

# Task 3 Image to Pencil Sketch(LGM)

By :- Rupali Rakhunde

## Import Dependencies

```
In [30]: import cv2
import numpy as np
from matplotlib import pyplot as plt

import warnings
warnings.filterwarnings('ignore')

In [31]: #READ AN IMAGE
img = cv2.imread("dhoni_img.jpg")

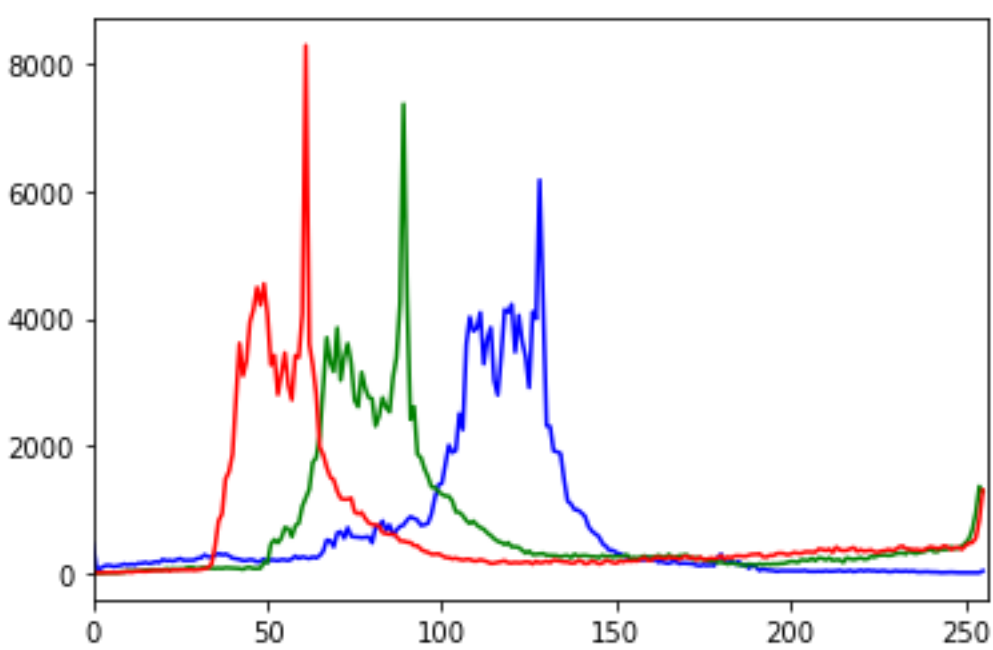
In [32]: #SEE THE TYPE OF IMAGE i.e AN NUMPY ARRAY
print(type(img))

<class 'numpy.ndarray'>

In [33]: #TO SEE THE IMAGE
plt.imshow(img)
plt.axis(False)
plt.show()
```



```
In [34]: color = ('b','g','r')
for i ,col in enumerate(color):
    histr = cv2.calcHist([img],[i],None,[256],[0,256])
    plt.plot(histr,color=col)
    plt.xlim([0,256])
plt.show()
```

