Image to Cartoon

ANIMATE THE WORLD TOGETHER

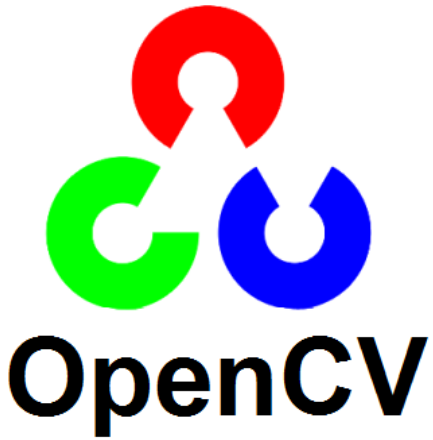
Presented By,
Div P, Group 09,

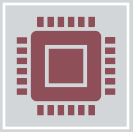
43-Atharvsinh Sawant
44- Avadhut Sawant
45- Sahil Sawant
46- Yashraj Sawant
47- Sanchit Sawarkar

- In the Memory of the Greatest Poineer Of Cartoons and Animation :

Walt Disney

OpenCV





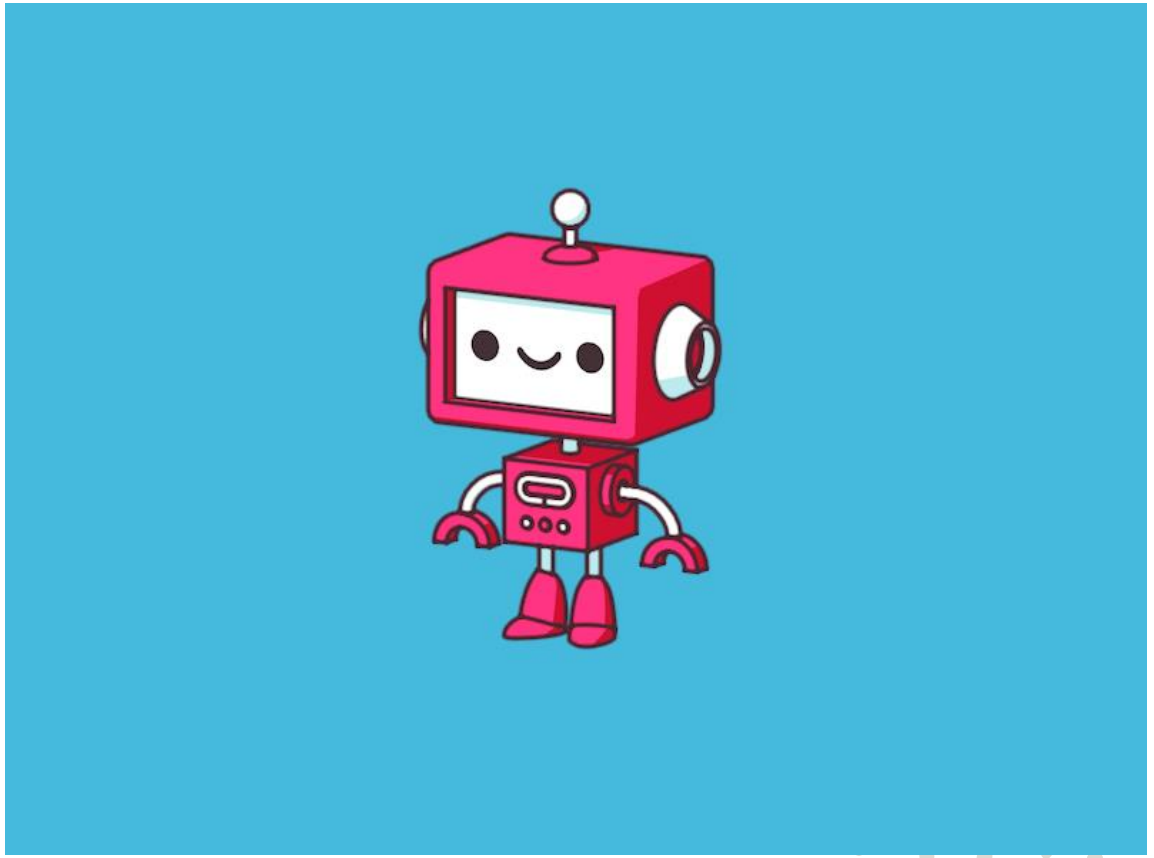
OpenCV – Open Source Computer Vision.

It is one of the most widely used tools for computer vision and image processing tasks.

