green border and background color to the input tag.
    - **btn-lg** will increase the size of the input tag.
- Inside the form tag, add another input tag that will be used for submitting the image selected by the user to the python code.
  - In this input tag we have attribute **type="submit"** which means this input will act as a submit button and also **value="Submit"** which means submit action to be performed as we get the user selected image file.

**Q** Why do we use **<img>** tag in html?

**A** We have defined `<img>` to display the uploaded image.

**Q** Why do we use the `url_for` method?

**A** `url_for` is the function of flask which is used to get an image and pass it to the defined folder by using url.

**Q** Why do we use the `display_image()` function in the html file?

**A** `display_image()` is a function that we will write in python which will help in getting the selected image by the user.

**Q** What does the `filename` variable hold?

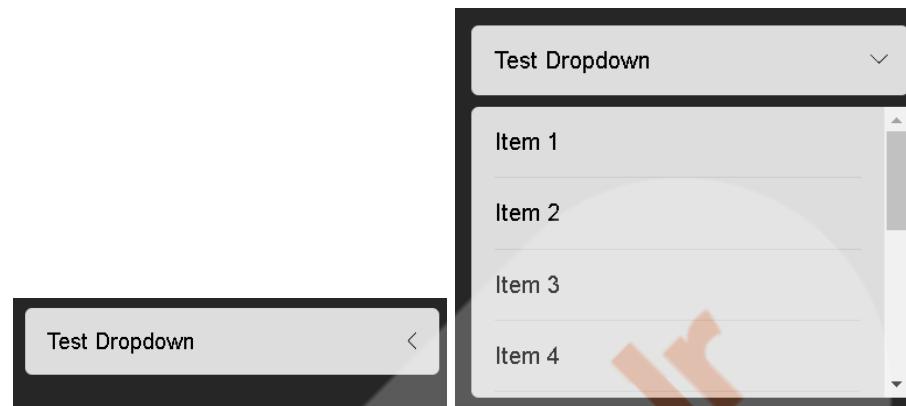
**A** `filename` is the variable which will be holding the selected image by the user.

**Code you need to write the following in the upload.html page:**

- At line no. 15, we were defining the `select` tag and this tag is used to add the multiple option to choose in the html file as dropdown menu as shown below example.

Select tag is used to create a drop-down list. Which means create a list as a dropdown to showcase the multiple options in it to select from that.

You can see the below example as shown.



Within the select tag we were writing the name "**image\_type\_selection**" by using the **name** attribute.

```
14
15      <select name="image_type_selection">
16
17
```

- Here we are setting the **name** as "**image\_type\_selection**" because this name will be used in the **app.py** file to get the selected option.
- We have to define the multiple option in the select tag to give multiple options to the user to select. In our case we will give two options: 1st gray and 2nd sketch. So users can convert the image to any of the above options.
  - So at line no.17, define the **option** tag and pass the value as Gray,
- At line no.17, define the **option** tag and pass the value as Gray,
- Now within the option tag, define the **class** attribute as "**option**" which will be used in the **style.css** file to add style in it.
- And the second parameter is **value** as "**gray**" which is used in the **app.py** file so that we can check that the user selected this option.
- Close the **option** tag.

- Now define another option by using the option tag and pass the value as Sketch.

```

 18      <option class="option" value="sketch">Sketch</option>
 19

```

- Now within the option tag, define the **class** attribute as "**option**" which will be used in the style.css file to add style in it.
- And the second parameter is **value** as "**sketch**" which is used in the app.py file so that we can check that the user selected this option.
- Close the **option** tag.

- Now, at line no. 20, end the select tag by defining the tag as `</select>`.

```

 19
 20      </select>

```

- At line no. 22, Inside the form tag, add an input tag that will be used by the user to choose the image.

```

 21
 22      <input class="btn btn-primary btn-lg" type="file" name="file" ><br><br>

```

- Within the input tag define the class as `class="btn btn-primary btn-lg"` which will provide the style to this input tag.
  - Here `btn` provides the padding and margin to the input tag.
  - `btn-primary` will add dark blue border and background color to the input tag.
  - `btn-lg` will increase the size of the input tag.
- Here we are setting the attribute `type` as `file` because we want the user to choose an image file.
- Here we are setting the `name='file'` which will be used in the python code for identifying this input tag.
- At the end of this input tag define two breaklink tags i.e `<br>` which will be used for better display of output. `<br>` define an empty single line and then go to the next. Here we had added two `<br>` which gives us two empty lines.

