

Programming for Geoinformatics _LAB_3

September 8, 2024

NAME : ASHWIN E

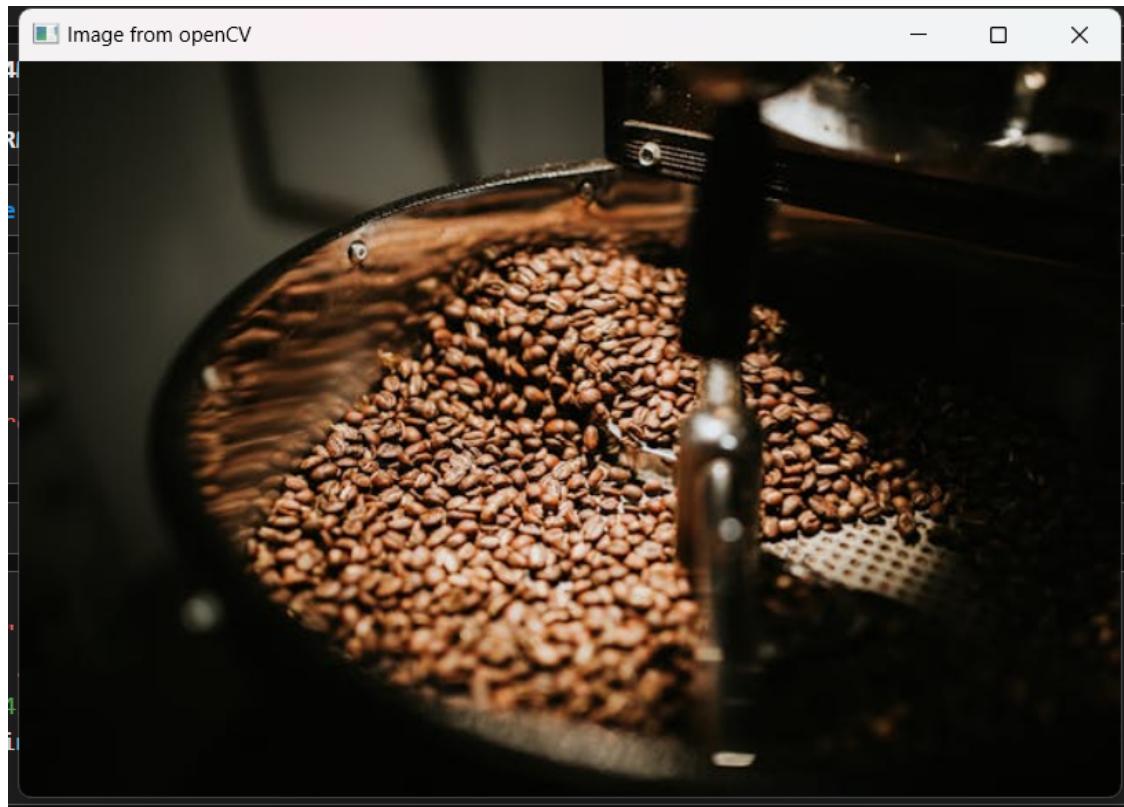
STUDENT ID : SC24M136

BRANCH : GEOINFORMATICS

1 LAB 3 Basic Image Processing in Python

1.1 PROGRAM 1

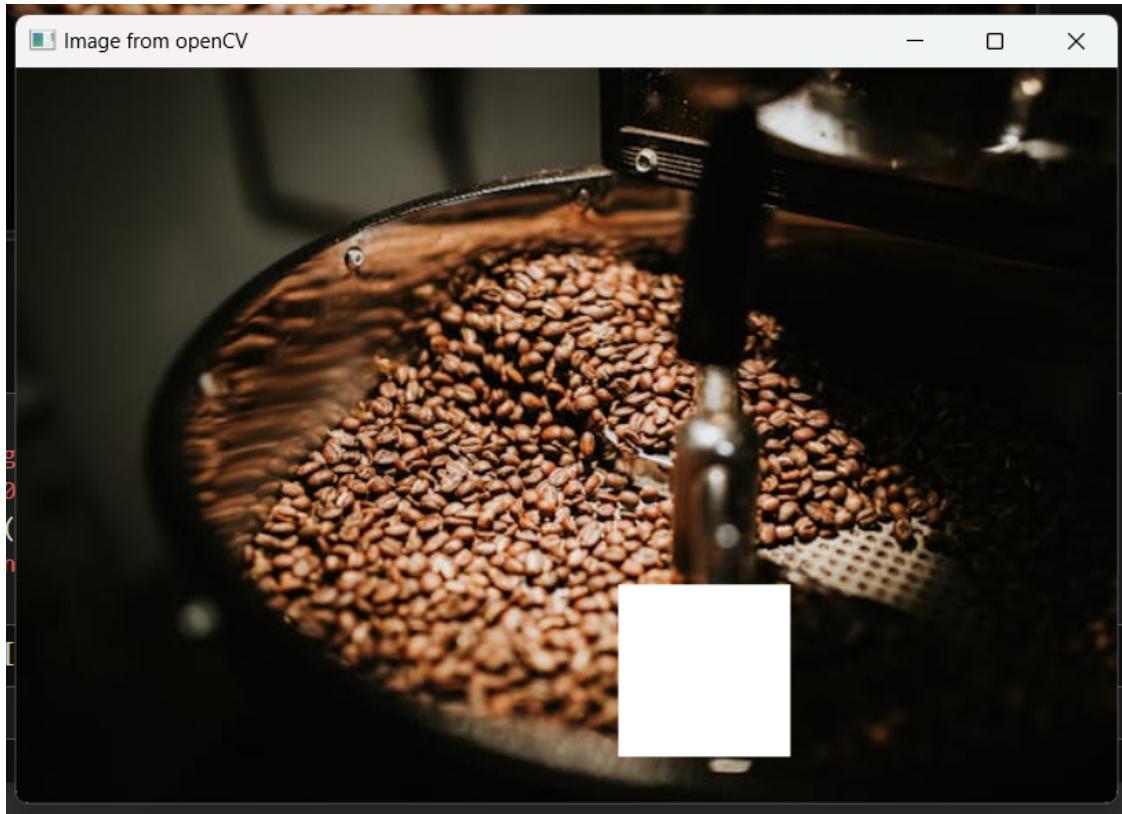
```
[ ]: import cv2 as cv
      image = cv.imread("img.jpg")
      cv.imshow('Image from openCV',image)
      cv.waitKey(0)
```



1.2 PROGRAM 2

```
[ ]: import cv2 as cv
image = cv.imread("img.jpg")
