**IMAGE PROCESSING**

* Image processing is by doing some operation with opencv python.
* By applying tkinter code I created window,frames and buttons to performing an image.
* Using opencv-python I can learn some operations and implement that operation through the Tkinter buttons.

**Step1:creating a window for application**

* Root.title() is a keyword for naming the window.
* Root.minsize() is a keyword for sizing the window

We can pass height and width inside it.

**Step2:creating a frame for displaying image**

* LabelFrame is a keyword which is used to design the frame with our own wish.
* Label is a widget that implements a display box where you can place.

**Step3:creating buttons and doing operation on it**

* **color to gray:**

This operation will convert the color image into grayscale image.

By using cv2.COLOR\_BGR2GRAY keyword we can convert the input image into grayscale image.

**Purpose:**

* RGB will take 24 bits for each color components(8-Bits of each).But we convert RGB into a grayscale only 8 bits is required to store a single pixel of the image.
* Grayscale image is much useful in variety of task like morphological operation and image segmentation.

**Result:**

Input: Output:

** **

* **Rotate\_180:**

This operation will convert the original image into 180 degree rotation.

By using cv2.ROTATE\_180 keyword we can show the image in 180 degree rotation.

**Result:**

Input: Output:

** **

* **Rotate\_90:**

This operation will convert the original image into rotating 90 degree counterclockwise.

By using cv2.ROTATE\_90\_COUNTERCLOCKWISE keyword we can show the image in 180 degree rotation.

**Result:**

Input: Output:

* **Blur:**

This operation show the output image in blur.

There are 4 types of blur,they’re

* Averaging
* Gaussian
* Median
* Bilateral

Here,I used Averaging technique to blur my input image.

Cv2.blur is a keyword which I used here to blur my image.we can pass the intensity of blurring inside that keyword.

**Purpose:**

* It helps in smoothing the images

For eg: In many cases police want to hide the face of the victim.In such case blurring is required.

**Result:**

Input: Output:

 ****

* **Re-Size:**

This operation shows the re-sized image when we passed the input image.we want to declare height and width in what size we want to resize the image.There are 3 types of keywords to resize.They,are

* cv2.INTER\_AREA:This is used when we need to shrink an image.
* cv2.INTER\_CUBIC:This is slow but efficient.
* cv2.INTER\_LINEAR:This is default interpolation technique in opencv.This is primarily used when zooming is required.

Here I used cv2.INTER\_AREA to resize my specific image.we can also use width,height to determine the image for output.

* .shape is keyword When working with **OpenCV Python**, images are stored in numpy ndarray. To get the image **shape** or size, use ndarray. **shape** to get the dimensions of the image.

**Purpose:**In Real life,the re-sizing is used in many ways.In some online application form the image should be in some certain size to upload.On that time

the re-sizing is used.

**Result:**

Input: Output:

 ****

* **Edge Detection:**

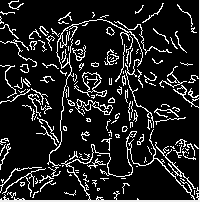
Edge detection is nothing but detecting the edges of input image.

* Here,I used cv2.canny keyword to detect the edges of the images.inside of cv2.canny we can pass the arguments like input,threshold1 and threshold2 to retrieving the output as edged image.

**Purpose:**It is used to find the boundary of the image(Object detection) and autonomous driving(self driving car).

**Result:**

Input: Output:

 ****

* **Threshold:**

Threshold is defined as process is sometimes described as separating an **image** into foreground values (black) and background values (white).

There are 2 types of threshold.They are,

* Simple Threshold-The threshold value is global i.e., for all pixels in the image.
* Adaptive Threshold-The threshold value is calculated for smaller region and therefore there are different threshold values for different region.

Here,I used simple Threshold.In that simple threshold there are some types with it.