It is used in various applications such as face detection, video capturing, tracking moving objects, object disclosure,

Nowadays in Covid applications such as face mask detection, social distancing, and many more.

A OpenCV is the Computer's View of Outside World



Literature review

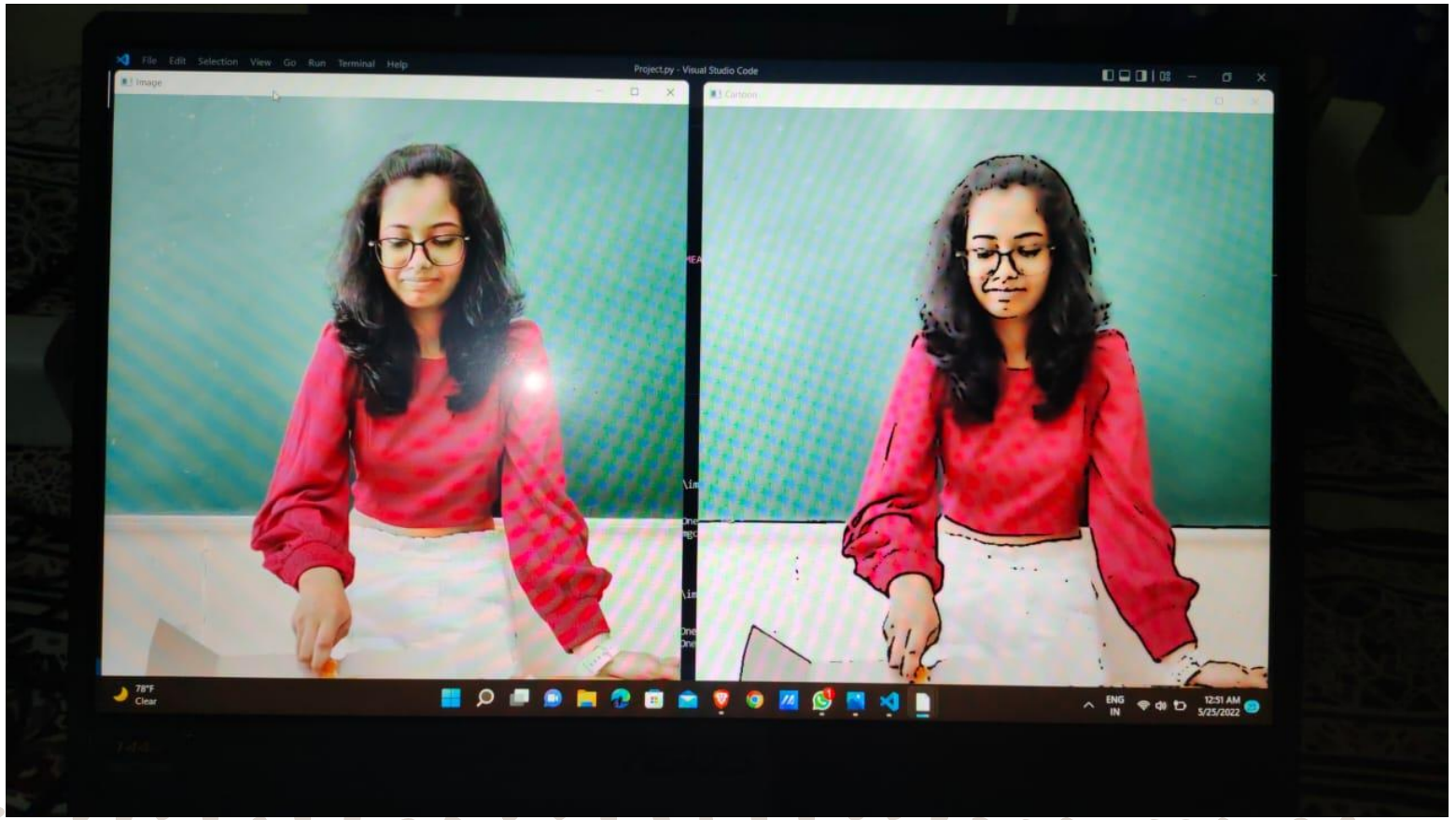
We took inspiration from different sites such as Cartoonize.net where we can convert a photo into cartoon image.We also refer some other website life kapwing , befunky .We took reference from different learning platforms like GeeksforGeeks

Discussion

- Edge Detection and Image Gradient
- Dilation, Opening, Closing, Errosion
- Perspective Transformation
- Grey Color (Black and White)
- Image Blur
- Edge Detector

There are different types of the Functions to pick from.

We, are going to cover only 3 of them.