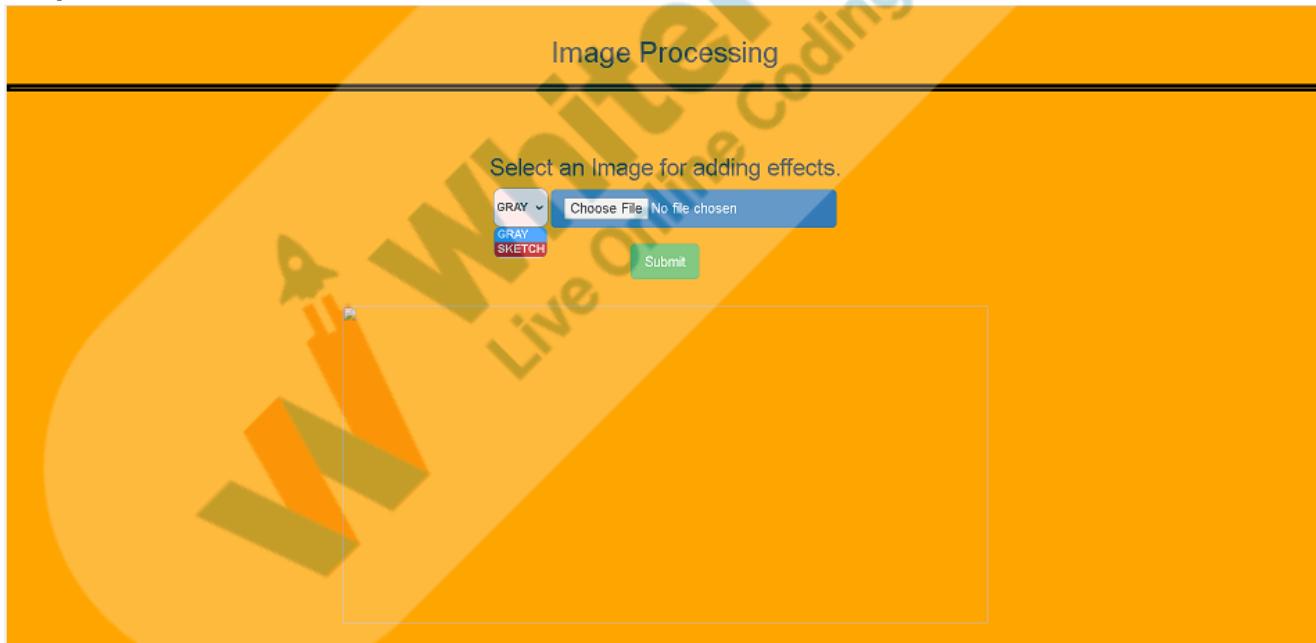
- Similarly, At line no. 23, Inside the form tag, add another input tag that will be used by the user to submit the chosen image.

 23  
 24

```
<input class="btn btn-success btn-lg" type="submit" value="Submit">
```

- Within the input tag define the class as class="btn btn-success btn-lg" which will provide the style to this input tag.
  - Here btn provides the padding and margin to the input tag.
  - btn-success will add dark green border and background color to the input tag.
  - btn-lg will increase the size of the input tag.
- Define type="submit" which means this input will act as a submit button and also value="Submit" which means submit action to be performed as we get the user selected the image file.

### Output till now:



Let's write our python file to give functionality.

Let's write following python code in app.py file which stored in  
**Documents\cloud\class\_266\app.py**

### 3. Python code.

```

1  import os
2  import cv2
3  import numpy as np
4  from flask import Flask, flash, request, redirect, url_for, render_template
5  from werkzeug.utils import secure_filename
6
7  app = Flask(__name__)
8
9  @app.route('/')
10 def upload_form():
11     return render_template('upload.html')
12
13
14 @app.route('/', methods=['POST'])
15 def upload_image():
16     operation_selection = request.form['image_type_selection']
17     image_file = request.files['file']
18     filename = secure_filename(image_file.filename)
19     reading_file_data = image_file.read()
20     image_array = np.fromstring(reading_file_data, dtype='uint8')
21     decode_array_to_img = cv2.imdecode(image_array, cv2.IMREAD_UNCHANGED)
22
23
24
25     if operation_selection == 'gray':
26         file_data = make_grayscale(decode_array_to_img)
27     elif operation_selection == 'sketch':
28         file_data = image_sketch(decode_array_to_img)
29     else:
30         print('No image selected')
31
32
33     with open(os.path.join('static/', filename),
34               'wb') as f:
35         f.write(file_data)
36
37     return render_template('upload.html', filename=filename)
38