print("Pixel Value at (300,350)",image[300,350])
image[300:400,350:450] = (255,255,255)
cv.imshow('Image from openCV',image)
cv.waitKey(0)
```

Pixel Value at (300,350) [21 46 116]



1.3 PROGRAM 3

```
[1]: import cv2 as cv
image = cv.imread("img.jpg")
print("Shape of the image",image.shape)
print("Data Type of the image",image.dtype)
print("Size of the image",image.size)
```

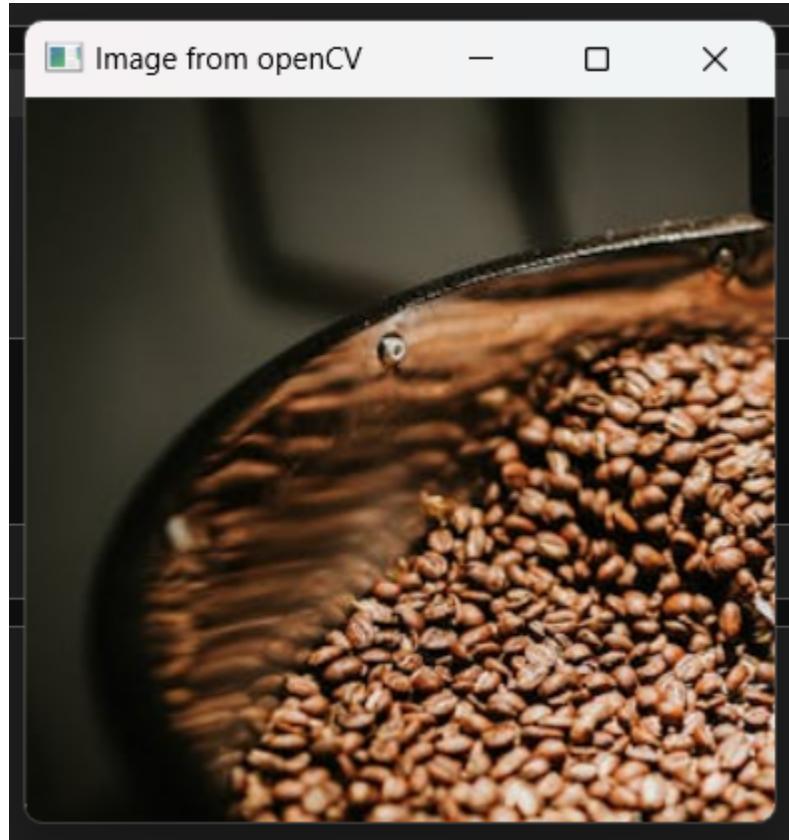
Shape of the image (427, 640, 3)