* cv2.THRESH\_BINARY
* cv2.THRESH\_BINARY\_INV
* cv2.THRESH\_TRUNC
* cv2.THRESH\_TOZERO
* cv2.THRESH\_TOZERO\_INV

**Purpose:**The threshold process is mainly used to (image segmentation) separate an image into foreground(black) values and background(white) values for our better understanding to analyse the image.

**Result:**

Input: Output:

 ****

**Morphological operation:**

* Dilation
* Erosion

Real Time Use Case of Morphological operation:

* Number plate Identification.
* Handwritten digits,character recognition.
* Text extraction
* **Dilation:**

Dilation add pixel to an image.

Dilation increases the brightness of the image.

By using, cv2.dilate() keyword we can pass the matrix of the size which is required to dilate that image.

**Purpose:**dilation is to get some more knowledge on that input image.

**For eg:**In real life,the detectives Dilates the photos to see the smaller details and evidence for their research.

**Result:**

Input: Output:

 ****

* **Erosion:**

Its opposite to the dilation process.It removes the pixel from an image.It decreases the brightness of the image.

By using, cv2.erode() keyword we can pass the matrix of the size which is required to erode the image.

**Purpose:**The real time purpose which is more over similar to dilation process.

**Result:**

Input: Output:

 ****

* **Border:**

By using the keyword cv2.copyMakeBorder() we can make a border for an output image .Next

We want to pass top,bottom,right,left width’s of the border inside a parameter.finally,we want to give flag or values inside it.here,I used cv2.BORDER\_CONSTANT to creating a border for an image.

* Cv2.BORDER\_REFELECT
* CV2.BORDER\_REFELECT\_101
* CV2.BORDER\_REPLICATE

These are the similar flags or value which cv2.copyMakeBorder() contains with it.

**Purpose:** Adding border to the image is for better look and good visualization of an image.

**Result:**

Input: Output:

 ****

* **Crop:**

By slicing the array we can able to crop the image.on 1st we want to pass the start Y point & end Y point.Then start X point & end X point.

**Purpose:** The purpose of cropping is to cut the image in whatever the part user need really.

**Result:**

Input: Output:

* **Contours:**
* Contour is an operation used for joining all the points along the boundary points.
* For better accuracy 1st we want to convert the input image as threshold image.
* By using the keyword cv2.findcontours() we can make the output boundary line for an input image.
* Inside the parameter we want to pass the threshold image,then contour retrieval mode and finally contour approximation method.
* By using the keyword cv2.drawcontours() we can able to draw the contours along the boundary of the image.
* we can pass the parameter inside this keyword like input image,colors of contours & thickness of contours.

**Purpose:** Contour is tool for shape analysis & object detection.for eg:By using contour we can detect the object whether its a human or animals or machine.

**Result:**

Input:Output:

 ****

* **Histogram:**
* creating histogram provides visual presentation of data distribution.
* we want import matplotlib to perform this operation.
* 1st we want to split the input image using cv2.split() keyword.
* plt.hist() is a keyword used for determining the output as pyplot.
* .ravel() is a keyword used for returned array has same data type as the input array.
* at last we want to pass the maximum pixel & range.

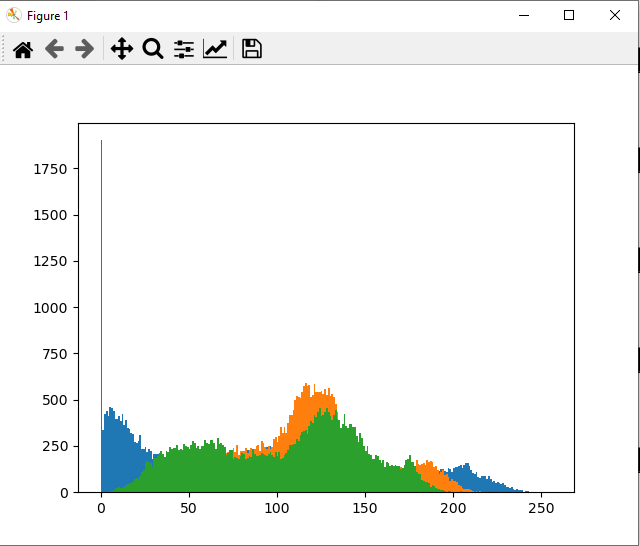
**Purpose:** we can predicting the properties of an image just by looking the details of my histograms.