```



**Importing Libraries**

**Initialised Flask**

**Route where we want to use this function.**

**Upload image function**

**Applying if...else condition on the select option to perform given function**

**Read and pass the image from uploaded folder and apply write function**

**Predefined Code**

**Code need to perform**

```

38
39 ▼ def make_grayscale(decode_array_to_img):
40
41     converted_gray_img = cv2.cvtColor(decode_array_to_img, cv2.COLOR_RGB2GRAY)
42     status, output_image = cv2.imencode('.PNG', converted_gray_img)
43
44     return output_image
45
46
47 # starts
48 ▼ def image_sketch(decode_array_to_img):
49
50     converted_gray_img = cv2.cvtColor(decode_array_to_img, cv2.COLOR_BGR2GRAY)
51     sharpening_gray_img = cv2.bitwise_not(converted_gray_img)
52     blur_img = cv2.GaussianBlur(sharpening_gray_img, (111, 111), 0)
53     sharpening_blur_img = cv2.bitwise_not(blur_img)
54     sketch_img = cv2.divide(converted_gray_img, sharpening_blur_img, scale=256.0)
55     status, output_img = cv2.imencode('.PNG', sketch_img)
56
57     return output_img
58 # ends
59
60 @app.route('/display/<filename>')
61 ▼ def display_image(filename):
62
63     return redirect(url_for('static', filename=filename))
64
65
66
67 if __name__ == "__main__":
68     app.run()
69

```

Define Grayscale function

Define Sketch() function and convert image into sketch

Display Uploaded image in webpage

Run the flask app

Predefined Code

Code need to perform

**Below Predefined code given in app.py file:**

### Importing libraries:

- Initially we have imported a few libraries such as flask library, Os library, werkzeug.utils Library, cv2 library and numpy library.

```

import os
import cv2
import numpy as np
from flask import Flask, flash, request, redirect, url_for, render_template
from werkzeug.utils import secure_filename

```

Q

What is the use of **os library**?

A

The **os library** is used to read and write on the image file which is

stored in our system.

**Q** What is the use of the **cv2 library**?

**A** **cv2 library** i.e **Opencv** is used to work with image processing which means converting images from one format to another.

**Q** What is the use of the **numpy library**?

**A** **numpy library** is used to work on the creation of an array of image pixels which will help to make changes in pixel values to get the desired output image.

### Initializing Flask:

- Then, At line no. 06, we had initialized Flask using the Flask function.

```
6  app = Flask(__name__)
```

**Q** What is the use of the **app = Flask(\_\_name\_\_)**?

**A** **app = Flask(\_\_name\_\_)** initialized the flask by passing **\_\_name\_\_** in the flask function as we want to run our web application using flask.

### Routing to home page and loading upload.html file:

- Then, At line no. 09, we are directing our web page to perform a function.

```
8  # Open and redirect to default
9  @app.route('/')
```

- At line no. 10, we have defined a `load_form()` function.

**Q**

Which function is used to render the html file?

**A**

`render_template()` method is used to render the `upload.html` webpage.

```

1
2
3
4
5
6
7
8  # Open and redirect to default upload webpage
9  @app.route('/')
10 def load_form():
11     return render_template('upload.html')
12

```

### Routing to home page and executing `upload_image()` function:

- Then from line no. 14, we had written a function to upload the selected image from the user into our static folder.

```

13
14 @app.route('/', methods=['POST'])
15 def upload_image():
16     operation_selection = request.form['image_type_selection']
17     image_file = request.files['file']
18     filename = secure_filename(image_file.filename)
19     reading_file_data = image_file.read()
20     image_array = np.fromstring(reading_file_data, dtype='uint8')
21
22     decode_array_to_img = cv2.imdecode(image_array, cv2.IMREAD_UNCHANGED)
23

```

- Then at line no. 16, we had used the `request.files[]` method of request to get the chosen option from the select tag of html by passing the “`image_type_selection`” name which was given to the select tag in html and then store this in the `operation_selection` variable.

```

15  def upload_image():
16      operation_selection = request.form['image_type_selection']

```

**Q** What is the functionality of the POST method?

**A** The POST method will help us to take data from the form and pass it to the defined url.

- Then we created the upload\_image function

```

def upload_image():
    operation_selection = request.form['image_type_selection']
    image_file = request.files['file']
    filename = secure_filename(image_file.filename)
    reading_file_data = image_file.read()
    image_array = np.fromstring(reading_file_data, dtype='uint8')
    decode_array_to_img = cv2.imdecode(image_array, cv2.IMREAD_UNCHANGED)

```

- First we get the selected file from the user and store it in operation\_selection variable
- Then we used the secure\_filename function to secure the image file stored in the operation\_selection variable.
- Then we read the secured image stored in the filename variable.
- Then we converted the image to an array using the fromstring function.