As Students of Vishwakarma Institute of Technology, we tried to make use of the Project in Real Life, and asked our Friend to be our first "TestSubject-1"

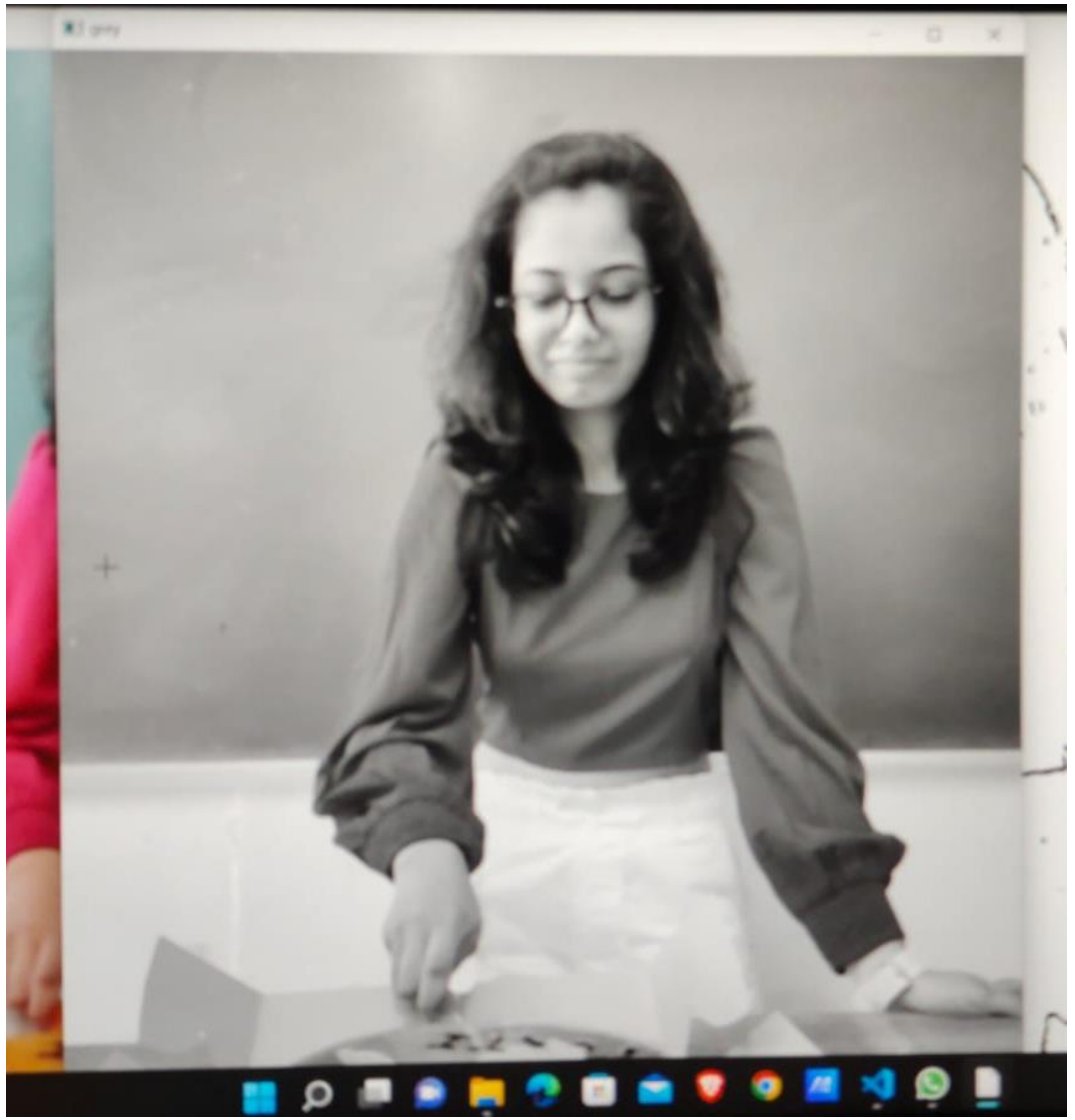
Image to Cartoon

GREY

- As first input, this function receives the original image. As second input, it receives the color space conversion code. Since we want to **convert our original image from the BGR color space to gray**, we use the code COLOR_BGR2GRAY. Now, to display the images, we simply need to call the imshow function of the cv2 module.
- **In Code :-**

