PubgRecognition

December 9, 2019

1

[466]:
```
from skimage import io, morphology, filters, color, data, measure
from matplotlib import pyplot as plt
import numpy as np
import cv2
import time

%matplotlib inline
```

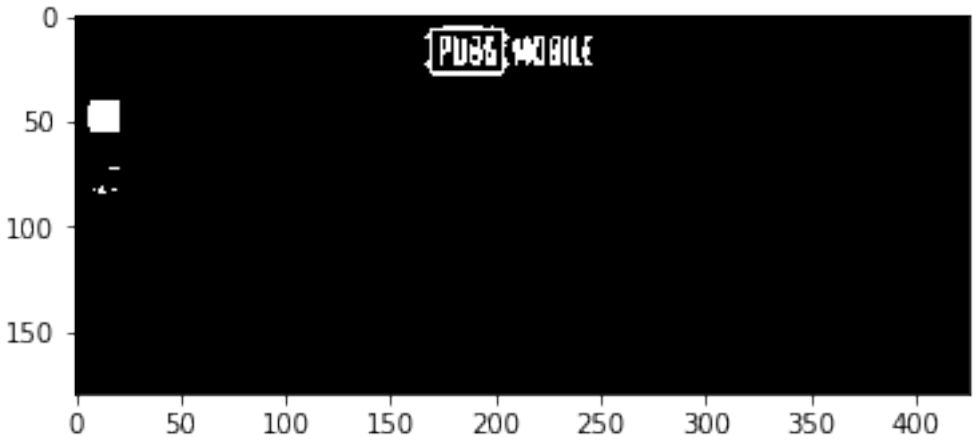
2 isInGame

[467]:
```
im = cv2.imread('3.jpg')
im = cv2.resize(im,(1280,720))
plt.imshow(im[:,:,::-1])
```

[467]:
```
<matplotlib.image.AxesImage at 0x7f510fe0fb00>
```

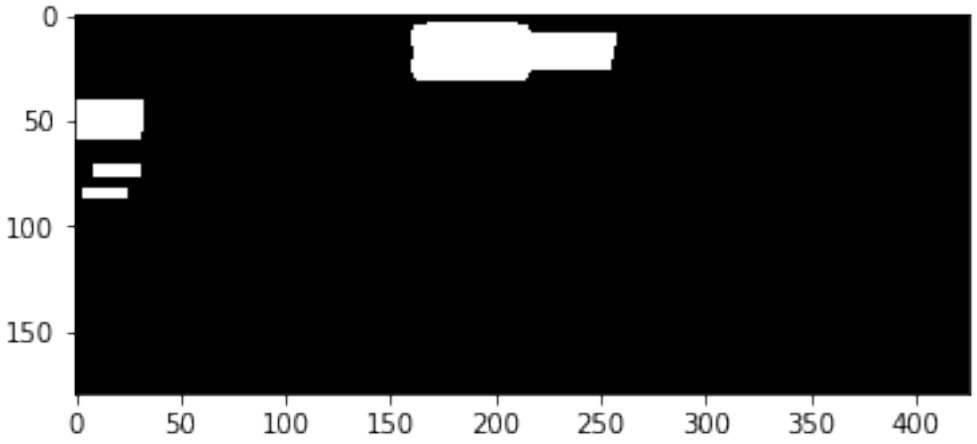
[468]:
```
#rgb2hsv,segment yellow part
h, w, c = im.shape
part = im[:h//4,:w//3,:]
hsv = cv2.cvtColor(part, cv2.COLOR_BGR2HSV)
hsv_low = np.array([15, 90, 150])
hsv_high = np.array([35, 255, 255])
mask = cv2.inRange(hsv, hsv_low, hsv_high)
plt.imshow(mask,cmap='gray')
```

[468]: <matplotlib.image.AxesImage at 0x7f510fdf94a8>



[469]:
```
#morphology op
k = cv2.getStructuringElement(cv2.MORPH_RECT,(2,2))
opening = cv2.morphologyEx(mask,cv2.MORPH_OPEN,k)
k2 = cv2.getStructuringElement(cv2.MORPH_RECT,(20,5))
dilated = cv2.dilate(opening,k2)
plt.imshow(dilated,cmap='gray')
```

[469]: <matplotlib.image.AxesImage at 0x7f510fd557f0>



[470]:
```
#contour analysis
_,contours,hierarchy = cv2.findContours(dilated,cv2.RETR_TREE,cv2.
 ↪CHAIN_APPROX_SIMPLE)
rects = []
for contour in contours:
    rect = cv2.boundingRect(contour)
    x,y,w,h = rect
    ratio = w/h
    if (h * w < 1000 or h * w > 6000 or h > 50 or ratio > 5 or ratio < 2):
        continue
    else:
        rects.append(rect)
rects
```

