

31. Morphological operations based on OpenCV using Opening technique.

**PROGRAM:**

```
import cv2
import numpy as np
img = cv2.imread(r"C:\Users\91824\OneDrive\meliodas.jpg", cv2.IMREAD_GRAYSCALE)
kernel = np.ones((5,5), np.uint8)
opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
cv2.imshow("Original", img)
cv2.imshow("opening", opening)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

**INPUT :**



**OUTPUT:**



32. Morphological operations based on OpenCV using Closing technique.

**PROGRAM:**

```
import cv2
import numpy as np
img = cv2.imread(r"C:\Users\91824\OneDrive\ban.jpg", cv2.IMREAD_GRAYSCALE)
kernel = np.ones((5,5), np.uint8)
closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
cv2.imshow("Original", img)
cv2.imshow("Closing", closing)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

**INPUT:**



**OUTPUT:**



### 33. Morphological operations based on OpenCV using Morphological Gradient technique

#### PROGRAM:

```
import cv2  
  
import numpy as np  
  
img = cv2.imread(r"C:\Users\91824\OneDrive\sin.jpg", cv2.IMREAD_GRAYSCALE)  
  
kernel = np.ones((5,5), np.uint8)  
  
grad = cv2.morphologyEx(img, cv2.MORPH_GRADIENT, kernel)  
  
cv2.imshow("Original", img)  
  
cv2.imshow("Gradient", grad)  
  
cv2.waitKey
```

#### INPUT :



#### OUTPUT :



34. Morphological operations based on OpenCV using Top hat technique.

**PROGRAM:**

```
import cv2
import numpy as np
img = cv2.imread(r"C:\Users\91824\OneDrive\hawk.jpg",cv2.IMREAD_GRAYSCALE)
kernel = np.ones((5,5), np.uint8)
tophat = cv2.morphologyEx(img, cv2.MORPH_TOPHAT, kernel)
cv2.imshow("Original", img)
cv2.imshow("Top Hat", tophat)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

**INPUT:**



**OUTPUT:**



35. Morphological operations based on OpenCV using Black hat technique.

**PROGRAM:**

```
import cv2  
  
import numpy as np  
  
img = cv2.imread(r"C:\Users\91824\OneDrive\king.jpg", cv2.IMREAD_GRAYSCALE)  
  
kernel = np.ones((5,5), np.uint8)  
  
blackhat = cv2.morphologyEx(img, cv2.MORPH_BLACKHAT, kernel)  
  
cv2.imshow("Original", img)  
  
cv2.imshow("Black Hat", blackhat)  
  
cv2.waitKey(0)  
  
cv2.destroyAllWindows
```

**INPUT :**



**OUTPUT :**



36. Recognise watch from the given image by general Object recognition using OpenCV.

**PROGRAM:**

```
import cv2

watch_cascade = cv2.CascadeClassifier("C:/Users/divya/OneDrive/Documents/COMPUTER
VISION/watch-cascade.xml")

img = cv2.imread("C:/Users/divya/OneDrive/Documents/COMPUTER VISION/COMPUTER
VISION/watch.jpg")

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

watches = watch_cascade.detectMultiScale(gray, scaleFactor=1.2, minNeighbors=5)

for (x, y, w, h) in watches:
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

cv2.imshow('Watches Detected', img)

cv2.waitKey(0)

cv2.destroyAllWindows()
```

**OUTPUT:**

### 37. Using Opencv play Video in Reverse mode.

#### PROGRAM:

```
import cv2

cap =
cv2.VideoCapture(r"C:\Users\91824\Videos\_@Chepparayey_Wednesday_2022_S01E08_720p_NF_
WEBRip_x265_10bit_Telugu.mkv")

total_frames = cap.get(cv2.CAP_PROP_FRAME_COUNT)

current_frame = total_frames - 1

while current_frame >= 0:

    cap.set(cv2.CAP_PROP_POS_FRAMES, current_frame)

    ret, frame = cap.read()

    if not ret:

        break

    cv2.imshow('Video in Reverse', frame)

    if cv2.waitKey(25) & 0xFF == ord('q'):

        break

    current_frame -= 1

cap.release()

cv2.destroyAllWindows()
```

#### OUTPUT :



### 38. Face Detection using Opencv

#### **PROGRAM:**

```
import cv2

img = cv2.imread("C:/Users/koppo/Downloads/20101123131216-1_0.jpg")

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

face_cascade =

cv2.CascadeClassifier("C:/Users/koppo/Downloads/haarcascade_frontalface_default.xml")

faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)

for (x, y, w, h) in faces:

    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

cv2.imshow('Faces Detected', img)

cv2.waitKey(0)

cv2.destroyAllWindows()
```

#### **OUTPUT :**



### 39. Vehicle Detection in a Video frame using OpenCV

#### **PROGRAM:**

```
import cv2

car_cascade = cv2.CascadeClassifier(r"C:/Users/divya/OneDrive/Documents/COMPUTER
VISION/cars.xml")

cap = cv2.VideoCapture("C:/Users/divya/Downloads/car.mp4")

while True:

    ret, frame = cap.read()

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    cars = car_cascade.detectMultiScale(gray, 1.1, 1)

    for (x,y,w,h) in cars:

        cv2.rectangle(frame, (x,y), (x+w,y+h), (0,0,255), 2)

    cv2.imshow('frame', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):

        break

cap.release()

cv2.destroyAllWindows()
```

#### **OUTPUT :**

#### 40. Draw Rectangular shape and extract objects

##### PROGRAM:

```
import cv2  
  
img = cv2.imread("C:/Users/divya/OneDrive/Documents/COMPUTER VISION/40.jpg")  
  
x, y = 100, 100  
  
width, height = 200, 150  
  
roi = img[y:y+height, x:x+width]  
  
cv2.imshow('ROI', roi)  
  
cv2.waitKey(0)  
  
cv2.destroyAllWindows()
```

##### INPUT :

66GE E



##### OUTPUT :