**Result:**

Input:



Output:



* **Blob\_Detection:**
* Blob is abbreviated Binary large object.
* The dark connected regions in the images are blobs.
* For doing this operation,we want to create a detector 1st and pass the image to that detector next.
* Read the input image in grayscale.
* By using the keyword cv2.simpleblobdetector() create one detector for detection.
* np.array is used to for creating array values,matrix data structure.
* cv2.DRAW\_MATCHES\_FLAGS\_DEFAULT() is a flag which draws a blob as red circle and the size of the circle is correspond to the size of the blob.

**Purpose:**Blob detection aimed at detecting points or regions in the image that differ in properties like brightness or color compared to surrounding.

**Result:**

Input: Output:

 ****

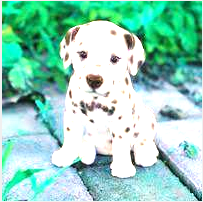
* **Contrast:**
* Increasing the contrast of the image by using function called cv2.addweighted().Inside this we can pass some parameters.They are,

1. Input image
2. Alpha-weight of the 1st element
3. Beta and gama-There is no additional image in my code so I declared beta and Gama as 0.

**Purpose**:contrast makes light areas lighter and dark area much darker.It will show the separation of darkened and brightened areas in the image.

**Result:**

Input: Output:

 ****

* **Upload Image:**

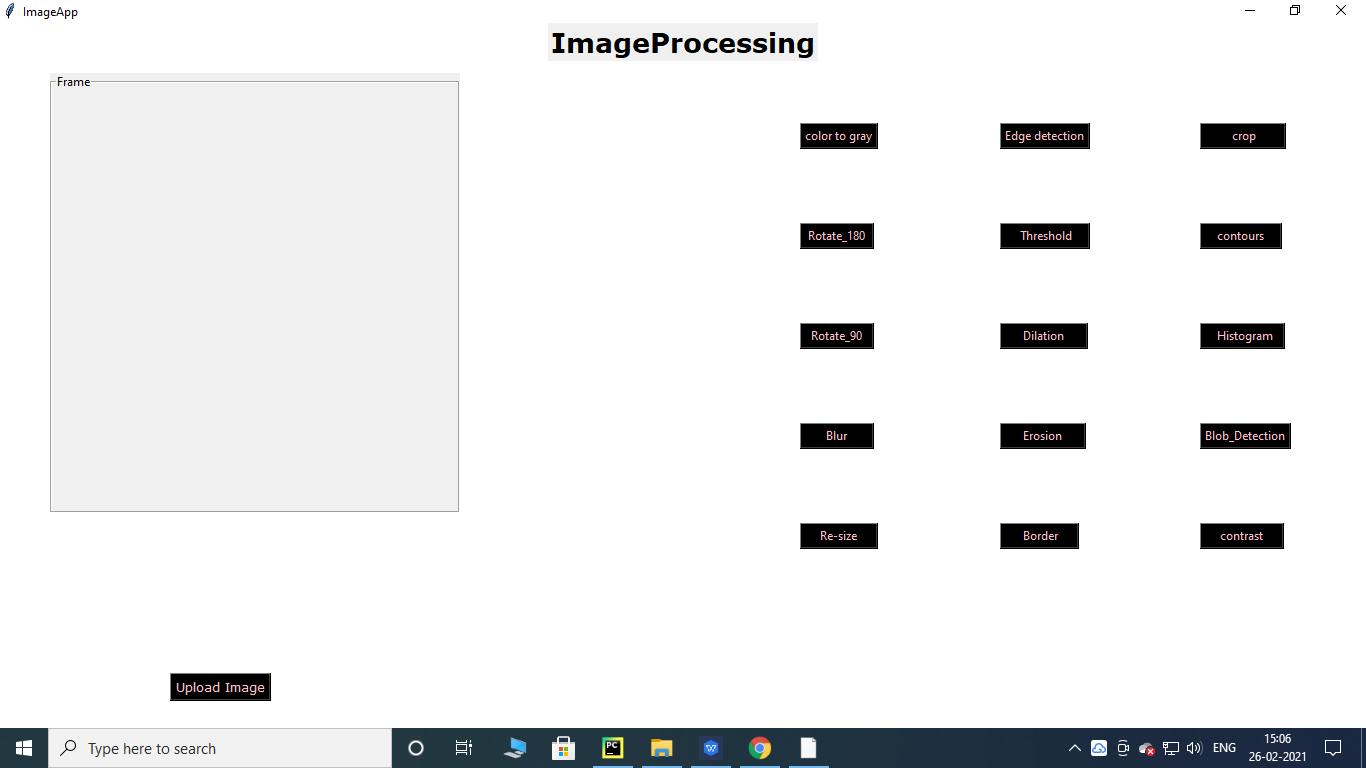
By using the filedialog.askopenfilaname() keyword we can access the image from our desired location.