[470]:
```
[(160, 4, 98, 28)]
```

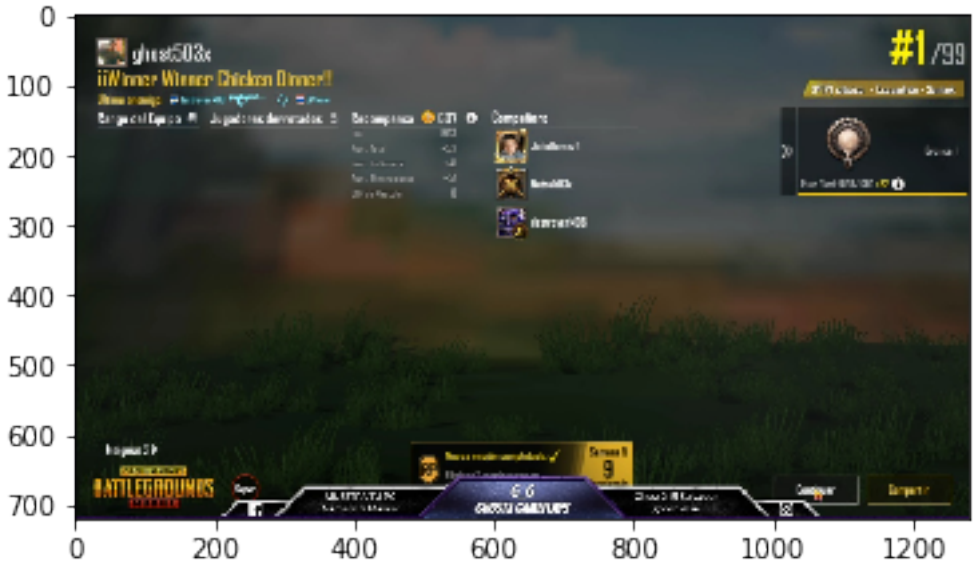
[471]:
```
#yellow pixel ratio
for rect in rects:
    x,y,w,h = rect
    roi = mask[y:y+h,x:x+w]
    ratio = (roi/255).sum()/(h*w)
    if(ratio>0.6 or ratio<0.2):
        continue
    else:
        print("isInGame")
```

```
isInGame
```

3 isSettlement

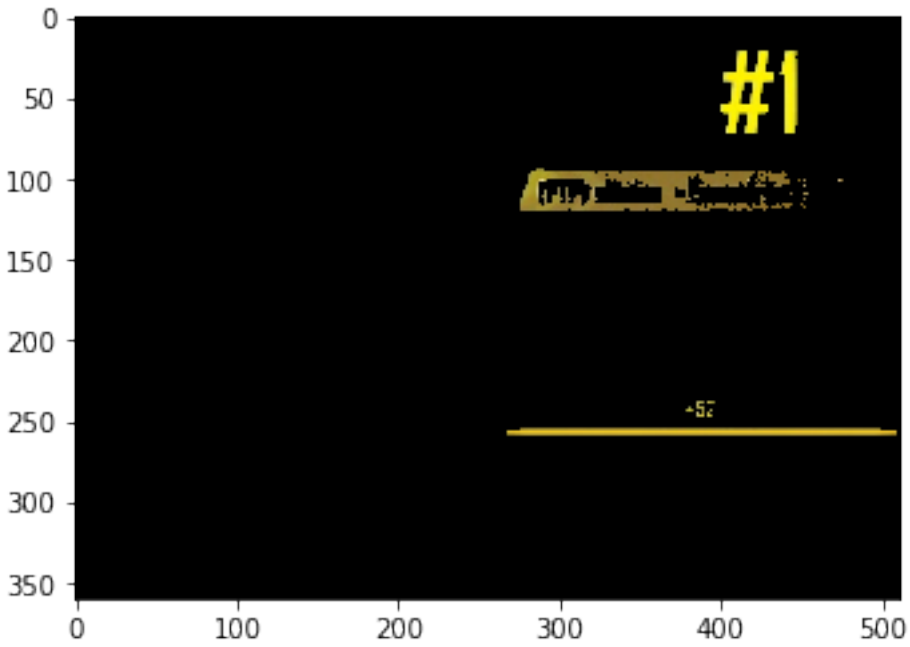
[472]:
```
im = cv2.imread('1.jpg')
im = cv2.resize(im,(1280,720))
plt.imshow(im[:,:,::-1])
```

[472]: <matplotlib.image.AxesImage at 0x7f510fd36a20>



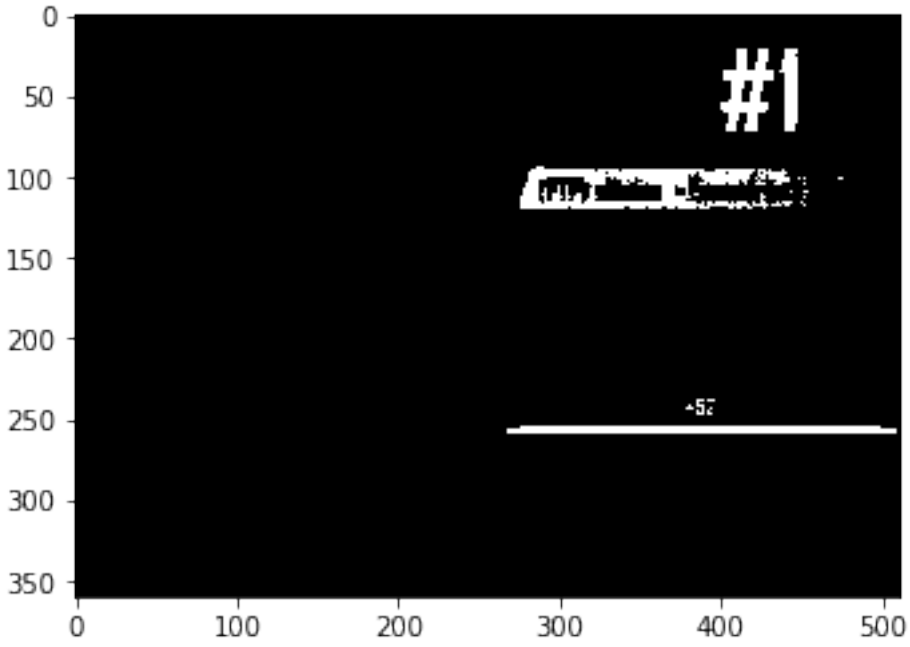
[473]:
```
#rgb2hsv,segment yellow part
h, w, c = im.shape
part = im[:h//2,int(w*0.6):,:]
hsv = cv2.cvtColor(part, cv2.COLOR_BGR2HSV)
hsv_low = np.array([20, 90, 140])
hsv_high = np.array([35, 255, 255])
mask = cv2.inRange(hsv, hsv_low, hsv_high)
yellow = cv2.bitwise_and(part, part, mask=mask)  #bgr
plt.imshow(yellow[:,:,::-1])
```

