

FOUNDATIONAL MATHEMATICS – MINI PROJECT

SANIYA SALUNKE

SYBSC.CS

4542

COLOR DETECTION

ACTUAL IMAGE USED:



CODE:

```
!pip install opencv-python  
  
import cv2  
import numpy as np  
import matplotlib.pyplot as plt
```

```
img = cv2.imread('C:\jupyternotebook\colours.jpg')
```

```
plt.figure(figsize=(20,8))  
plt.imshow(img)
```



CONVERTING IT INTO RGB IMAGE:

```
[18]: grid_RGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

```
[19]: plt.figure(figsize=(20,8))  
      plt.imshow(grid_RGB)
```



EXTRACTIING THE AREA OR A BOX WE WANT, USING MASK

DETECTING COLOUR YELLOW :

CODE:-

```
[32]: grid_HSV = cv2.cvtColor(grid_RGB, cv2.COLOR_RGB2HSV)

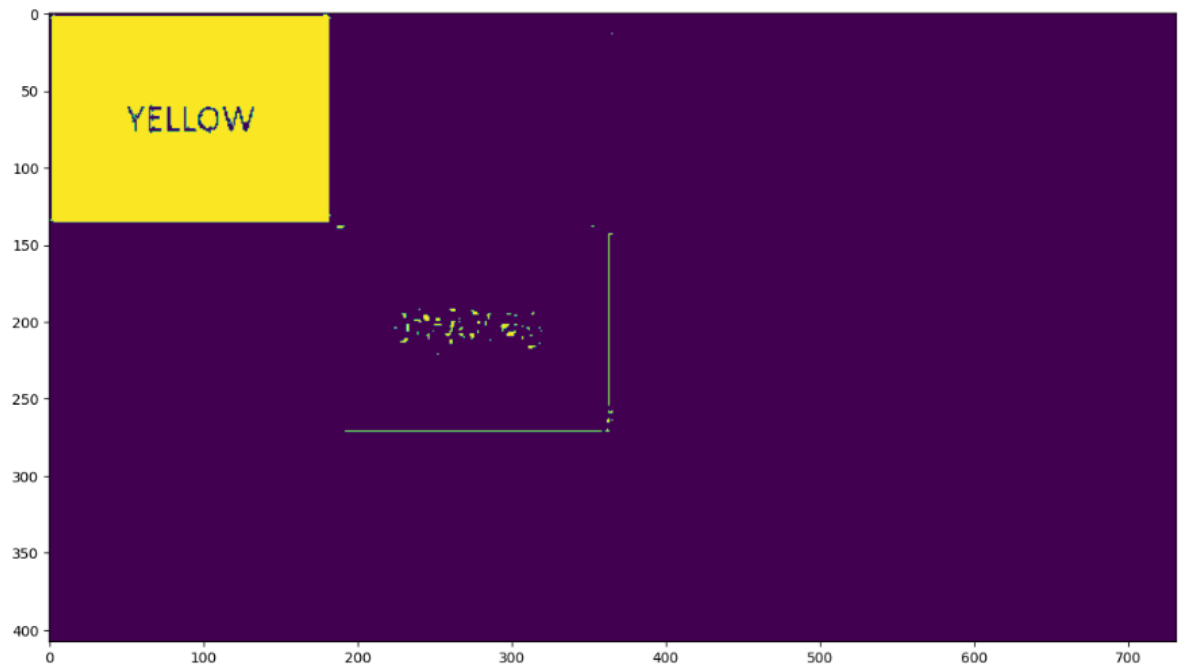
[46]: lower = np.array([25, 150, 50])
      upper = np.array([35, 255, 255])

[47]: mask = cv2.inRange(grid_HSV, lower, upper)

[48]: plt.figure(figsize=(20,8))
      plt.imshow(mask)
```

CREATED MASK ON YELLOW COLOUR