**Q** What does **cv2.imdecode()** do?

**A** **cv2.imdecode()** convert the numeric array of image pixels back to the image which will be further passed to convert the image into a black and white image

**Q** Why do we use the **render\_template()** function?

**A** We use **render\_template()** function to display the uploaded image

on the browser and the messenger for successfully uploading the image, so we used flask's `render_template()` method and passed the `upload.html` file along with the image name stored in the `filename` variable and displayed a message in it.

```
37     return render_template('upload.html', filename=filename)
38
```

### Make\_grayscale function:

**Q** What does `make_grayscale()` function do?

**A** The `make_grayscale()` function will convert the uploaded image into black and white and store the converted image into a static folder as PNG format.

```
38
39 ▼ def make_grayscale(decode_array_to_img):
40
41     converted_gray_img = cv2.cvtColor(decode_array_to_img, cv2.COLOR_RGB2GRAY)
42     status, output_image = cv2.imencode('.PNG', converted_gray_img)
43
44     return output_image
45
```

### Display\_image() function

**Q** What does `display_image()` function do?

**A** The `display_image()` function is used to display the uploaded and converted image. This image will get displayed on the html page when we run the localhost on web browser.

```

59
60  @app.route('/display/<filename>')
61 ▼ def display_image(filename):
62
63      return redirect(url_for('static', filename=filename))
64
65

```

- In the beginning of the code we had initialized our flask application like this  
`app = Flask(__name__)`
- So At line 67

```

66
67  if __name__ == "__main__":
68      app.run()
69

```

We will check if we are still under flask code by writing the if condition and if the condition satisfies then we will run our flask application by using run() function followed by the variable app which holds the initialization of flask.

- This will result in running url 127.0.0.1:5000

### So let's Start the coding for today's class.

- Now, we are going to apply an if-else statement to the operation\_selection variable which holds the selected option which the user has chosen for image processing from the select tag of HTML.

```

25  if operation_selection == 'gray':
26      file_data = make_grayscale(decode_array_to_img)
27  elif operation_selection == 'sketch':
28      file_data = image_sketch(decode_array_to_img)
29  else:
30      print('No image')
31

```

- At line no. 25, we had applied the if condition on **operation\_selection** to check if it contains '**gray**'. Which means if the selected option is gray then we are going to make an image as black.

```

25     if operation_selection == 'gray':
26         file_data = make_grayscale(decode_array_to_img)
  
```

- Here, in the **if** statement we will apply the condition as **operation\_selection == 'gray'**.
- If this condition is satisfied, then we will call **make\_grayscale** function and pass the **decode\_array\_to\_img** variable which holds the decoded image pixels which we want to convert into a black and white image and store this in **file\_data** variable.
- At line no. 27, we had applied the **elif** condition on **operation\_selection** on the variable as '**sketch**'. Which means if the selected option is **sketch** then we are going to make an image as in sketch form.

```

27     elif operation_selection == 'sketch':
28         file_data = image_sketch(decode_array_to_img)
  
```

- Here, in the **elif** statement we had applied the condition as **operation\_selection == 'sketch'**.
- If this condition is satisfied, then we will call **image\_sketch()** function and pass the **decode\_array\_to\_img** variable which holds the decoded image pixels which we want to convert into a sketched image and store this in **file\_data** variable.
- At line no. 29, we have to apply the **else** condition, if both the above options were not selected, then within the **else** we will use the **print()** statement to print the text as '**No image selected**'.

```

29     else:
30         print('No image selected')
  
```

### Image sketch function:

- Now at line no. 48, we are going to define the `image_sketch()` function in which we are applying the various methods to convert the uploaded image into a sketch.

```

48  def image_sketch(decode_array_to_img):
49
50      converted_gray_img = cv2.cvtColor(decode_array_to_img, cv2.COLOR_BGR2GRAY)
51      sharpening_gray_img = cv2.bitwise_not(converted_gray_img)
52      blur_img = cv2.GaussianBlur(sharpening_gray_img, (111, 111), 0)
53      sharpening_blur_img = cv2.bitwise_not(blur_img)
54      sketch_img = cv2.divide(converted_gray_img, sharpening_blur_img, scale=256.0)
55      status, output_img = cv2.imencode('.PNG', sketch_img)
56
57      return output_img