[473]: <matplotlib.image.AxesImage at 0x7f510fc9f668>



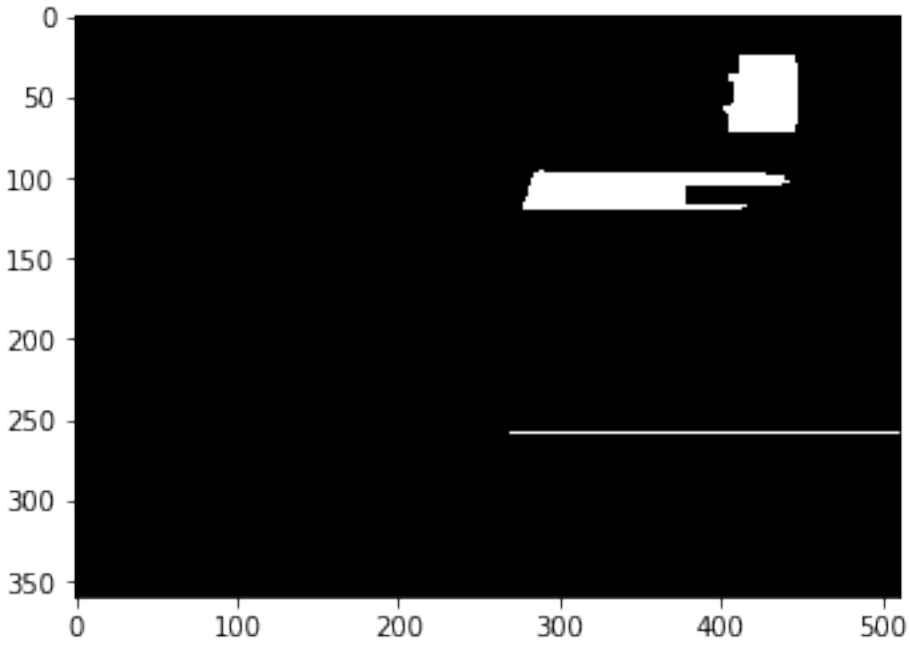
[474]:
```
#bgr2gray and binarize
gray = cv2.cvtColor(yellow,cv2.COLOR_BGR2GRAY)
thresh,binar= cv2.threshold(gray,0,255,cv2.THRESH_OTSU)
plt.imshow(binar,cmap='gray')
```

[474]: <matplotlib.image.AxesImage at 0x7f510fc7e780>



[475]:
```
#morphology op
k = cv2.getStructuringElement(cv2.MORPH_RECT,(3,3))
opening = cv2.morphologyEx(binar,cv2.MORPH_RECT,k)
k2 = cv2.getStructuringElement(cv2.MORPH_RECT,(70,10))
closing = cv2.morphologyEx(opening,cv2.MORPH_CLOSE,k2)
plt.imshow(closing,cmap='gray')
```

[475]:
```
<matplotlib.image.AxesImage at 0x7f510fbe5780>
```



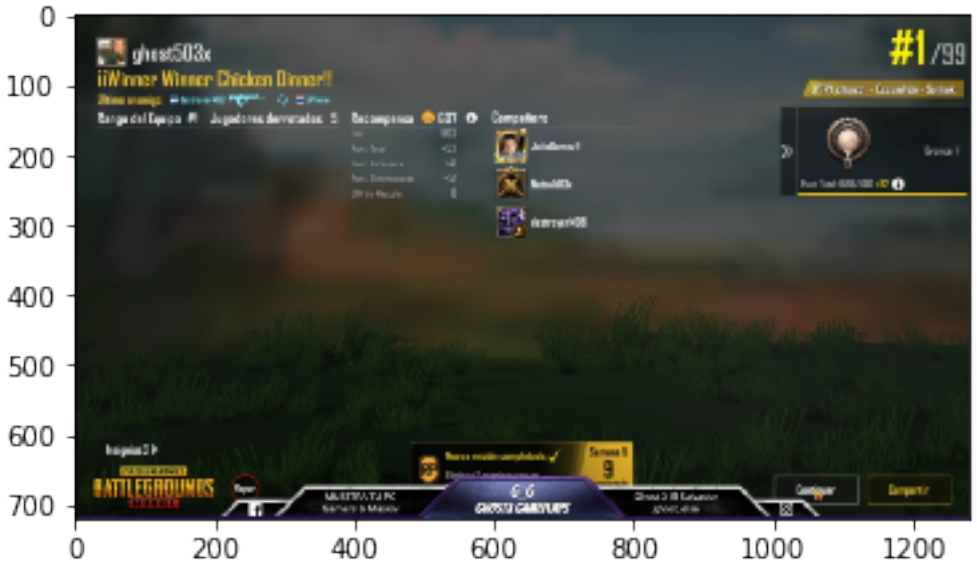
[476]:
```
#contour analysis
_,contours,hierarchy = cv2.findContours(closing,cv2.RETR_TREE,cv2.
 ↪CHAIN_APPROX_SIMPLE)
for contour in contours:
        x,y,w,h = cv2.boundingRect(contour)
        if(w>3*h or h>3*w or h*w<2000):
            continue
        else:
            print("isSettlement")
```

```
isSettlement
```

4 locateResult

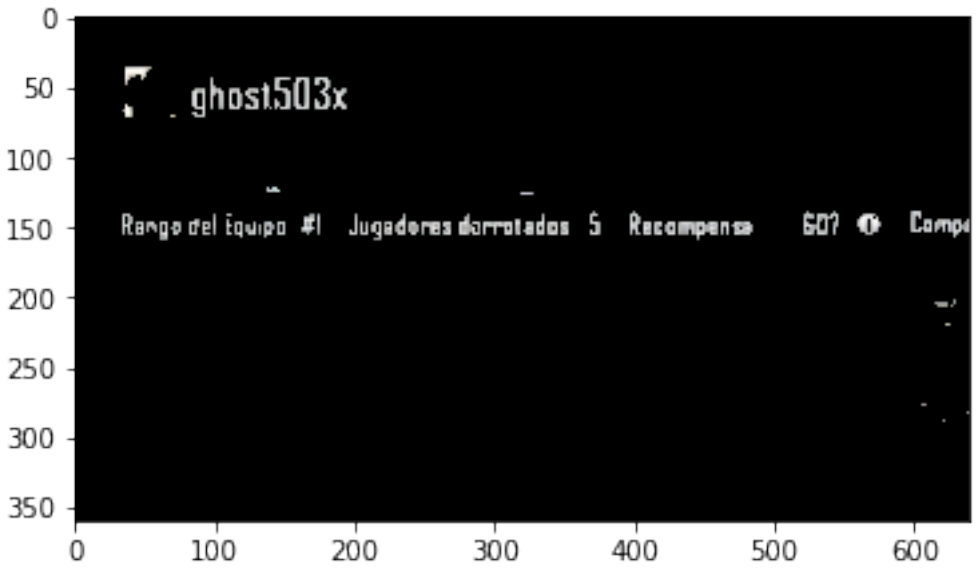
[477]:
```
im = cv2.imread('1.jpg')
im = cv2.resize(im,(1280,720))
im = cv2.GaussianBlur(im,(3,3),0)
plt.imshow(im[:,:,::-1])
```

[477]: <matplotlib.image.AxesImage at 0x7f510fb4c0f0>



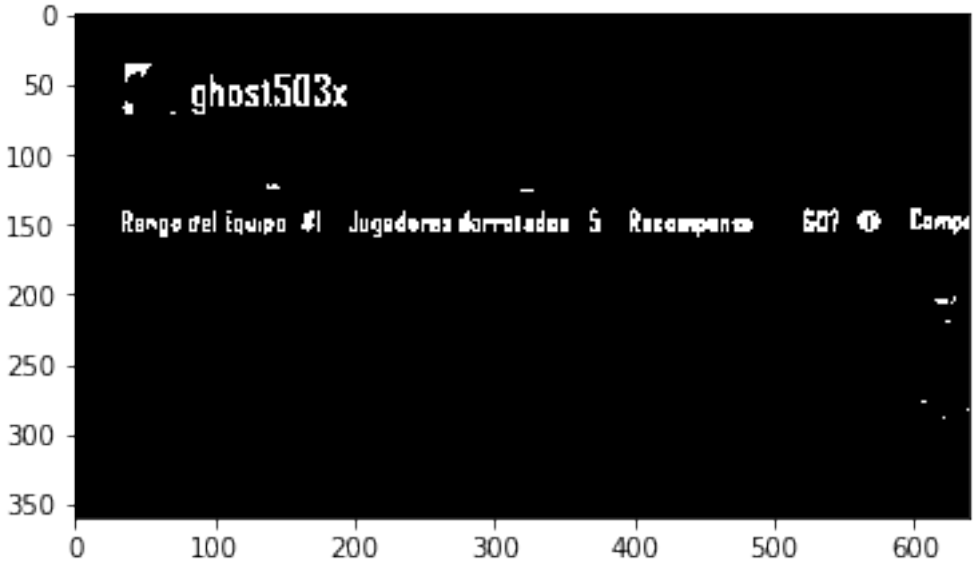
[478]:
```
#rgb2hsv,segment white part
h, w, c = im.shape
part = im[:h//2,:w//2,:]
hsv = cv2.cvtColor(part, cv2.COLOR_BGR2HSV)
hsv_low = np.array([0, 0, 150])
hsv_high = np.array([150, 30, 255])
mask = cv2.inRange(hsv, hsv_low, hsv_high)
white = cv2.bitwise_and(part, part, mask=mask)  #bgr
plt.imshow(white[:,:,::-1])
```