Data Type of the image uint8

Size of the image 819840

1.4 PROGRAM 4

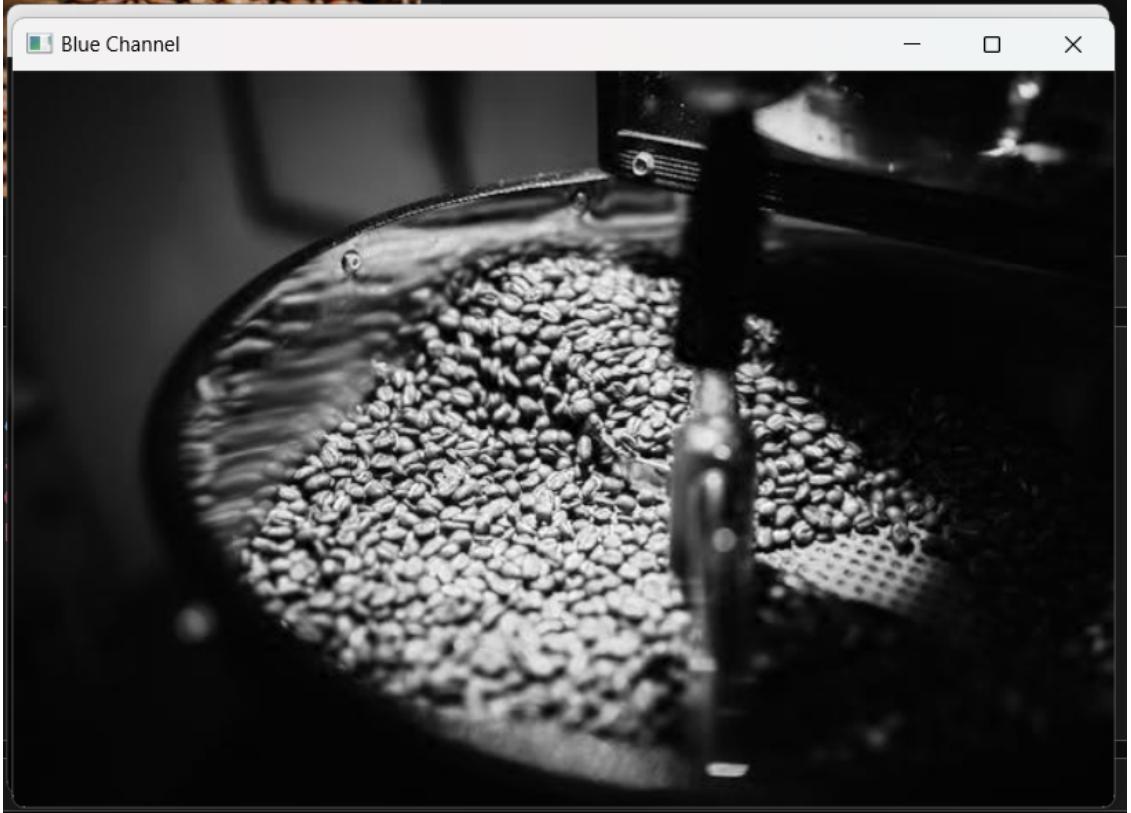
```
[ ]: import cv2 as cv
image = cv.imread("img.jpg")
region = image[10:300,50:350]
cv.imshow('Image from openCV',region)
cv.waitKey(0)
```

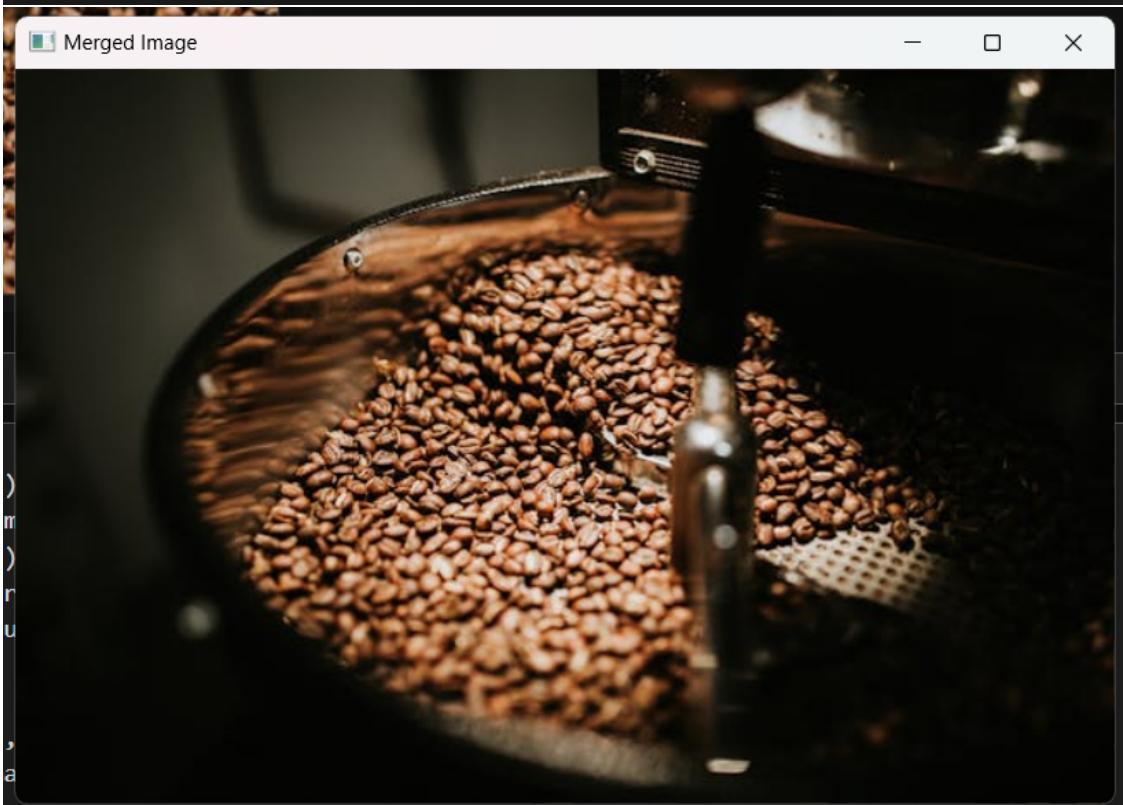
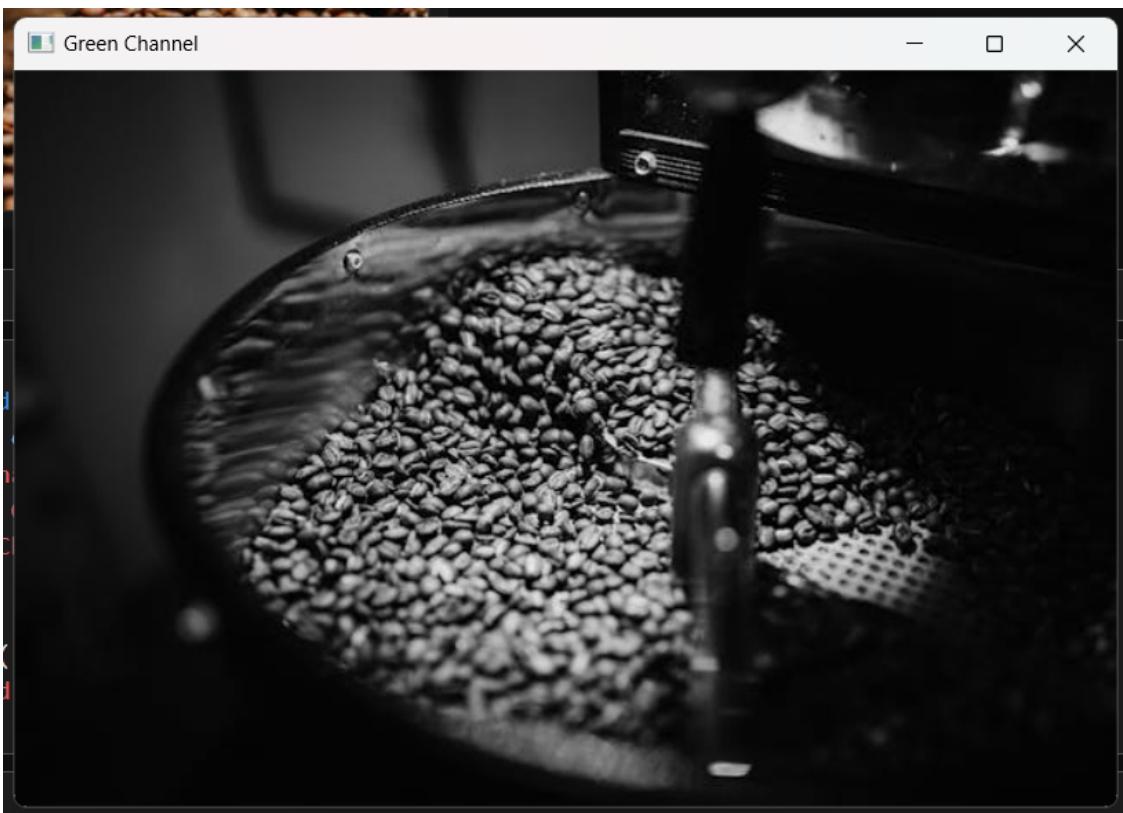


1.5 PROGRAM 5

```
[ ]: import cv2 as cv
image = cv.imread("img.jpg")
red,green,blue = cv.split(image)
cv.imshow('Red Channel',red)
cv.imshow('Green Channel',green)
cv.imshow('Blue Channel',blue)
cv.waitKey(0)

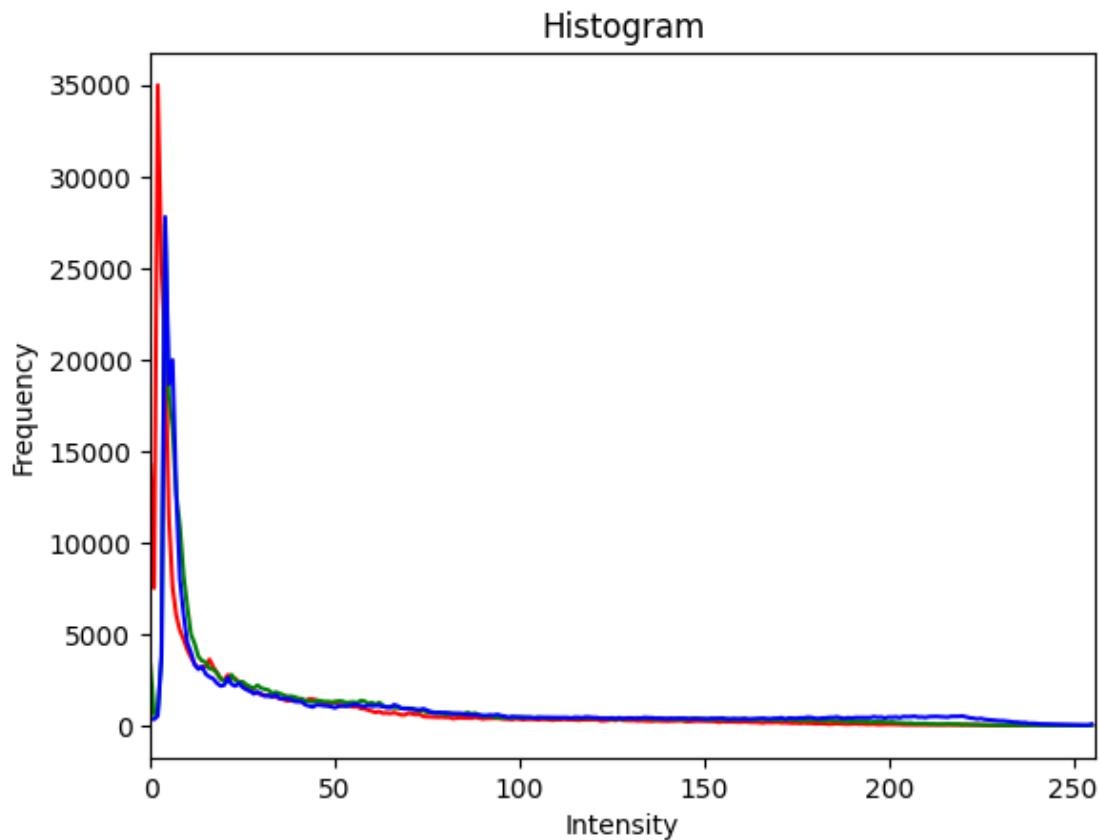
image = cv.merge([red,green,blue])
cv.imshow('Merged Image',image)
cv.waitKey(0)
```





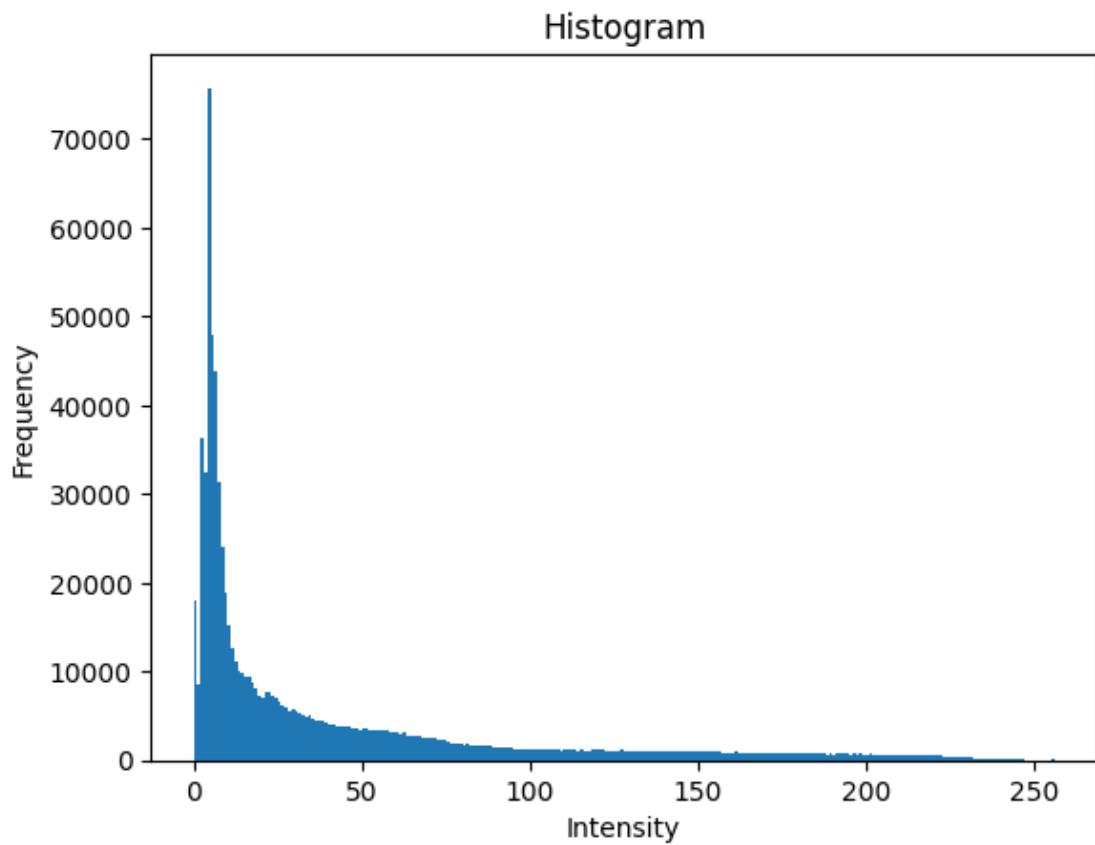
1.6 PROGRAM 6

```
[3]: import matplotlib.pyplot as graph
import cv2 as cv
import numpy as np
image = cv.imread("img.jpg")
red,green,blue = cv.split(image)
colors = ('red', 'green', 'blue')
for index,colo in enumerate(colors):
    histogram = cv.calcHist([image],[index],None,[256],[0,256])
    graph.plot(histogram,color = colo)
    graph.xlim([0,256])
graph.title("Histogram")
graph.xlabel("Intensity")
graph.ylabel("Frequency")
graph.show()
```



