```python
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

```python
from google.colab import drive
drive.mount('/content/gdrive')
linken = "gdirve/My Drive/TTH_Digital_Image_Processing/"
```

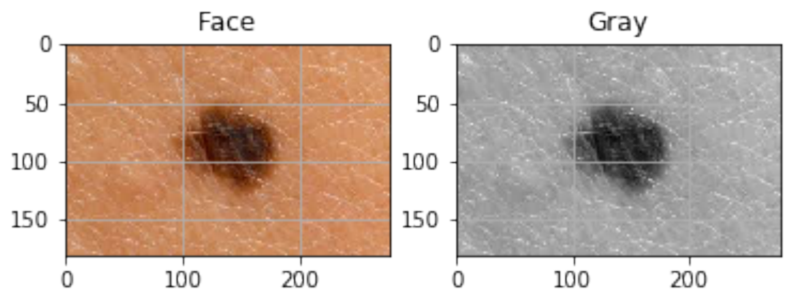
```
Drive already mounted at /content/gdrive; to attempt to forcibly remount,
```

```python
import os
linken = "gdrive/My Drive/TTH_Digital_Image_Processing/"
print(os.path.isdir(linken))
print(os.path.isfile(linken + "Dataset_05/Melanoma 01.jpg"))
print(os.path.isfile(linken + "Dataset_05/Melanoma 02.jpg"))
print(os.path.isfile(linken + "Dataset_05/Melanoma 03.jpg"))
```

```
True
True
True
True
```

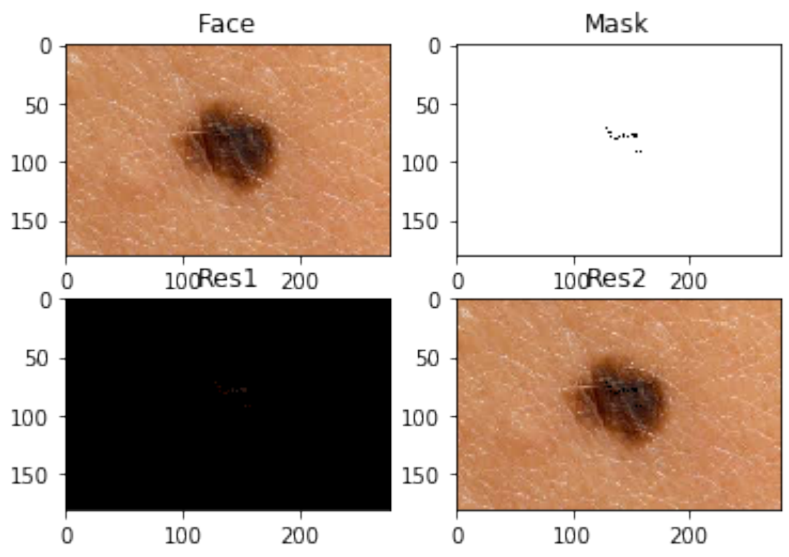
```python
import cv2
from matplotlib import pyplot as plt
from skimage.color import rgb2gray
from pylab import imread


def imshows(ImageData, LableData, rows, columns, gridType = True):
  ImageArray = list(ImageData)
  LableArray = list(LableData)

  from matplotlib import pyplot as plt
  fig = plt.figure()
  for i in range (1,rows*columns +1):
    fig.add_subplot(rows, columns, i)
    image = ImageArray[i -1]
    if(len(image.shape)<3):
      plt.imshow(image,plt.cm.gray)
      plt.grid(gridType)
    else:
      plt.imshow(image)
      plt.grid(gridType)
    plt.title(LableArray[i - 1])
  plt.show()
```

```
image = imread(linken + "Dataset_05/Melanoma 01.jpg")
image_gray = rgb2gray(image)

imshows([image, image_gray],["Face","Gray"],1,2,gridType = True)

print("Intensity Min: " + str(image_gray.min()))
print("Intensity Max: " + str(image_gray.max()))
```