[478]: <matplotlib.image.AxesImage at 0x7f510fb2ea58>



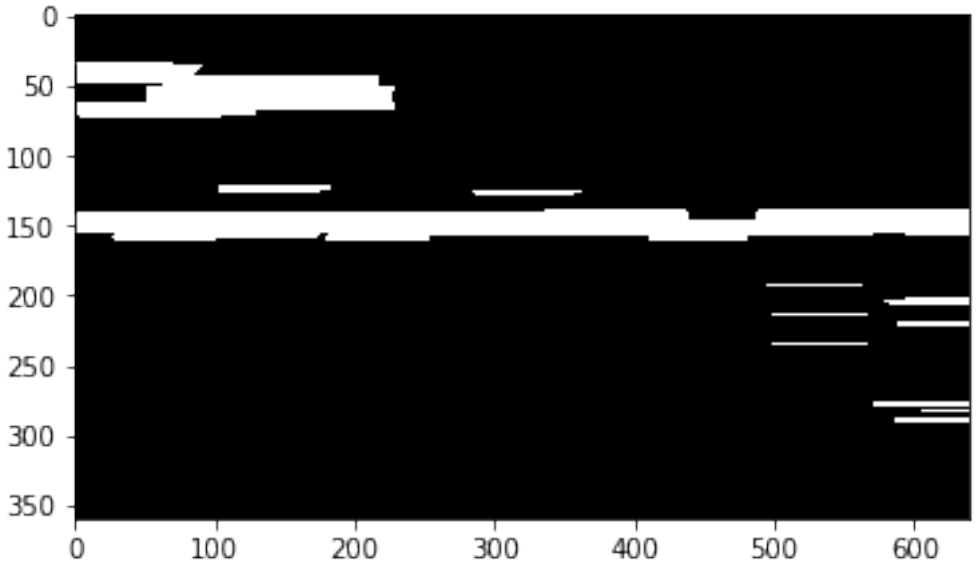
[479]:
```
#rgb2gray and binarize
gray = cv2.cvtColor(white,cv2.COLOR_BGR2GRAY)
thresh,binar = cv2.threshold(gray,0,255,cv2.THRESH_OTSU)
plt.imshow(binar,cmap='gray')
```

[479]: <matplotlib.image.AxesImage at 0x7f510fa96160>



[480]:
```
#morphology op
k = cv2.getStructuringElement(cv2.MORPH_RECT,(70,2))
dilated = cv2.dilate(binar,k)
plt.imshow(dilated,cmap='gray')
```

[480]: <matplotlib.image.AxesImage at 0x7f510fa7c1d0>

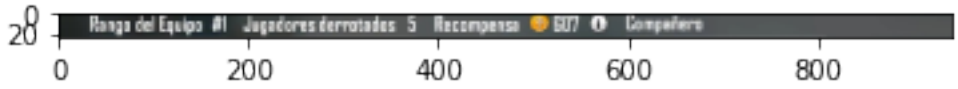


[481]:
```
#return the longest bar
maxWidth=0
maxRect=(0,0,0,0)
_,contours,_= cv2.findContours(dilated,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
for contour in contours:
    x,y,w,h = cv2.boundingRect(contour)
    if(w>maxWidth and h<50 and w>400):
        maxWidth=w
        maxRect=x,y,w,h
maxRect
```

[481]: (0, 139, 640, 22)

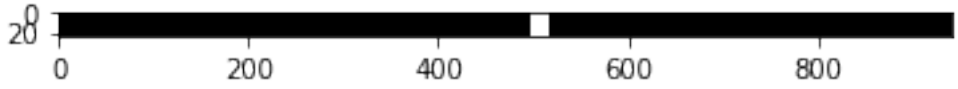
[482]:
```
x,y,w,h=maxRect
h = h+2
w = w+300
x = x
y = max(y-1,0)
bar = im[y:y+h,x:x+w]
plt.imshow(bar[:,:,::-1])
```

[482]: <matplotlib.image.AxesImage at 0x7f510f9ddb38>



[483]:
```
#find yellow circle
hsv = cv2.cvtColor(bar,cv2.COLOR_BGR2HSV)
k = cv2.getStructuringElement(cv2.MORPH_RECT,(10,h))
hsv_low = np.array([15, 150, 150])
hsv_high = np.array([35, 255, 255])
mask = cv2.inRange(hsv, hsv_low, hsv_high)
closing = cv2.morphologyEx(mask,cv2.MORPH_CLOSE,k)
plt.imshow(closing,cmap='gray')
```

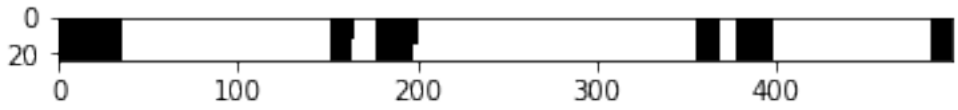
[483]: <matplotlib.image.AxesImage at 0x7f510f9b4240>