```

- At line no. 48, we had defined the `image_sketch()` function and passed the `decode_array_to_img` variable in it.

```

48  def image_sketch(decode_array_to_img):
49

```

- Here `decode_array_to_img` is a variable which holds the decoded image pixels of an uploaded image.
- Now at line no. 50, we have apply the `cv2.cvtColor()` function of opencv to convert the image into black and white by using `cv2.COLOR_RGB2GRAY` and store that in the `converted_gray_img` `converted_gray_img` variable.

```

49
50      converted_gray_img = cv2.cvtColor(decode_array_to_img, cv2.COLOR_BGR2GRAY)

```

- cv2.cvtColor** - This function is used to convert the color of an image from one type to another type.  
It takes 2 parameters :decode image and `COLOR_RGB2GRAY` function.
- decode\_array\_to\_img** is the variable which holds the decoded array of image pixels back to the original image which will be used to convert into a black and white image.
- cv2.COLOR\_RGB2GRAY** - It is the function of OpenCV which converts the colored image into Black and White.
- converted\_gray\_img** - this variable will hold the converted Black and white image.

- Here is an example of a gray image after conversion of a colored image.



- Now at line no. 51, we have to apply the **cv2.bitwise\_not()** function of the opencv library on **converted\_gray\_img** to sharpen the converted grayscale image and store the outcome in the **sharpening\_gray\_img** variable.

```
51     sharpening_gray_img = cv2.bitwise_not(converted_gray_img)
```

- **cv2.bitwise\_not()** This function of opencv is used for sharpening

the image. Sharpening of an image means enhancing the definition of edges in an image.

This function takes 1 parameter which the image on which sharpening effects needs to be added.

- **Converted\_gray\_img** is the variable which holds the converted black and white image pixels.
- **sharpening\_gray\_img** - this variable will hold the converted sharpen image.
- You can see the difference between how a normal image looks and how a sharp image looks after applying sharpness.



- Here is a sample output of a sharpened gray scale image.



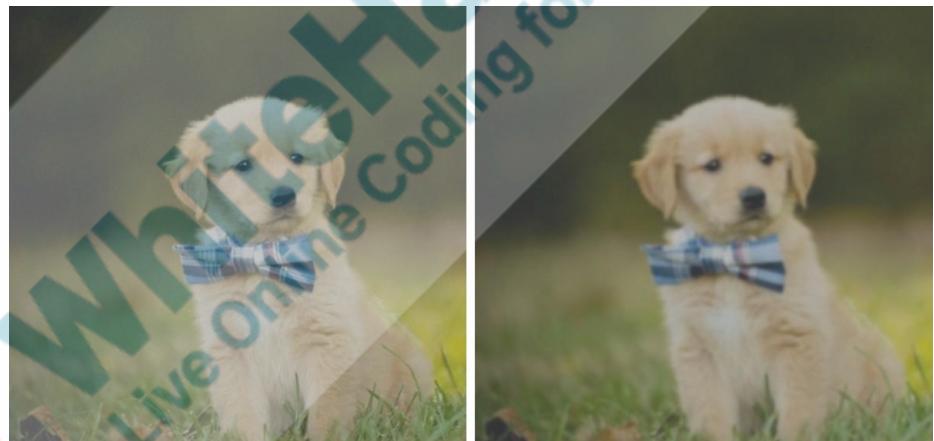
- Now at line no. 52, we have to apply the **cv2.GaussianBlur()** function of the opencv library on **sharpen\_gray\_img** variable to apply blurriness on the sharpen image and store the outcome in the **blur\_img** variable.

```
52     blur_img = cv2.GaussianBlur(sharpen_gray_img, (111, 111), 0)
```

- **cv2.GaussianBlur()** This function of opencv is used to apply blurriness on images.

This function takes 2 parameters: sharpen image and width, height, and angle.

- **sharpening\_gray\_img** is the variable which holds the converted sharpen image pixels.
- **(111, 111)** - is the defined height and width value till where we need to apply a blur filter. Here we are defining this value as the height and width of the output image to display.
- **0** - is the angle value which means we need to apply the blur filter. But here we need to apply the blur filter on the whole image equally that's why we had given the value as 0.
- You can see the difference between how a normal image looks and how a GaussianBlur image looks after applying GaussianBlur.