```
grey = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

GREY

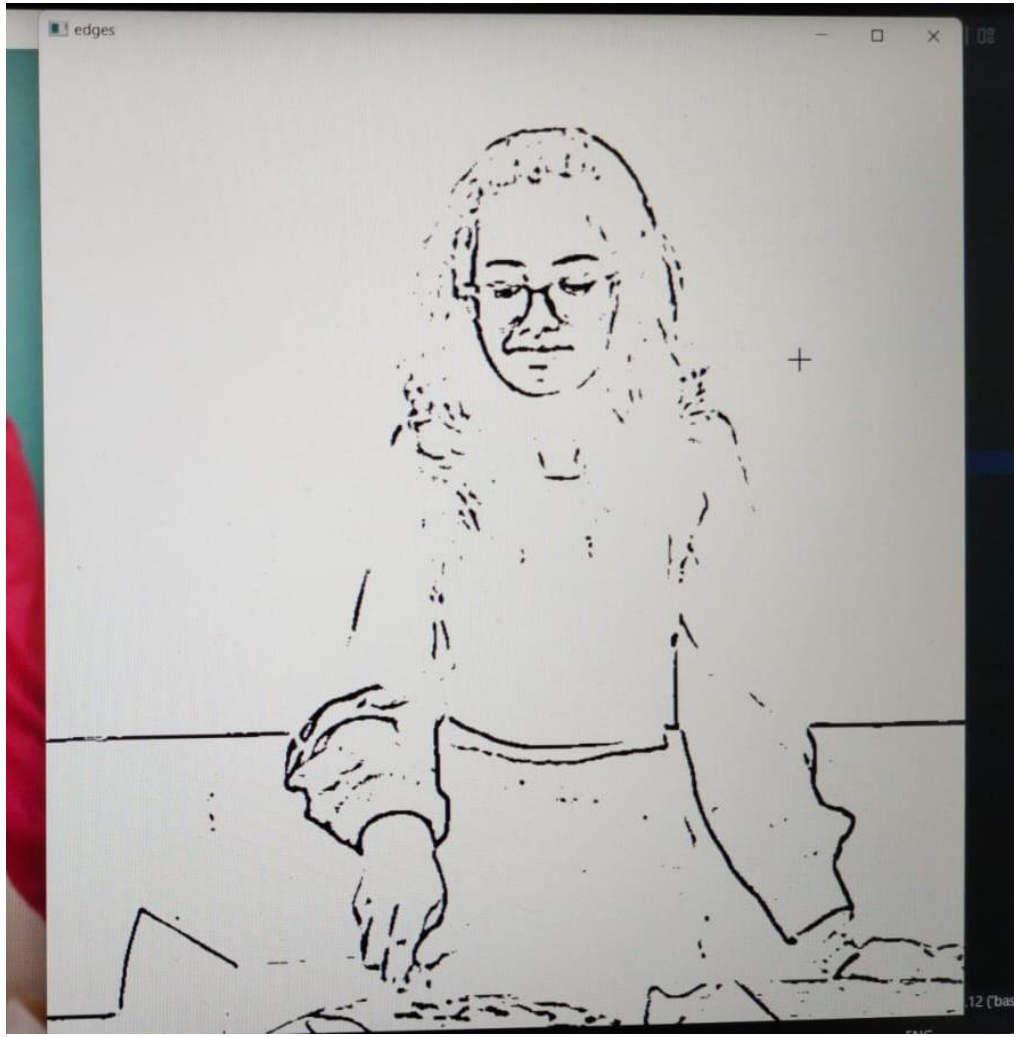


EDGE

- The Edge detection technique plays very important role in computer vision systems. Edges define the boundaries between different regions in an image, which helps in matching the pattern, segment, and recognize an object. In many applications the overall performance of the system depends on the proper detection of the edges such as Text Detection, Shape detection, Finger Print Recognition, Pattern Recognition etc..
- **In Code :-**