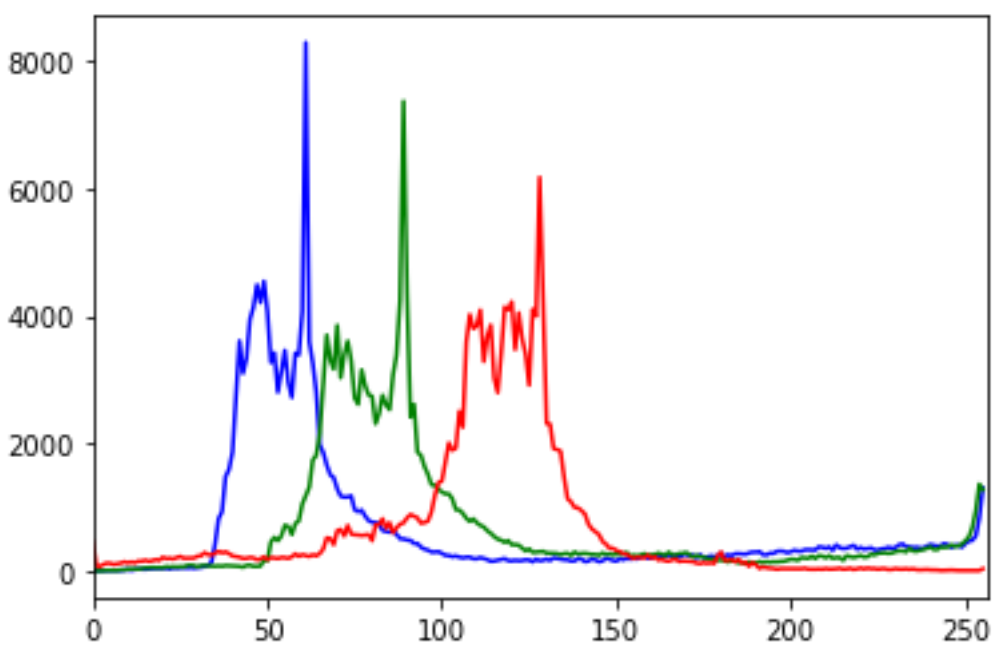


## convert an image from BGR to RGB

```
In [35]: #convert bgr to rgb
img_RGB = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
plt.imshow(img_RGB)
plt.axis(False)
plt.show()
```



```
In [36]: color = ('b','g','r')
for i ,col in enumerate(color):
    histr = cv2.calcHist([img_RGB],[i],None,[256],[0,256])
    plt.plot(histr,color=col)
    plt.xlim([0,256])
plt.show()
```



## Convert RGB to GrayScale

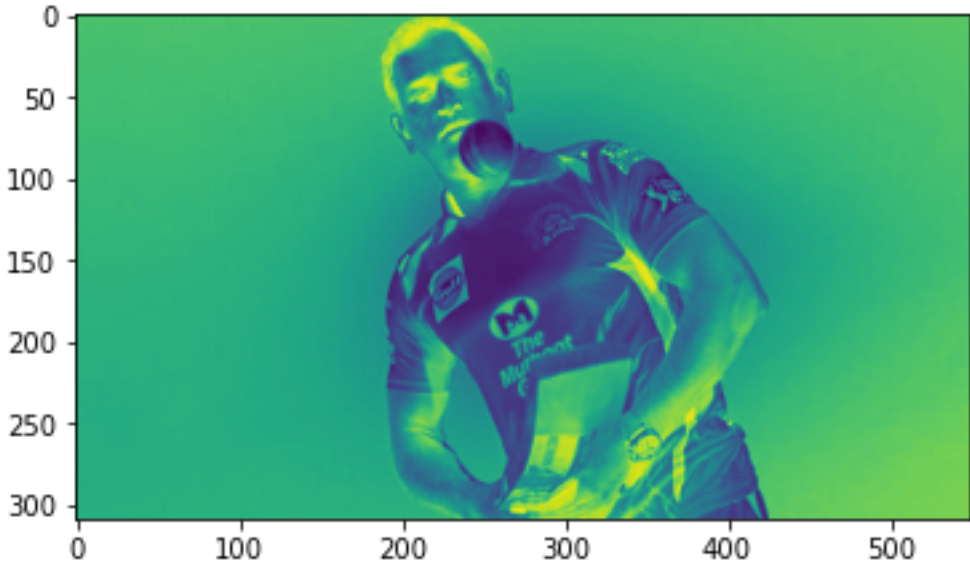
```
In [37]: img_gray = cv2.cvtColor(img,cv2.COLOR_RGB2GRAY)
plt.imshow(img_gray)
plt.axis(False)
plt.show()
```



## Inverting the image

```
In [10]: img_invert = cv2.bitwise_not(img_gray)
plt.imshow(img_invert)
```

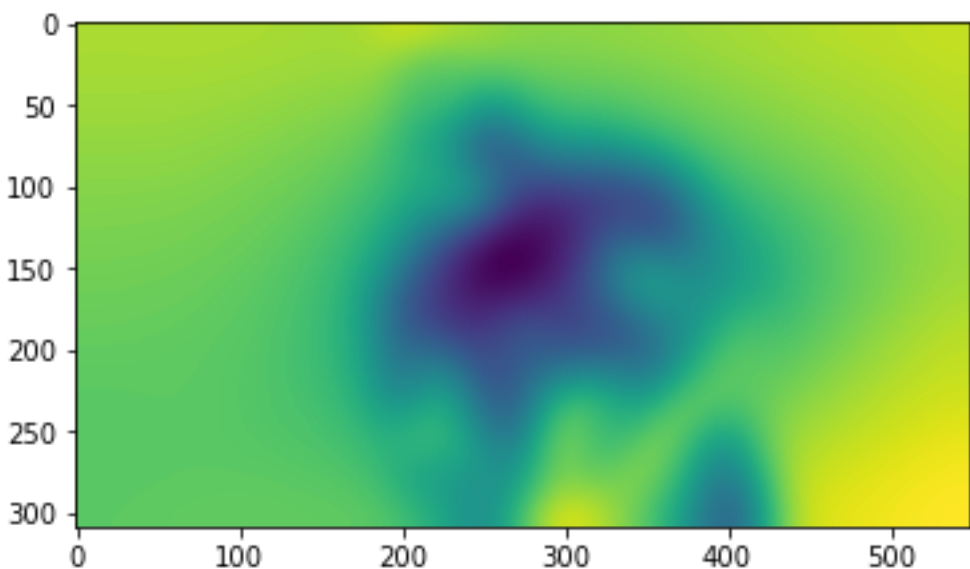
Out[10]: <matplotlib.image.AxesImage at 0x1cb46f92af0>