```
Intensity Min: 0.0225
Intensity Max: 0.890149411764706
```

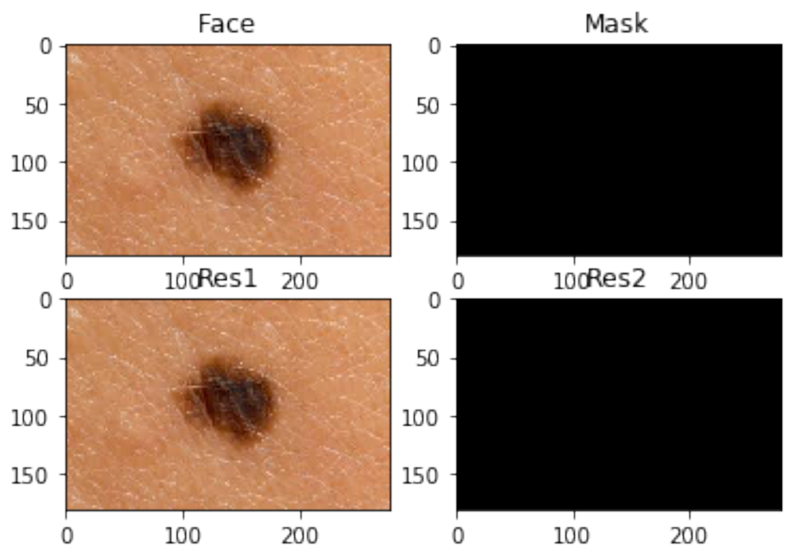
```
import cv2
import numpy as np

image = imread(linken + "Dataset_05/Melanoma 01.jpg")

hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

lower_blue = np.array([100,50,50])
upper_blue = np.array([130,255,255])

mask = cv2.inRange(hsv, lower_blue, upper_blue)

res1 = cv2.bitwise_and(image, image, mask = 255 - mask)
res2 = cv2.bitwise_and(image, image, mask = mask)

imshows([image,mask, res1, res2],["Face","Mask","Res1","Res2"],2,2,gridType = Fa
```



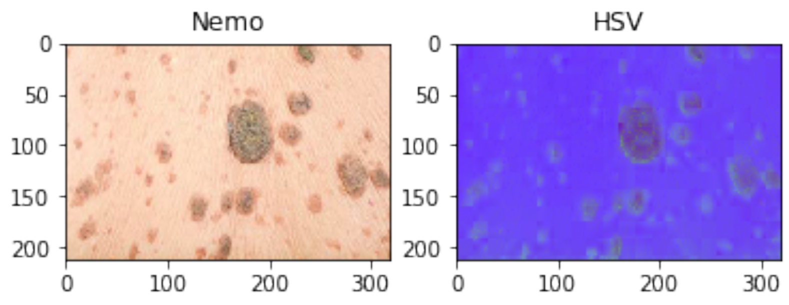
```
green = np.uint8([[[255,255,0]]])
hsv_green = cv2.cvtColor(green,cv2.COLOR_BGR2HSV)
print(hsv_green)
```

```
[[[ 90 255 255]]]
```

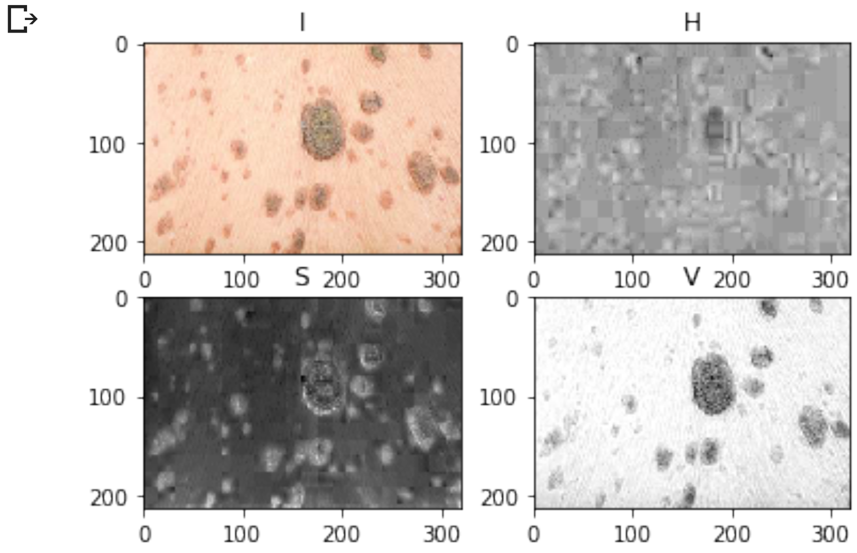
```
import cv2
import numpy as np

image = imread(linken + "Dataset_05/Melanoma 01.jpg")

hsv = cv2.cvtColor(image,cv2.COLOR_BGR2HSV)

lower_blue = np.array([80,150,150])
upper_blue = np.array([100,255,255])

mask = cv2.inRange(hsv, lower_blue, upper_blue)

res1 = cv2.bitwise_and(image,image, mask = 255 - mask)
res2 = cv2.bitwise_and(image,image, mask = mask)