```
edges = cv2.adaptiveThreshold(grey, 255, cv2.ADAPTIVE_THRESH_MEAN_C,
cv2.THRESH_BINARY, 9, 9)
```

EDGE :

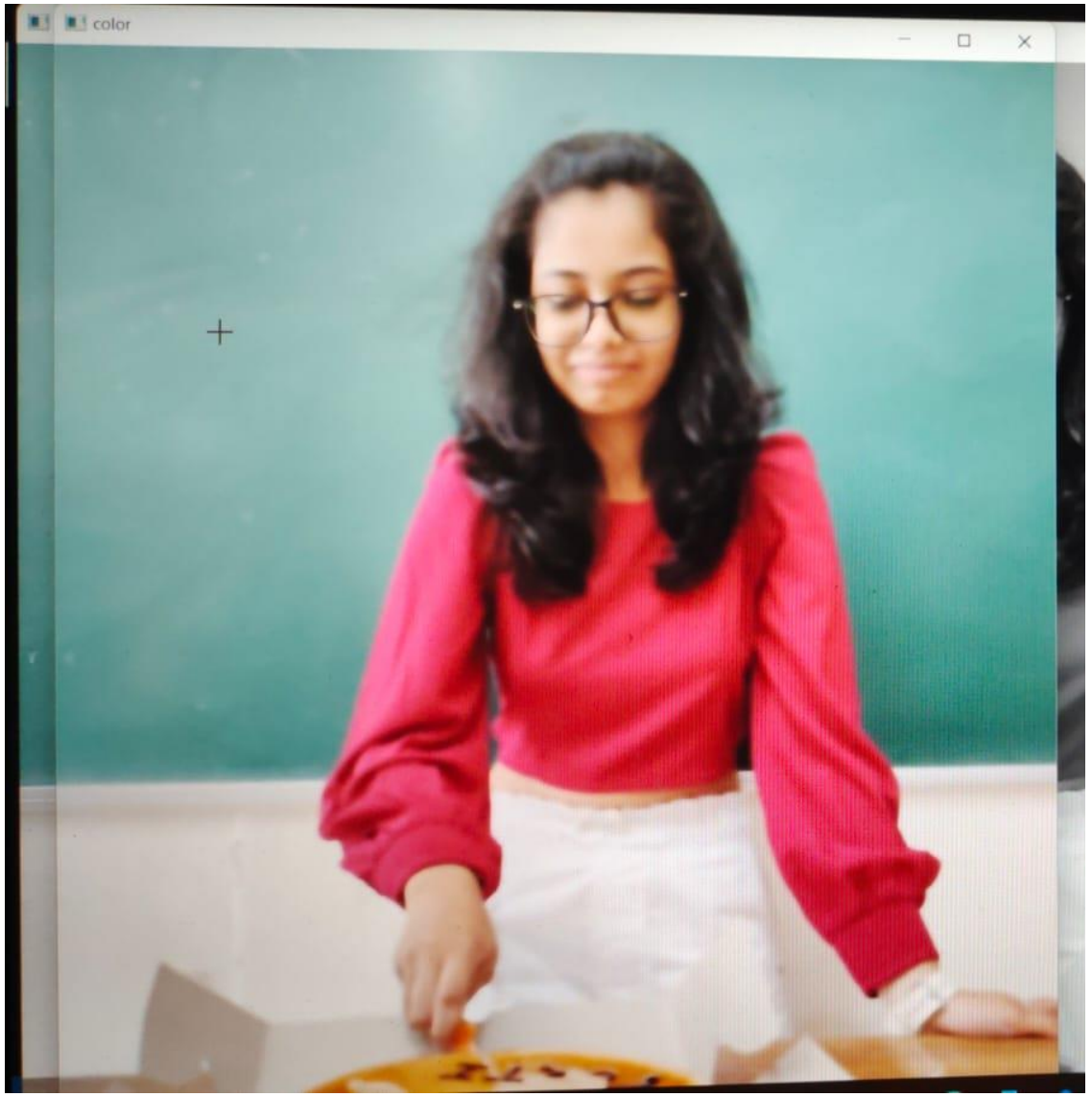


COLOR (bilateral filter) :

- A bilateral filter is used for smoothening images and reducing noise, while **preserving edges**. This article explains an approach using the averaging filter, while this article provides one using a median filter. However, these convolutions often result in a loss of important edge information, since they blur out everything, irrespective of it being noise or an edge. To counter this problem, the non-linear bilateral filter was introduced.
- **In Code :-**

```
color = cv2.bilateralFilter(img, 9, 250, 250)
```

COLOR :



PROCEDURE :

- The Real Image or the Original Image is taken by the module
- The real Image is converted to the Grey Scale.
- The Grey Scale Adjusts the Intensities of the Similar Composition and then we get 2 categories of the Image :- The 90-100% black part is called Edge or the Outline and the other part is called the Blur.
- The Blur is then worked upon by the Bilateral Filter and given the new name COLOR including the RGB colors (Reg, Green, Blue and compositions respectively.)
- The Color and Edge are then Overlapped to form the Cartoonised Image.

Disadvantage

- It won't work on the OLDER Photos. Since they had lower Resolution, the Blur used to get reflected along with the Reflection of the Light Source in the Camera (System). Causing it to remain Untraceable.

Eg. Observe the Forehead and the Neck Region of the Cartoonised Image.

Cartoon