## Blur the image

```
In [42]: img_blur = cv2.GaussianBlur(img_invert,(111,111),0)
plt.imshow(img_blur)
```

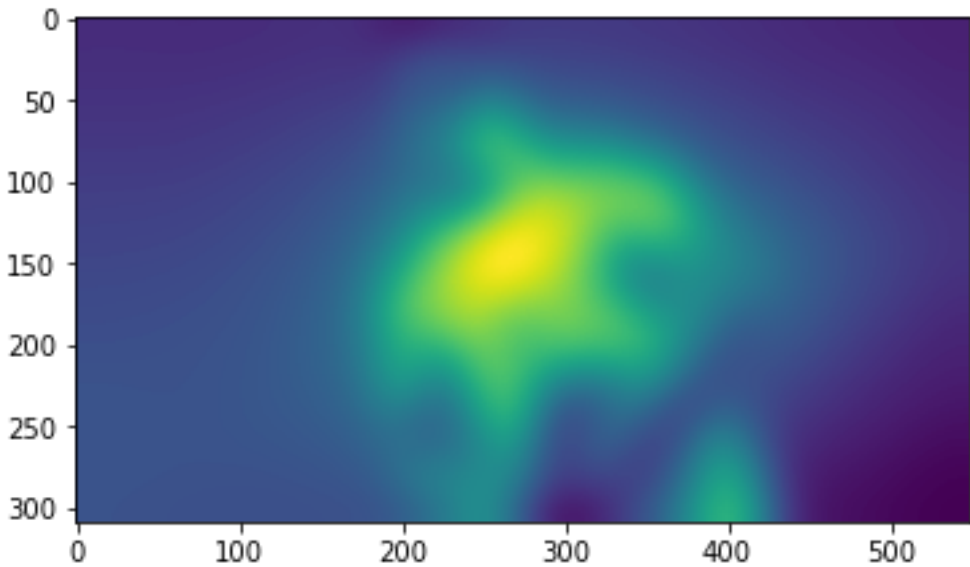
Out[42]: <matplotlib.image.AxesImage at 0x1cb485146d0>



## Inverting Blur Image

```
In [22]: invert_blur_img = cv2.bitwise_not(img_blur)
plt.imshow(invert_blur_img)
```

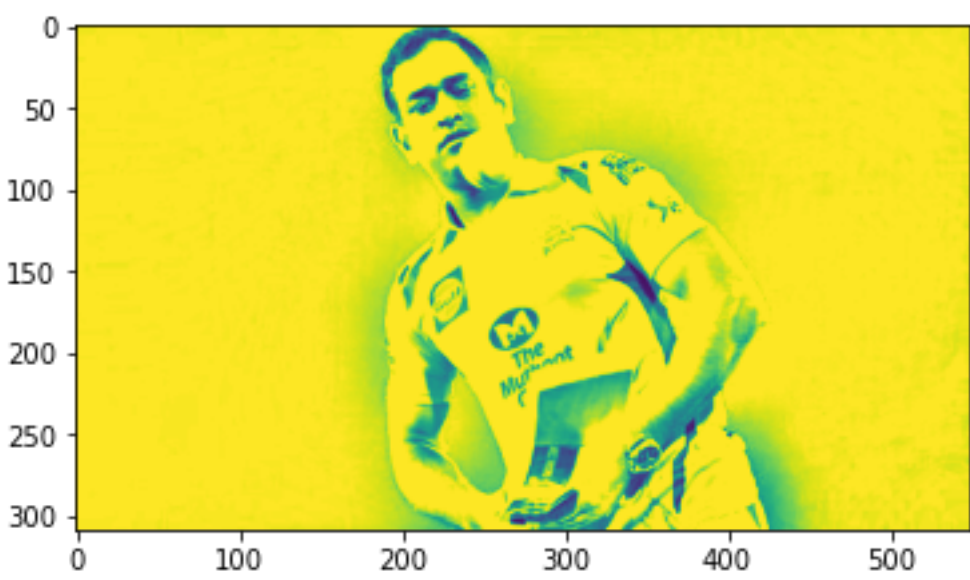
Out[22]: <matplotlib.image.AxesImage at 0x1cb480fb5e0>



## Convert it to Sketch Image

```
In [23]: img_sketch = cv2.divide(img_gray,invert_blur_img,scale=256.0)
plt.imshow(img_sketch)
```

Out[23]: <matplotlib.image.AxesImage at 0x1cb4815cdf0>



## Convert sketch to BGR

```
In [25]: plt.figure(figsize = (10,10))
img2 = cv2.cvtColor(img_sketch,cv2.COLOR_RGB2BGR)
plt.imshow(img2)
plt.axis(False)
plt.show()
```



In [ ]: