CV assignment 2

1. Load Image

```
import cv2
from PIL import Image
def show(x): # show the image in the cell of the jupyter without jumping every time the window
    display(Image.fromarray(x))

original =cv2.imread("img2.bmp", 0)
#load all the numbers and pass them to grey image with threshold
zero = cv2.imread(r"NUMBERS0-9\0.bmp",0)
one = cv2.imread(r"NUMBERS0-9\1.bmp",0)
two = cv2.imread(r"NUMBERS0-9\2.bmp",0)
three = cv2.imread(r"NUMBERS0-9\3.bmp",0)
four = cv2.imread(r"NUMBERS0-9\4.bmp",0)
five = cv2.imread(r"NUMBERS0-9\5.bmp",0)
six = cv2.imread(r"NUMBERS0-9\6.bmp",0)
seven = cv2.imread(r"NUMBERS0-9\7.bmp",0)
eight = cv2.imread(r"NUMBERS0-9\8.bmp",0)
nine = cv2.imread(r"NUMBERS0-9\9.bmp",0)
```

2. Grey image by threshold

```
im_bw, b_zero = cv2.threshold(zero, 80, 255, cv2.THRESH_BINARY)
im_bw, b_one = cv2.threshold(one, 80, 255, cv2.THRESH_BINARY)
im_bw, b_two = cv2.threshold(two, 80, 255, cv2.THRESH_BINARY)
im_bw, b_three = cv2.threshold(three, 80, 255, cv2.THRESH_BINARY)
im_bw, b_four = cv2.threshold(four, 80, 255, cv2.THRESH_BINARY)
im_bw, b_five = cv2.threshold(five, 80, 255, cv2.THRESH_BINARY)
im_bw, b_six = cv2.threshold(six, 80, 255, cv2.THRESH_BINARY)
im_bw, b_seven = cv2.threshold(seven, 80, 255, cv2.THRESH_BINARY)
im_bw, b_eight = cv2.threshold(eight, 80, 255, cv2.THRESH_BINARY)
im_bw, b_nine = cv2.threshold(nine, 80, 255, cv2.THRESH_BINARY)
```

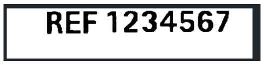
3. Cropping the image

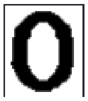
Use the kernel of size 2,2 and erode it to get the image clear.

```
# Cropping an image and transform it to grey image
print(original.shape)
cropped_image = original[500:600,900:1400]
show(cropped_image)
im_bw, b_crop = cv2.threshold(cropped_image, 100, 255, cv2.THRESH_BINARY)
b_crop = cv2.morphologyEx(b_crop,cv2.MORPH_ERODE,np.ones((2,2))) # in normal case it must be dilation, however we have an inverted grey ima
```

4. show the processed images

- processed image for both original cropped imaged and the numbers





5. Use cv2.matchTemplate to find patterns of the mask

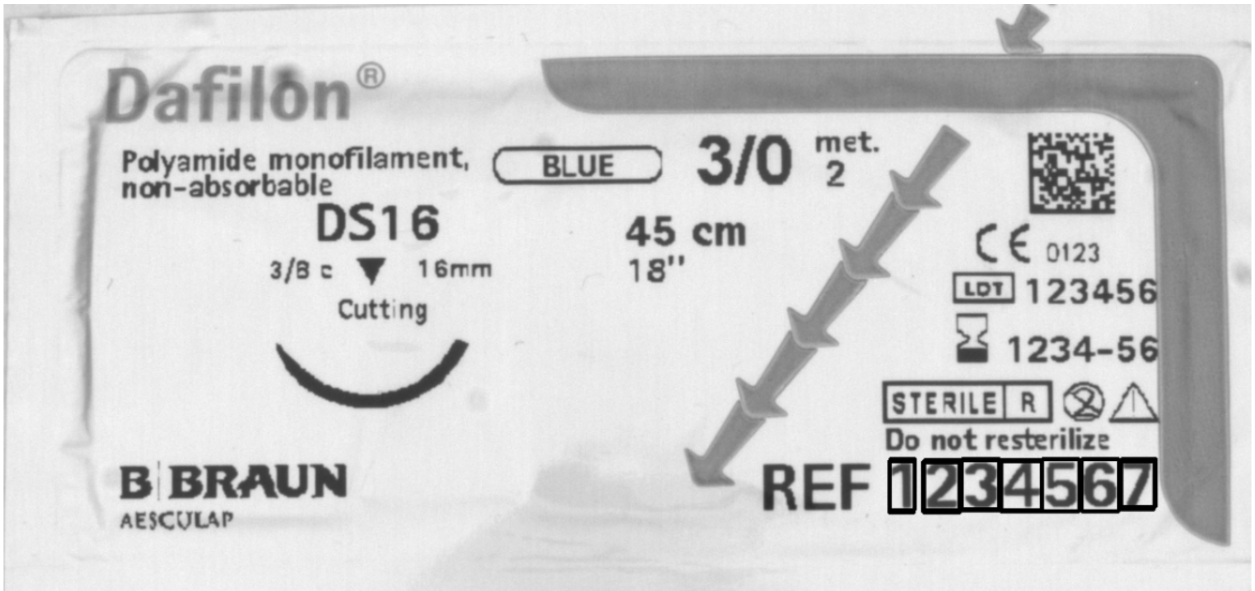
```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt

ref = []
for num,i in enumerate(nums): #apply each mask to the image
    w,h = i.shape[::-1]
    res = cv.matchTemplate(b_crop,i,cv.TM_CCOEFF_NORMED)
```

6. show the detected image with bounding box on the screen

```
ref = []
for num,i in enumerate(nums):
    w,h = i.shape[::-1]
    res = cv.matchTemplate(b_crop,i,cv.TM_CCOEFF_NORMED)
    threshold = 0.75
    loc = np.where( res >= threshold)
    many= len(loc[0])//4
    previous = [0,0]
    for pt in zip(*loc[::-1]):
        if (np.abs(pt[0]-previous[0])) <7:
            previous = pt[0],pt[1]
            pass
        else:
            cv2.rectangle(original,(pt[0]+900,pt[1]+500), (pt[0]+900 + w, pt[1]+500 + h), (0,0,255), 2) #500:600 - 900:1400
            ref.append([num,pt[0]])

        previous = pt[0],pt[1]
```

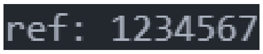


7. Counts the number

> Noted that the output in this cannot cannot be generalized since for different images the threshold of grey level is different to filter the noise and unecessary stuffs as well as the size and location of the cropping.

```
msg =""
for i in ref:
    msg +=str(i[0])
print("ref:",msg)
show(original)
## order the numbers by pt[0]
```

output: REF number



```
ref: 1234567
```