- Here is a sample output of a blur image after conversion of a sharpened grayscale image.



- Now at line no. 53, we have to apply the **cv2.bitwise\_not()** function of the opencv library on **blur\_img** to sharpen the converted blur image and store the outcome in the **sharpening\_blur\_img** variable.
- This function will enhance the edges of the image to get the more depth

of image resolution so that we will get the detailing of the image clear.

```
53     sharpening_blur_img = cv2.bitwise_not(blur_img)
```

- **cv2.bitwise\_not()** This function of opencv is used for sharpening the image. This function takes 1 parameter which the image on which sharpening effects needs to be added.
- **blur\_img** is the variable which holds the converted blur image pixels.
- **sharpening\_blur\_img** - this variable will hold the converted sharpen image.
- Here is a sample output of a sharpened blur image after conversion of a blur image.



- Now at line no. 54, we have to apply the `cv2.divide()` function of the opencv library and pass the `converted_gray_img` variable and `sharpening_blur_img` variable which will divide the image pixels and on the `scale` as `256` i.e range to get the sketch image and store the outcome in the `sketch_img` variable.

54

```
sketch_img = cv2.divide(converted_gray_img, sharpening.blur_img, scale=256.0)
```

- **cv2.divide()** is the function of opencv which takes three parameters pixels of black and white image(image 1), pixels of sharpened image(image 2) and scale value, then the image 1 and image 2 are as divides the pixel values of **converted\_gray\_img** and **sharpening.blur\_img**
- **converted\_gray\_img** is the variable which holds the converted black and white image pixels.
- **sharpening.blur\_img** - this variable will hold the converted sharpen image.
- **scale=256.0** is the range to get the sketch image.
- **sketch\_img** - this variable will hold the converted sketch image.
- Here is a sample output of a sketch image after conversion of a sharpened blur image.



- Now at line no. 55, Then encode the converted sketch image so that the image will be at its original format which means its original height and

width will be stored in the **output\_img** variable.

```
55     status, output_img = cv2.imencode('.PNG', sketch_img)
```

- Here **cv.imencode** is the function of OpenCV library which encodes the image to its original format by maintaining its original height and width even after modifying the pixel values.
- Here **cv.imencode** requires two parameters i.e the output format of image and the converted image.
  - **'.PNG'** is the format which we want to get as output.
  - **sketch\_img**- It is the variable which stores the converted pixels of a sketch image.
- **output\_img** is the variable which stores the converted sketch image in the format of PNG.
- **status** - It is the parameter which gives the output as boolean value i.e True or False. If the image gets successfully converted then the output will be True or else the output value will be False.
- Now at line no. 57, we will define the return function and return the Converted Sketched image as output and display it in the Html page.
- It will return the converted image as output which will be displayed on webportal.

```
57     return output_img
```

## Output:

**(Note: While converting the image format, only upload jpg image not jpeg image format).**



- Converted Image into Black and White by choosing and applying the GRAY option.



- Converted Image into Sketch by choosing and applying the SKETCH option.



**(Note: The uploaded images will get stored in the static folder inside class\_c266 folder).**

### Teacher Stops Screen Share

Now it is your turn.

- Ask the Student to press the ESC key to come back to the panel.
- Guide the Student to start Screen Share.
- The Teacher gets into Fullscreen.

**Step 3:  
Student-Led  
Activity  
(30 mins)**



[\*\*Student-Activity-1- PREDEFINED  
CODE\*\*](#)

[\*\*Student Activity 2 - CODE  
DIAGRAM\*\*](#)

**Task 01:**

1. Download the source code from [Student-Activity-1](#)
2. Unzip the downloaded source code in the document\cloud folder.
3. Some code is given predefined. You need to complete the code.
  - In CSS file
    - Complete code for styling is given.
  - In HTML file
    - Major HTML code is given to you which you had coded in previous class.
  - In a Python file.
    - Major python code is given to you which you had coded in previous class

**Coding:**

1. Complete Predefined CSS code.

```
1
2 body{
3     background-color: orange;
4     text-align: center;
5 }
6
7 input[type=file] {
8     display:inline-block;
9 }
10
11 img {
12     width:70%;
13 }
14
15 hr{
16     border: 5px solid black;
17 }
18
19 select {
20     height: 50px;
21     width: 70px;
22     border-radius: 10px;
23     font-weight: bold !important;
24     color: #000 !important;
25     background: #ffe6e6 !important;
26     text-transform: uppercase;
27 }
28
29 .option {
30     font-weight: bold !important;
31     color: #fff !important;
32     background: #bc0000 !important;
33     text-transform: uppercase;
34 }
```