[484]:
```
#segment the longest bar according to the yellow circle
_,contours,_ = cv2.findContours(closing,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
if(len(contours)==1):
    c_x,c_y,c_w,c_h=cv2.boundingRect(contours[0])
    hsv_l = hsv[:,0:c_x]
    hsv_r = hsv[:,c_x+c_w:bar.shape[1]]
```

[485]:
```
#find number in left region
hsv_low = np.array([0, 0, 150])
hsv_high = np.array([150, 30, 255])
mask = cv2.inRange(hsv_l, hsv_low, hsv_high)
closing_l = cv2.morphologyEx(mask,cv2.MORPH_CLOSE,k)
_,contours_l,_ = cv2.findContours(closing_l,cv2.RETR_TREE,cv2.
 ↪CHAIN_APPROX_SIMPLE)
plt.imshow(closing_l,cmap='gray')
```

[485]: <matplotlib.image.AxesImage at 0x7f510f903780>



[486]:
```
rects =[]
for contour in contours_l:
    rect = cv2.boundingRect(contour)
    rects.append(rect)

rects = sorted(rects,key=lambda x:x[0])
boxes = []
x,y,w,h = rects[1]
x = max(0,x-3)
w = min(w+6,hsv_l.shape[1]-x)
if(w/h<2 and h/w<8):
    print("the first number")
    boxes.append((x,y,w,h))
```

```
the first number
```

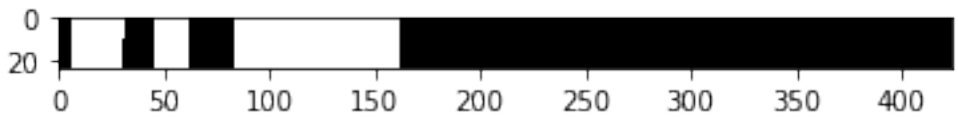
[487]:
```
x,y,w,h = rects[3]
x = max(0,x-3)
w = min(w+6,hsv_l.shape[1]-x)
if(w/h<2 and h/w<8):
    print("the second number")
    boxes.append((x,y,w,h))
```

```
the second number
```

[488]:
```
#find number in right region
hsv_low = np.array([0, 0, 150])
hsv_high = np.array([150, 30, 255])
mask = cv2.inRange(hsv_r, hsv_low, hsv_high)
closing_r = cv2.morphologyEx(mask,cv2.MORPH_CLOSE,k)
_,contours_r,_ = cv2.findContours(closing_r,cv2.RETR_TREE,cv2.
 ↪CHAIN_APPROX_SIMPLE)
plt.imshow(closing_r,cmap='gray')
boxes
```

[488]:
```
<matplotlib.image.AxesImage at 0x7f510f8db0b8>
```

[488]:
```
[(161, 0, 19, 24), (366, 0, 14, 24)]
```



[489]:
```
rects =[]
for contour in contours_r:
    rect = cv2.boundingRect(contour)
    rects.append(rect)

rects = sorted(rects,key=lambda x:x[0])
x,y,w,h = rects[0]
w = min(w+6,hsv_r.shape[1]-x)
x = max(0,x-3)+c_x+c_w
if(w/h<2 and h/w<8):
    print("the third nummber")
    boxes.append((x,y,w,h))
boxes
```

```
the third nummber
```

[489]:
```
[(161, 0, 19, 24), (366, 0, 14, 24), (519, 0, 32, 24)]
```

[490]:
```
blocks = []
for box in boxes:
    x,y,w,h = box
    blocks.append(bar[y:y+h,x:x+w])
plt.imshow(blocks[2][:,:,::-1])
```

[490]:
```
<matplotlib.image.AxesImage at 0x7f510f8af8d0>
```

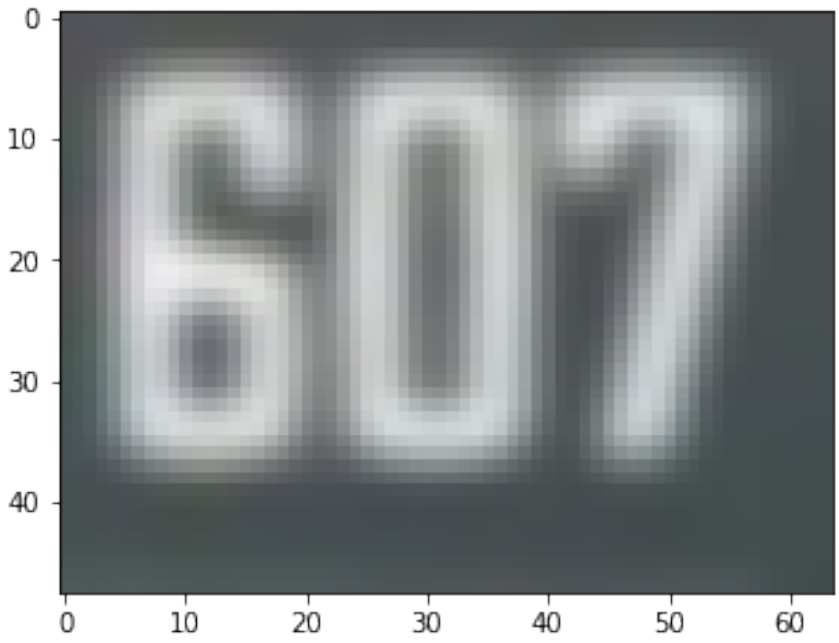
[491]:
```
#evaluate rank icon location
y1 = max(0,y-100)
y2 = min(y+300,im.shape[0])
x1 = max(0,im.shape[1]-350)
x2 = im.shape[1]
icon= im[y1:y2,x1:x2]
plt.imshow(icon[:,:,::-1])
```