1.7 PROGRAM 7

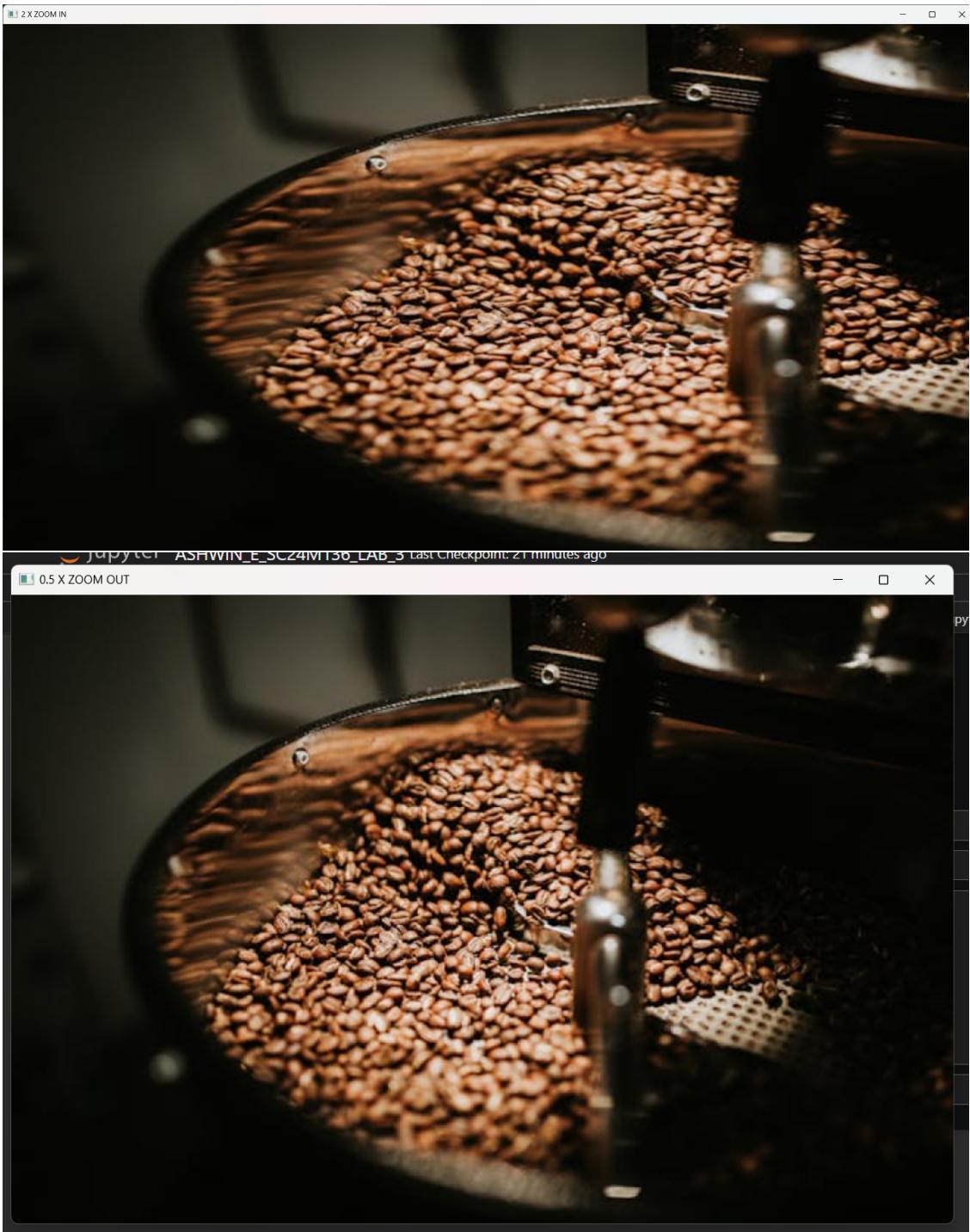
```
[9]: import cv2 as cv
import matplotlib.pyplot as graph
image = cv.imread("img.jpg")
grayscale_image = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
cv.imshow('Grayscale', grayscale_image)
cv.waitKey(0)
graph.hist(image.ravel(), 256, [0, 256])
graph.title("Histogram")
graph.xlabel("Intensity")
graph.ylabel("Frequency")
graph.show()
```





1.8 PROGRAM 8

```
[11]: import cv2 as cv
image = cv.imread("img.jpg")
z_in = cv.resize(image,None,fx=3.0,fy=2.0,interpolation=cv.INTER_LINEAR)
out = cv.resize(image,None,fx=1.5,fy=1.5,interpolation=cv.INTER_AREA)
cv.imshow('2 X ZOOM IN', z_in)
cv.imshow('0.5 X ZOOM OUT', out)
cv.waitKey(0)
cv.destroyAllWindows()
```