2. Complete the HTML code

```

1  <!-- frontend code for image processing -->
2  <html>
3  <title> Image Processing </title>
4  <meta name="viewport" content="width=device-width, initial-scale=1">
5  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
6  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}>
7
8  <body>
9  <center>
10 <br>
11  <h1>Image Processing</h1><hr>
12  <form method="post" action="/" enctype="multipart/form-data"><br><br>
13  <h2>Select an Image for adding effects.</h2>
14
15  <select name="image_type_selection">
16
17      <option class="option" value="gray">Gray</option>
18      <option class="option" value="sketch">Sketch</option>
19
20  </select>
21
22  <input class="btn btn-primary btn-lg" type="file" name="file" ><br><br>
23  <input class="btn btn-success btn-lg" type="submit" value="Submit"><br><br>
24
25  </form>
26
27  Step 4:<br/>Wrap-Up<br/>(5 mins)</b> | <p>You did great today as well.<br/>Great! You have two hats off.</p> <p><b>Q</b> What does <code>cv2.divide()</code> use function for?</p> <p><b>A</b> <code>cv2.divide()</code> is the function of opencv which divides the pixel values of <code>converted_gray_img</code> and <code>sharpening_blur_img</code></p> <p><b>Q</b> What is the use of the <code>cv2.bitwise_not()</code> function?</p> <p><b>A</b> <code>cv2.bitwise_not()</code> function of opencv is used for sharpening the image.</p> | <p>(Give at least 2 hats offs)<br/>Press the <b>Hats Off</b> Icon for <b>Creatively Solving Activities</b>.</p>  <p>Press the <b>Hats Off</b> Icon for <b>Great Question</b>.</p>  <p>Press the <b>Hats Off</b> Icon for <b>Strong Concentration</b>.</p>  <p><b>If you don't have time to perform additional activities, ask the</b></p> |
|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|  |  |                                                                                                                                                                                                                                                                                                                                                                      |
|--|--|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  |  | <p>student to perform all the additional activities after the class. Additional activities are <b>VERY</b> important for kids, so that they are ready for the next module. And some challenging concepts are coming ahead.</p> <p>Also remind the student to refer to the Student Reference activity for increasing his/her knowledge. This should also be done.</p> |
|--|--|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

#### Teacher Initiates Screen Share

|                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                       |
|----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| <p><b>Step 5:</b><br/> <b>Project</b><br/> <b>Pointers and</b><br/> <b>Cues</b><br/> <b>(5 mins)</b></p> | <p><b>ROTATING IMAGE</b></p> <p><b>Goals of the Project:</b></p> <p>In today's class, we learnt about Image Processing and its different types of Images that are made up of various pixel combinations. We learnt about image processing and steps of the same. Also we saw how to use OpenCv for image processing.</p> <p><b>ROTATING IMAGE:</b></p> <p>In this project, we want you to create a web based application to do the task of uploading an image and rotating the image as per the angle value by the user on an HTML</p> | <p>Open the <b>Project Solution link</b> and showcase the output.</p> |
|----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|

|                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                    |  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|                                                                                                                                                                                                              | <p>file using the Python and Flask concepts.</p> <p><b>Story:</b><br/>Raju is a smart student like you. When he was looking at photos in the gallery, he was amazed about image rotation. So he planned to execute it by doing code. He is in need of help from you. As a coding expert, please help him to write the code by using python code for rotating the image and hosting the same.</p> <p>Good Luck!</p> |  |
| <p><b>For the solution of all the Additional Activity, open <a href="#">Teacher-Activity-3</a> and navigate to class number <a href="#">C266</a>.</b></p>                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                    |  |
| <p><b>Additional Activity 1 -</b><br/>Run <a href="#">Student-Activity-3</a> from the <a href="#">panel</a>.<br/>The <a href="#">TASK</a> and <a href="#">HINTS</a> are mentioned on the website itself.</p> |                                                                                                                                                                                                                                                                                                                                                                                                                    |  |
| <p><b>Additional Activity 2 -</b><br/>Run <a href="#">Student-Activity-4</a> from the <a href="#">panel</a>.<br/>The <a href="#">TASK</a> and <a href="#">HINTS</a> are mentioned on the website itself.</p> |                                                                                                                                                                                                                                                                                                                                                                                                                    |  |