[491]: <matplotlib.image.AxesImage at 0x7f510f808e10>

5 recognizeResult

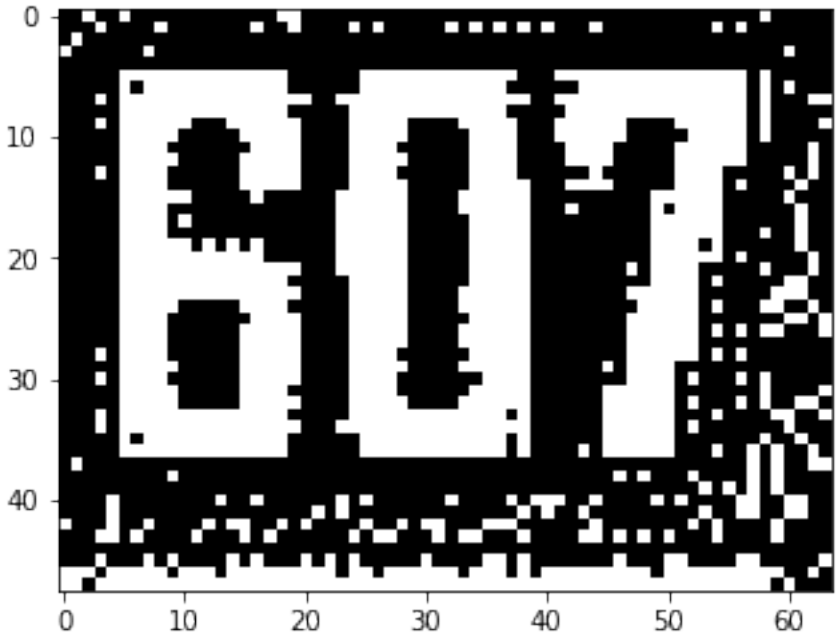
[492]:
```
#load result_sum model and predict
block = blocks[2]
block = cv2.resize(block,None,fx=2, fy=2, interpolation=cv2.INTER_CUBIC)
plt.imshow(block[:,:,::-1])
```

[492]: <matplotlib.image.AxesImage at 0x7f510f7e5da0>



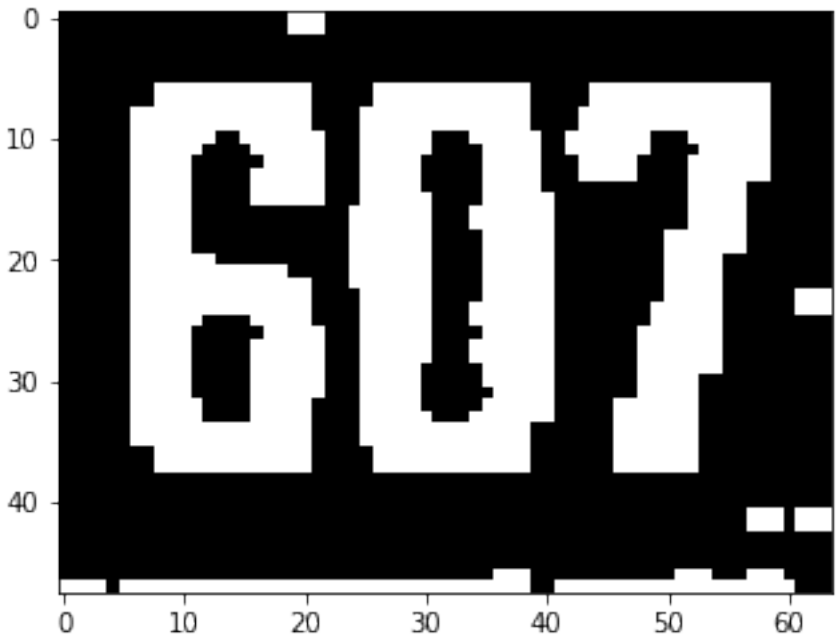
[493]:
```
#segment numbers
gray = cv2.cvtColor(block,cv2.COLOR_BGR2GRAY)
laplacian = cv2.Laplacian(gray,cv2.CV_64F)
laplacian = laplacian.astype(np.uint8)
thresh,binar = cv2.threshold(laplacian,0,255,cv2.THRESH_OTSU)
plt.imshow(binar,cmap='gray')
```

[493]:
```
<matplotlib.image.AxesImage at 0x7f510f74f0f0>
```



[494]:
```
#morphology op
k = cv2.getStructuringElement(cv2.MORPH_RECT,(2,2))
opening = cv2.morphologyEx(binar,cv2.MORPH_OPEN,k)
k2 = cv2.getStructuringElement(cv2.MORPH_RECT,(2,1))
dilated = cv2.dilate(opening,k2)
plt.imshow(dilated,cmap='gray')
```

[494]: <matplotlib.image.AxesImage at 0x7f510f72d208>



[495]:
```
#contours analysis
_,contours,_ = cv2.findContours(dilated,cv2.RETR_EXTERNAL,cv2.
 ↪CHAIN_APPROX_SIMPLE)
boxes = []
for contour in contours:
    if(cv2.contourArea(contour)>100):
        x,y,w,h = cv2.boundingRect(contour)
        if(w>h):
            continue
        else:
            boxes.append((x,y,w,h))
boxes = sorted(boxes,key=lambda x:x[0])
boxes
```

