Template matching

The best matches can be found as global minimums when CV_TM_SQDIFF is used or maximums when CV_TM_CCORR or CV_TM_CCOEFF is used

In [27]:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('images/messi5.jpg',0)
template = cv2.imread('images/messi_face.jpg',0)

w, h = template.shape[::-1]

res = cv2.matchTemplate(img,template,cv2.TM_)
min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(res)

print(type(min_val))
print(max_val)

print(min_loc)
print(max_loc)


top_left = max_loc
bottom_right = (top_left[0] + w, top_left[1] + h)

cv2.rectangle(img,top_left, bottom_right, 255, 2)

cv2.imshow("image",img)
cv2.waitKey(5000)
cv2.destroyAllWindows()
```

```
<class 'float'>
6683491.0
(228, 271)
(222, 84)
```

Canny edge detection

1. gaussian blur using 5x5 kernel to remove noise
2. gradient about x and y using sobel filter also find the direction of gradient which is always perpendicular to the edge.
3. detecting local maximums in the dicretion of the gradient (if not suppressed)
4. hysterisis thresholding ()

In [4]:

```
import cv2
import numpy as np
```

```
from matplotlib import pyplot as plt

img = cv2.imread('images/messi5.jpg',0)
edges = cv2.Canny(img,200,400) # min and max values for hytserisis thresholding

cv2.imshow("edges",edges)
cv2.imshow('messi',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Home Work

1) Using template matching make the mouse click automatically on a icon on your screen

Solution - problem 1

In [ ]:

```
from cv2 import cv2
import numpy as np
from matplotlib import pyplot as plt
import pyautogui as p
import time


# img1 = cv2.imread('images/mail.png',0)
template = cv2.imread('images/google.jpg',0)
p.hotkey('win','d')
time.sleep(0.5)

img = p.screenshot()


img = cv2.cvtColor(np.array(img),cv2.COLOR_RGB2BGR)
img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

# h1, w1 = img.shape[::-1]
h, w = template.shape[::-1]


# img = cv2.resize(img,(int(0.5*h1),int(0.5*w1)))
# img1 = cv2.resize(img1,(int(0.5*h1),int(0.5*w1)))
# template = cv2.resize(template,(int(0.5*h),int(0.5*w)))


res = cv2.matchTemplate(img,template,cv2.TM_CCORR_NORMED)
min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(res)

print(type(min_val))
print(max_val)

print(min_loc)
print(max_loc)


# top_left = max_loc
# bottom_right = (top_left[0] + int(0.5*h2), top_left[1] + int(0.5*w2))
```

```python
p.click(max_loc[0]+5,max_loc[1]+5)

# cv2.rectangle(img1,top_left, bottom_right, (255,0,0), 4)

# cv2.imshow("image",img1)
# cv2.waitKey(0)
# cv2.destroyAllWindows()
```