imshows([image,mask,res1,res2],["Face","Mask","Res1","Res2"],2,2,gridType = Fals
```



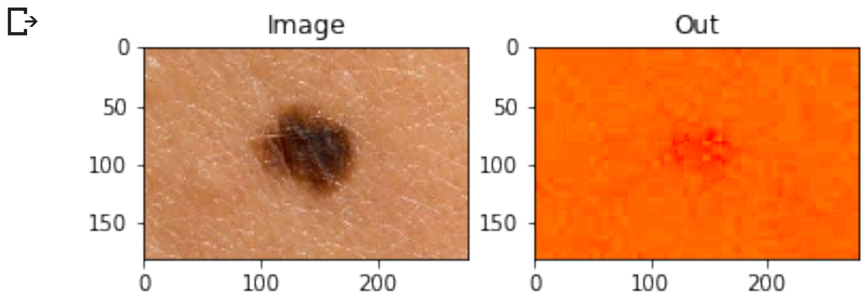
```
image = imread(linken + "Dataset_05/Melanoma 03.jpg")
hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

imshows([image, hsv],["Nemo","HSV"],1,2,gridType = False)
```



```
H, S, V = cv2.split(hsv)

imshows([image, H, S, V],["I","H","S","V"],2,2, gridType = False)
```



```
import cv2

image = imread(linken + "Dataset_05/Melanoma 01.jpg")
hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

h,s,v = cv2.split(hsv_image)
s.fill(255)
v.fill(255)
hsv_image = cv2.merge([h,s,v])

out = cv2.cvtColor(hsv_image, cv2.COLOR_HSV2BGR)

imshows([image, out],["Image","Out"],1,2, False)
```



```
##melanoma = segment_melanoma(image)
##imshows([image, melanoma],["Image","Melanoma"],1,2,False)
```

```python
def segment_melanoma(image):
  hsv_image = cv2.cvtColor(image, cv2.COLOR_RGB2HSV)
  light_brown = (1,60,10)
  dark_brown = (9,255,255)
  mask = cv2.inRange(hsv_image, light_brown, dark_brown)
  light_white = (0, 0, 100)
  dark_white = (0,255,255)
  mask_white = cv2.inRange(hsv_image, light_white, dark_white)
  final_mask = mask + mask_white
  result = cv2.bitwise_and(image, image, mask = final_mask)
  result = cv2.GaussianBlur(result,(7,7),0)
  return result
```

```python
image = imread(linken + "Dataset_05/Melanoma 01.jpg")
melanoma = segment_melanoma(image)
imshows([image, melanoma],["Image","Melanoma"],1,2, 0)

image = imread(linken + "Dataset_05/Melanoma 02.jpg")
melanoma = segment_melanoma(image)
imshows([image, melanoma],["Image","Melanoma"],1,2,0 )