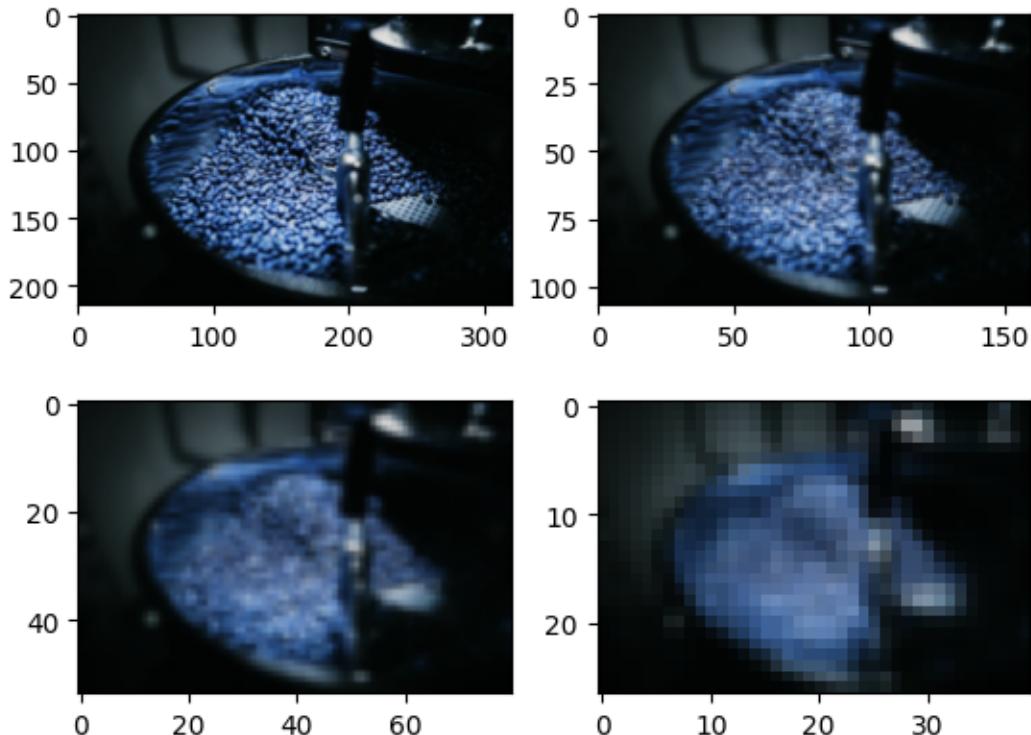


1.9 PROGRAM 9

```
[14]: import cv2 as cv
import matplotlib.pyplot as graph
image = cv.imread("img.jpg")
layer = image.copy()
for i in range(4):
```

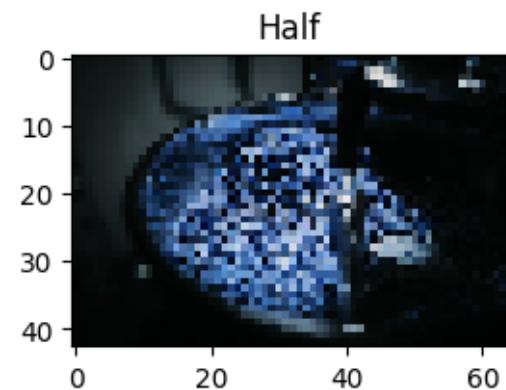
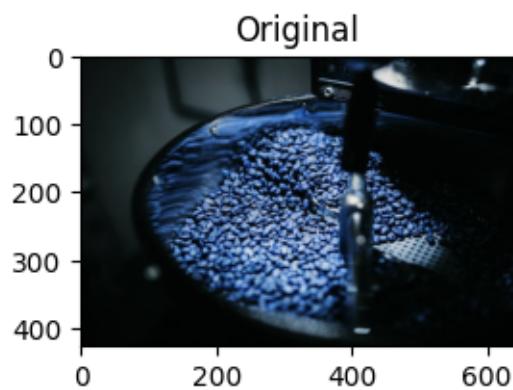
```
graph.subplot(2, 2, i + 1)
layer = cv.pyrDown(layer)
graph.imshow(layer)
cv.waitKey(0)
```

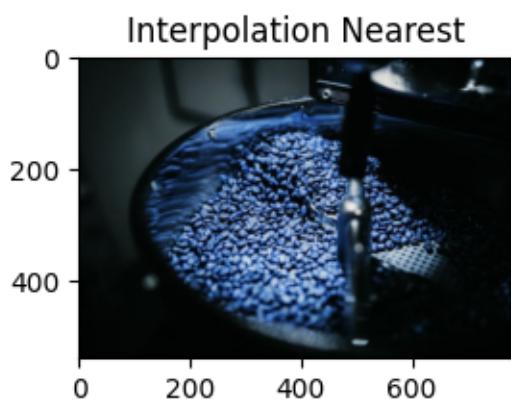
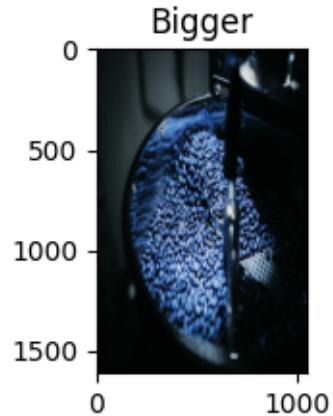
[14]: -1



1.10 PROGRAM 10

```
[17]: import cv2 as cv
import matplotlib.pyplot as graph
image = cv.imread("img.jpg")
half = cv.resize(image, (0, 0), fx = 0.1, fy = 0.1)
bigger = cv.resize(image, (1050, 1610))
stretch_near = cv.resize(image, (780, 540), interpolation = cv.INTER_LINEAR)
Titles =["Original", "Half", "Bigger", "Interpolation Nearest"]
images =[image, half, bigger, stretch_near]
count = 4
for i in range(count):
    graph.subplot(2, 2, i + 1)
    graph.title(Titles[i])
    graph.imshow(images[i])
    graph.show()
```





1.11 INFERENCES

- 2 Learnt how to import and find the attributes of an image.
- 3 Learnt how to use image manipulation operations like converting to other formats, converting to different channels, greyscale and bands in OpenCV.
- 4 Plotted pixel density histograms using matplotlib module.