* thumbnail() is a keyword used to compress the image with in a frame.
* Image.fromarray() is a keyword is used for creating a memory.we can pass the image as parameter.
* ImageTk.photoImage() is keyword is used to display the expected image.

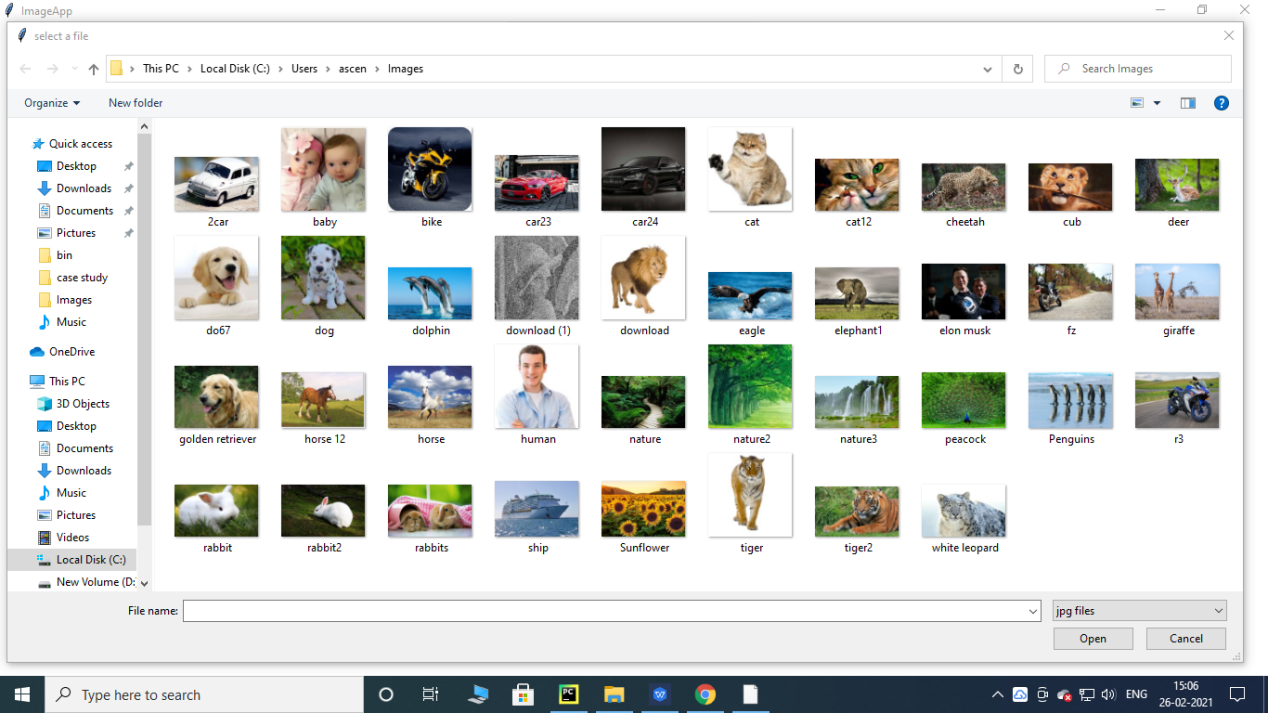
**Step4:**By defining the function I connect the Open-CV code and tkinter code to perform that operation,when button is been clicked.

