The Process to covert any colour image into it's sketch is following:

first we have to grayscale the image second we have to invert it then we have to blur the inverted image dodge merge the blurred and grayscale image.

In [16]:

```
import cv2
import numpy as np
import imageio
import scipy.ndimage
```

In [17]:

```
#img="pic.jpg"
img="https://i.dailymail.co.uk/i/pix/2018/01/02/11/47B0378700000578-5228403-image-a-60_
1514894368306.jpg"
```

In [18]:

```
#numpy doesnt convert any image to b&w(grayscale) but using the formula we can convert
 it:
#formula is: Y= 0.299 R + 0.587 G + 0.114 B
def grayscale(rgb):
    return np.dot(rgb[...,:3],[0.299,.587,0.114])
```

In [19]:

```
#We can invert images simply by subtracting from 255.
#as grayscale images are 8 bit images or have a maximum of 256 tones.
#The Colour Dodge blend mode divides the bottom layer by the inverted top layer.
#This lightens the bottom layer depending on the value of the top layer.
#We have the blurred image, which highlights the boldest edges.
def dodge(front,back):
    result=front*255/(255-back)
    result[result>255]=255
    result[result==255]=255
    return result.astype('uint8')
```

In [20]:

```
s=imageio.imread(img)
```

In [21]:

```
g=grayscale(s)
i=255-g
```

In [22]:

```
#Now we have to blur the image.
#we can do so by using filters
#here we are using Gaussian filter in the inverted image to make it blur.
#Sigma value determines the amount of blurring
#As sigma increases, the image becomes more blurred. Sigma controls the extent of the v
ariance and thus, the degree of blurring.

b=scipy.ndimage.filters.gaussian_filter(i,sigma=7)
```

In [23]:

```
#r=final_image
#b=blur_img
#g=gray_image
r=dodge(b,g)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:7: Runtim
eWarning: divide by zero encountered in true_divide
  import sys
```

In [9]:

```
#cv2.imwrite('pic1.png',r)
```

Out[9]:

```
True
```

In [25]:

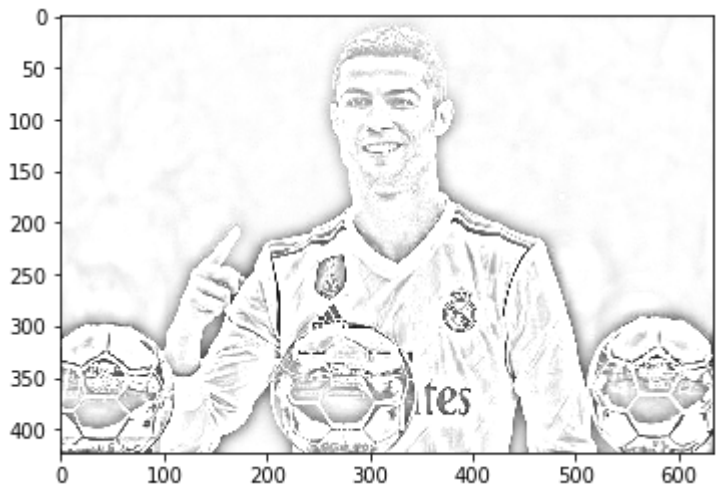
```
#plotting and saving
import matplotlib.pyplot as plt
```

In [26]:

```
plt.imshow(r,cmap="gray")
```

Out[26]:

```
<matplotlib.image.AxesImage at 0x28c7c072c48>
```



In [27]:

```
plt.imsave('img2.png', r, cmap='gray', vmin=0, vmax=255)
```

In [ ]:

```

```