```
[48]: <matplotlib.image.AxesImage at 0x1478024b1d0>
```



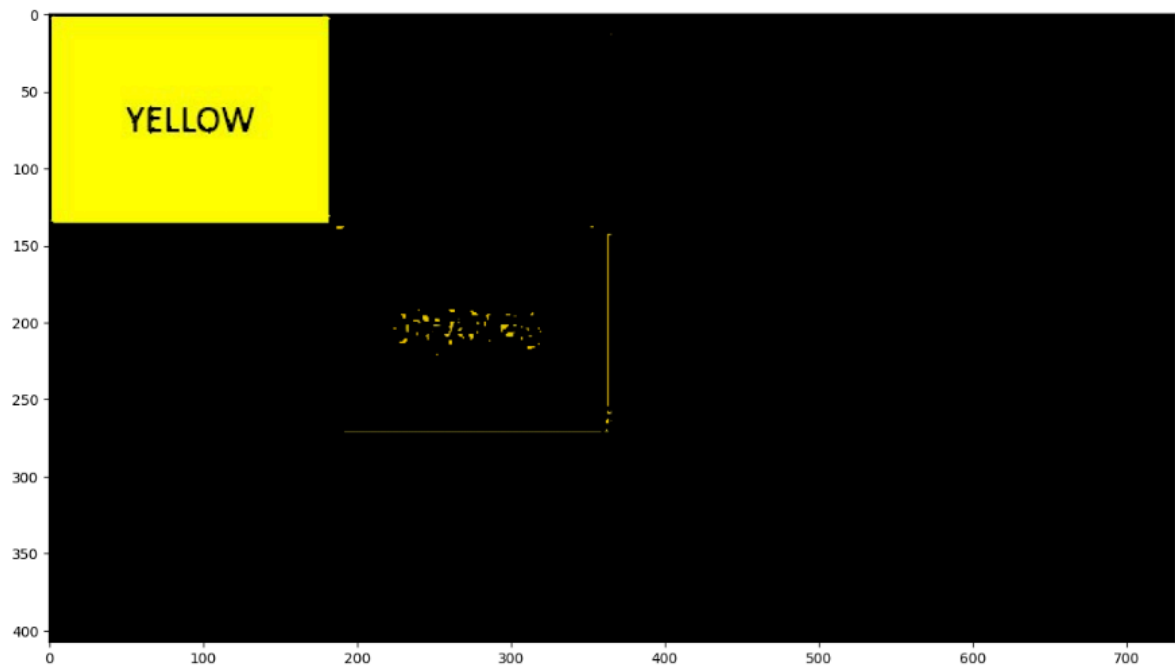
AFTER BITWISE OPERATION:

```
[49]: res = cv2.bitwise_and(grid_RGB, grid_RGB, mask=mask)
```

```
[50]: plt.figure(figsize=(20,8))  
plt.imshow(res)
```

```
[50]: <matplotlib.image.AxesImage at 0x147f994f5f0>
```

```
[50]: <matplotlib.image.AxesImage at 0x147f994f5f0>
```



DETECTING DARK BLUE COLOR:-

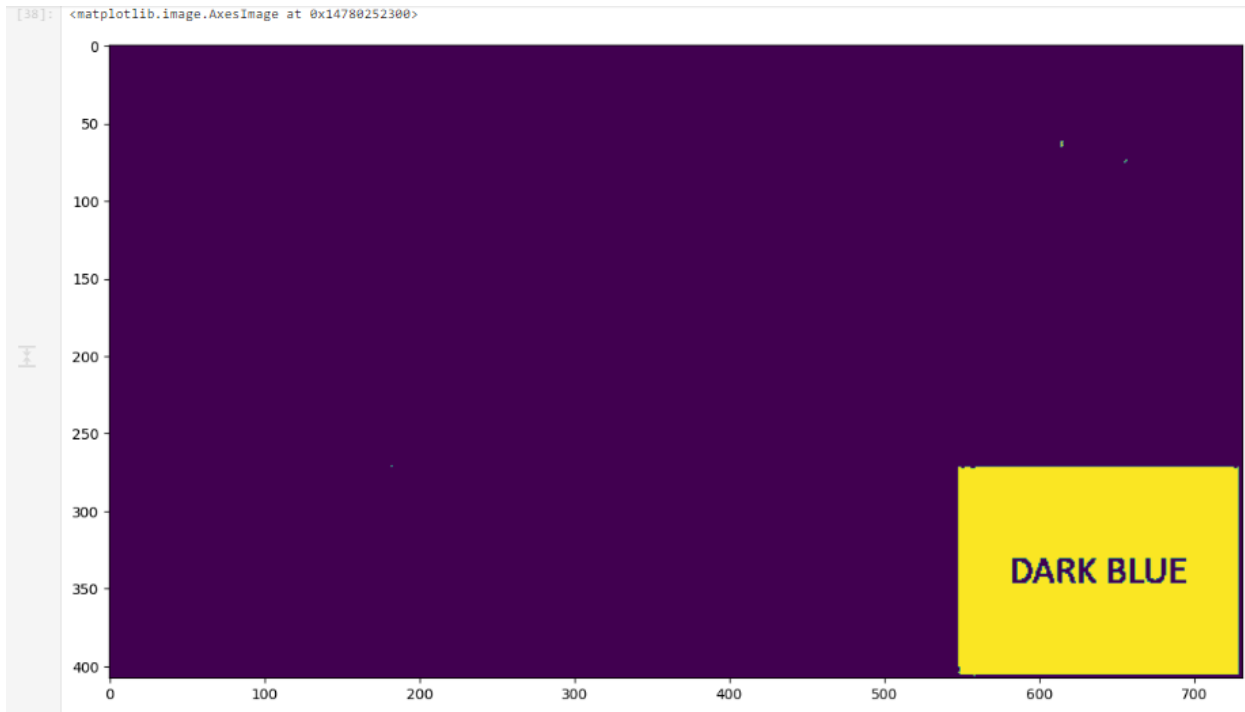
```
[32]: grid_HSV = cv2.cvtColor(grid_RGB, cv2.COLOR_RGB2HSV)
```

```
[51]: lower = np.array([115, 150, 50])  
upper = np.array([125, 255, 255])
```

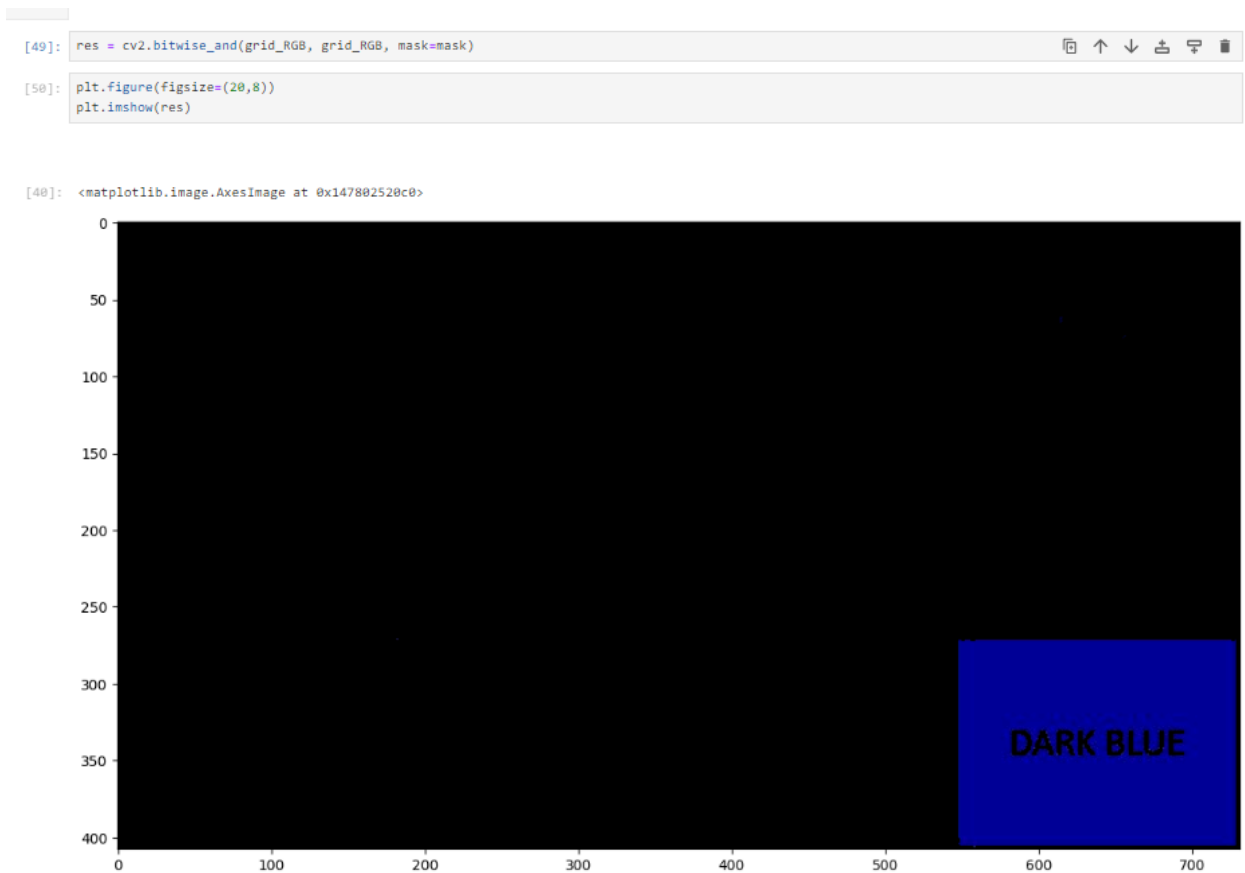
```
[52]: mask = cv2.inRange(grid_HSV, lower, upper)
```

```
[53]: plt.figure(figsize=(20,8))  
plt.imshow(mask)
```

CREATED MASK ON TOP OF DARK BLUE COLOUR



AFTER RUNNING BITWISE OPERATOR



DETECTING IMAGE

IMAGE USED:



DETECTING IMAGE USING CV:

```
!pip install opencv-python  
  
import cv2  
import numpy as np  
import matplotlib.pyplot as plt
```

```
[11]: img = cv2.imread('C:\jupyternotebook\mountains.jpg')
```

```
[12]: plt.figure(figsize=(20,8))  
      plt.imshow(img)
```



USING RGB

```
[13]: grid_RGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

```
[14]: plt.figure(figsize=(20,8))  
plt.imshow(grid_RGB)
```



```
[14]: <matplotlib.image.AxesImage at 0x1368c303740>
```



DETECTING GREEN COLOUR GRASS WITH MASK

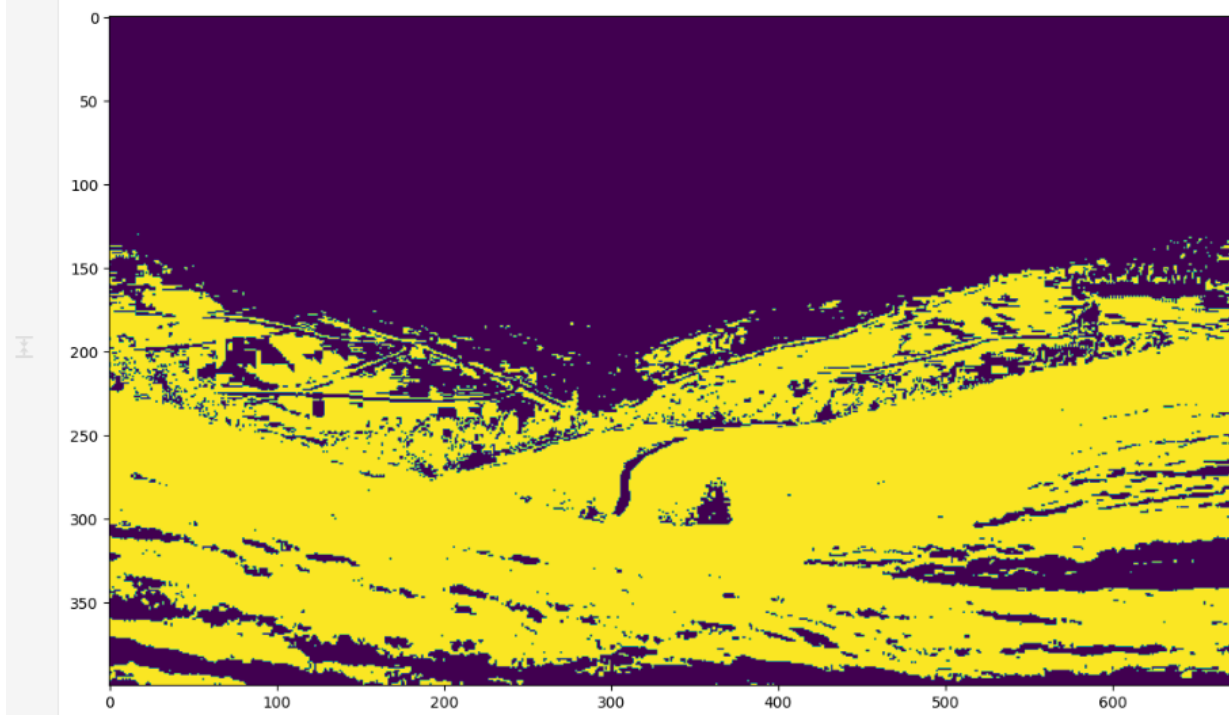
```
[21]: grid_HSV = cv2.cvtColor(grid_RGB, cv2.COLOR_RGB2HSV)
```

```
[22]: lower = np.array([35, 150, 50])  
upper = np.array([75, 255, 255])
```

```
[23]: mask = cv2.inRange(grid_HSV, lower, upper)
```

```
[24]: plt.figure(figsize=(20,8))  
plt.imshow(mask)
```

```
[24]: <matplotlib.image.AxesImage at 0x1368d7fbd10>
```

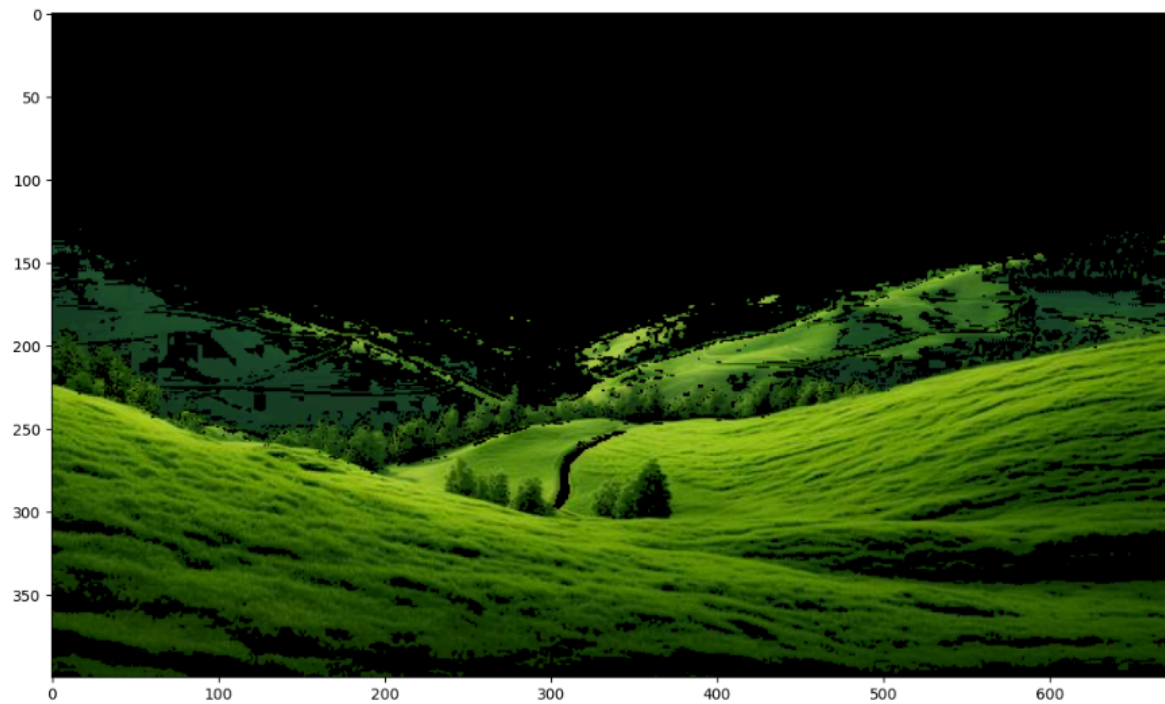


BITWISE OPERATION

```
[25]: res = cv2.bitwise_and(grid_RGB, grid_RGB, mask=mask)
```

```
[26]: plt.figure(figsize=(20,8))  
plt.imshow(res)
```

```
[26]: <matplotlib.image.AxesImage at 0x1368d863710>
```



DETECTING LIGHT BLUE COLOUR (SKY) IN IMAGE:

MASK

```
[31]: grid_HSV = cv2.cvtColor(grid_RGB, cv2.COLOR_RGB2HSV)
```

```
[32]: lower = np.array([95, 150, 0])  
upper = np.array([110, 255, 255])
```

```
[33]: mask = cv2.inRange(grid_HSV, lower, upper)
```

```
[34]: plt.figure(figsize=(20,8))  
plt.imshow(mask)
```

```
[34]: <matplotlib.image.AxesImage at 0x136908484d0>
```



BITWISE

```
[36]: res = cv2.bitwise_and(grid_RGB, grid_RGB, mask=mask)
```

```
[37]: plt.figure(figsize=(20,8))  
plt.imshow(res)
```

```
[37]: <matplotlib.image.AxesImage at 0x1369086dfd0>
```