**Result:**

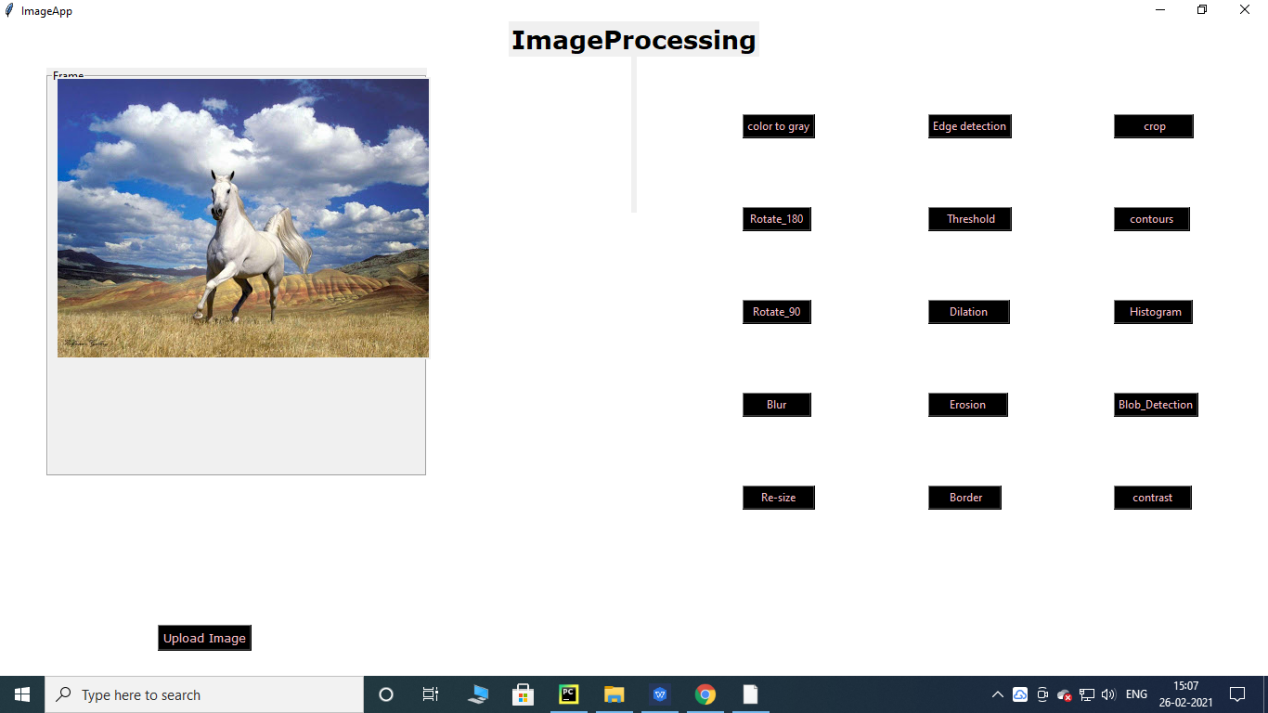
Output when run the application:



Output when click Upload Image:



Output when choose Images from path folder:



These are all the operation what I did in Image processing.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*