image = imread(linken + "Dataset_05/Melanoma 06.jpg")
melanoma= segment_melanoma(image)
imshows([image, melanoma],["Image","Melanoma"],1,2, 0)
```

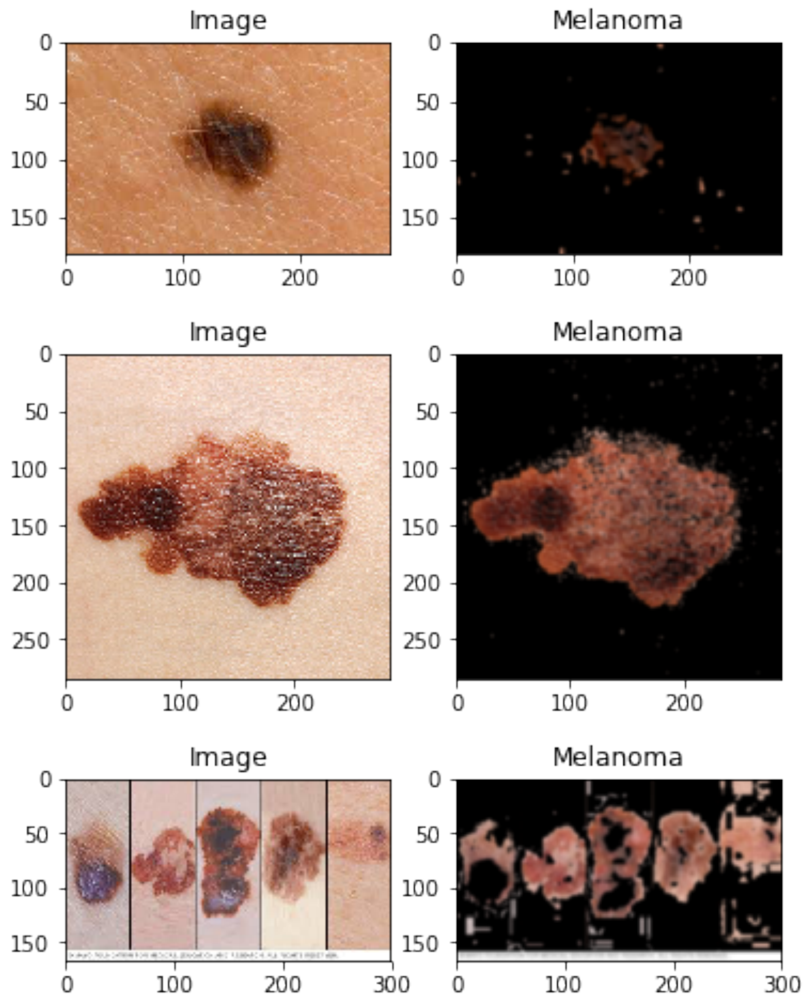
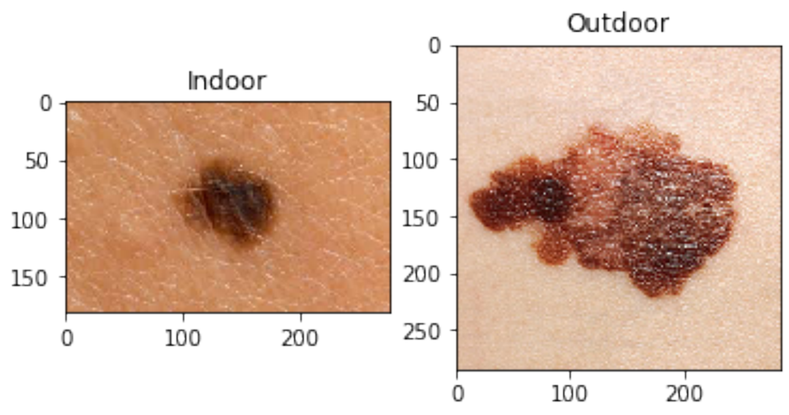


Image
Melanoma
Image
Melanoma
Image
Melanoma
0
50
100
150
200
250
300

```
dot_indoor = imread(linken + "Dataset_05/Melanoma 01.jpg")
dot_outdoor = imread(linken + "Dataset_05/Melanoma 02.jpg")

imshows([dot_indoor,dot_outdoor],["Indoor","Outdoor"],1,2,0)
```



```
R,G,B = cv2.split(dot_indoor)
imshows([cube_indoor, R, G, B],["I","R","G","B"],1,4,0)
R,G,B = cv2.split(dot_outdoor)
imshows([dot_outdoor, R, G, B],["I","R","G","B"],1,4,0)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last
<ipython-input-16-9b893d6ca5ec> in <module>()
      1 R,G,B = cv2.split(dot_indoor)
----> 2 imshows([cube_indoor, R, G, B],["I","R","G","B"],1,4,0)
      3 R,G,B = cv2.split(dot_outdoor)
      4 imshows([dot_outdoor, R, G, B],["I","R","G","B"],1,4,0)

NameError: name 'cube_indoor' is not defined
```

SEARCH STACK OVERFLOW

```
brightYCbCr = cv2.cvtColor(dot_outdoor, cv2.COLOR_BGR2YCrCb)
darkYCbCr = cv2.cvtColor(dot_indoor, cv2.COLOR_BGR2YCrCb)

Y, Cb, Cr = cv2.split(brightYCbCr)
imshows([dot_outdoor, Y, Cb, Cr],["Outdoor","Y","Cb","Cr"],1,4,0)

Y, Cb, Cr = cv2.split(darkYCbCr)
imshows([dot_indoor, Y, Cb, Cr],["Indoor","Y","Cb","Cr"],1,4,0)
```

```
image = imread(linken + "Dataset_05/Melanoma 01.jpg")
dotYCbCr = cv2.cvtColor(image, cv2.COLOR_BGR2YCrCb)

Y, Cb, Cr = cv2.split(dotYCbCr)
imshows([image,Y, Cb,Cr],["I","Y","Cb","Cr"],2,2,0)
```

```
brightHSV = cv2.cvtColor(dot_outdoor, cv2.COLOR_BGR2HSV)
darkHSV = cv2.cvtColor(dot_indoor, cv2.COLOR_BGR2HSV)

H, S, V = cv2.split(brightHSV)
imshows([dot_outdoor, H,S,V],["I","H","S","V"],1,4,0)
H, S, V = cv2.split(darkHSV)
imshows([dot_indoor, H,S,V],["I","H","S","V"],1,4,0)
```