| <p><b>Additional Activity 3 -</b><br/>Run <a href="#">Student-Activity-5</a> from the <a href="#">panel</a>.<br/>The <a href="#">TASK</a> and <a href="#">HINTS</a> are mentioned on the website itself.</p> |                                                                                                                                                                                                                                                                                                                                                                                                                    |  |
| <p><b>Additional Activity 4 -</b><br/>Run <a href="#">Student-Activity-6</a> from the <a href="#">panel</a>.<br/>The <a href="#">TASK</a> and <a href="#">HINTS</a> are mentioned on the website itself.</p> |                                                                                                                                                                                                                                                                                                                                                                                                                    |  |
| <p><b>Additional Activity 5 -</b></p>                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                    |  |

Run **Student-Activity-7** from the panel.  
 The **TASK** and **HINTS** are mentioned on the website itself.

Teacher Clicks

× End Class

| Activity                     | Activity Name           | Links                                                                                                                                                                                                                                                                           |
|------------------------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Teacher Activity 1           | SOURCE CODE             | <a href="https://drive.google.com/drive/folders/1kBpN7y2SQKBhckqwuQ4ocgeSWo1skxp2?usp=sharing">https://drive.google.com/drive/folders/1kBpN7y2SQKBhckqwuQ4ocgeSWo1skxp2?usp=sharing</a>                                                                                         |
| Teacher Activity 2           | CODE DIAGRAM            | <a href="https://docs.google.com/document/d/e/2PACX-1vThkFoXlxINLe858Stfpz9Y2YAqD0moDmfj7DDa3uCa895VKuBtSItXg5qL-bhRN DsbRHr84gvHB2Hp/pub">https://docs.google.com/document/d/e/2PACX-1vThkFoXlxINLe858Stfpz9Y2YAqD0moDmfj7DDa3uCa895VKuBtSItXg5qL-bhRN DsbRHr84gvHB2Hp/pub</a> |
| Teacher Activity 3           | ADDITIONAL ACTIVITY     | <a href="https://mahdihat791.github.io/v2/additional_activities_solution.html">https://mahdihat791.github.io/v2/additional_activities_solution.html</a>                                                                                                                         |
| Student Activity 1           | PREDEFINED CODE         | <a href="https://drive.google.com/drive/folders/1bhdM-wFXicywjBcayS4tsJXu-rGJzaZ6?usp=sharing">https://drive.google.com/drive/folders/1bhdM-wFXicywjBcayS4tsJXu-rGJzaZ6?usp=sharing</a>                                                                                         |
| Student Activity 2           | CODE DIAGRAM            | <a href="https://docs.google.com/document/d/e/2PACX-1vThkFoXlxINLe858Stfpz9Y2YAqD0moDmfj7DDa3uCa895VKuBtSItXg5qL-bhRN DsbRHr84gvHB2Hp/pub">https://docs.google.com/document/d/e/2PACX-1vThkFoXlxINLe858Stfpz9Y2YAqD0moDmfj7DDa3uCa895VKuBtSItXg5qL-bhRN DsbRHr84gvHB2Hp/pub</a> |
| Student Reference Activity 1 | Image Processing        | <a href="https://www.sciencedirect.com/topics/engineering/image-processing">https://www.sciencedirect.com/topics/engineering/image-processing</a>                                                                                                                               |
| Student Reference Activity 2 | OpenCV                  | <a href="https://docs.opencv.org/4.x/d2/d96/tutorial_py_table_of_contents_imgproc.html">https://docs.opencv.org/4.x/d2/d96/tutorial_py_table_of_contents_imgproc.html</a>                                                                                                       |
| Student Reference Activity 3 | RECORD SCREEN REFERENCE | <a href="https://curriculum.whitehatjr.com/ADV+Asset/C145+-Guide+to+loom+recording.pdf">https://curriculum.whitehatjr.com/ADV+Asset/C145+-Guide+to+loom+recording.pdf</a>                                                                                                       |

|                  |                |                                                                                                                                                                                                                                                                               |
|------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Project solution | Rotating Image | <a href="https://docs.google.com/document/d/e/2PA_CX-1vQ_gkxx8jh6bwUW8S019TcW-fwJ8OcsUctrST5eCpLsUlyffkfYj5vIb_uXxKibL5weYYD7pE5KSdJ/pub">https://docs.google.com/document/d/e/2PA_CX-1vQ_gkxx8jh6bwUW8S019TcW-fwJ8OcsUctrST5eCpLsUlyffkfYj5vIb_uXxKibL5weYYD7pE5KSdJ/pub</a> |
|------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