[495]:
```
[(6, 6, 16, 32), (24, 6, 17, 32), (42, 6, 17, 32)]
```

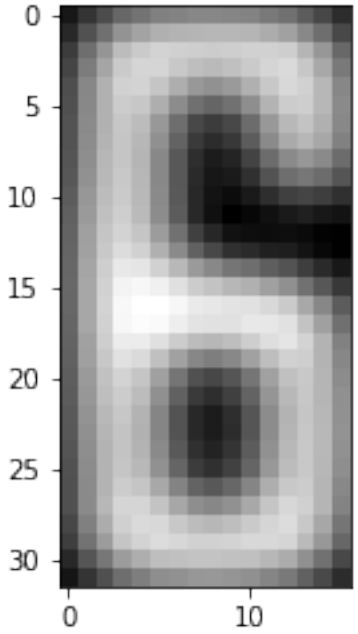
[496]:
```
boxes
maxh=0
for box in boxes:
    if box[3]>maxh:
        maxh=box[3]
x,y,w,h = boxes[0]
if(h/w>1.2 and h>maxh*0.8):
    y = max(y-1,0)
    x = max(x-2,0)
```

```
    h = min(h,gray.shape[0]-y+1)
    w = min(w,gray.shape[1]-x+2)
    num = gray[y:y+h,x:x+w]

plt.imshow(num,cmap='gray')
```

[496]:
```
[(6, 6, 16, 32), (24, 6, 17, 32), (42, 6, 17, 32)]
```

[496]:
```
<matplotlib.image.AxesImage at 0x7f510f68f358>
```

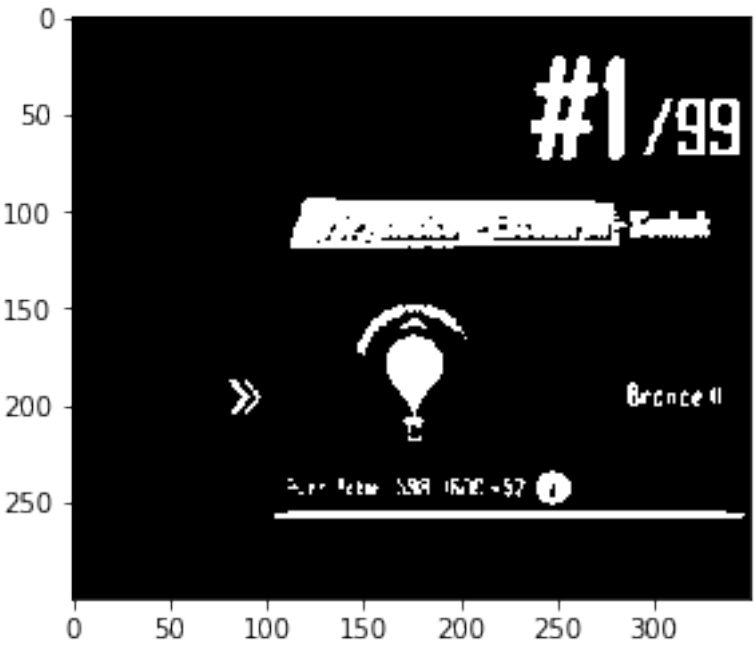


[497]:
```
#load sum model and recognize number
pass
```

6 locateRank

[498]:
```
gray = cv2.cvtColor(icon,cv2.COLOR_BGR2GRAY)
thresh,binar = cv2.threshold(gray,110,255,cv2.THRESH_BINARY)
plt.imshow(binar,cmap='gray')
```

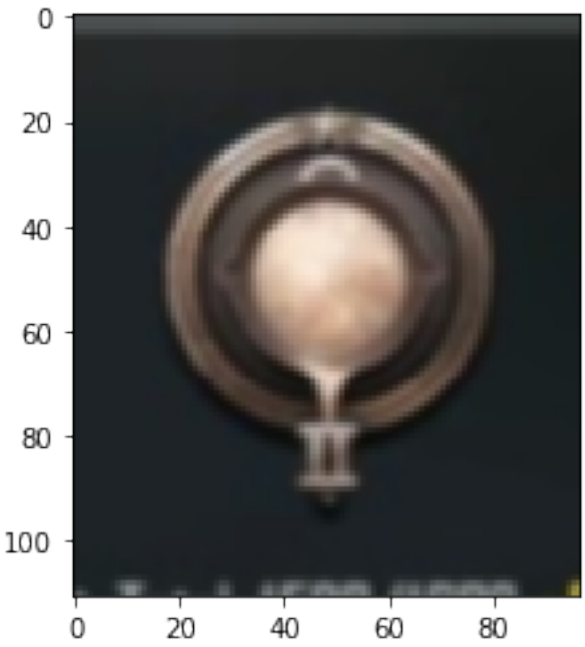
[498]:
```
<matplotlib.image.AxesImage at 0x7f510f8cfcf8>
```



[499]:
```
#locate rank icon
k = cv2.getStructuringElement(cv2.MORPH_RECT,(10,10))
closing = cv2.morphologyEx(binar,cv2.MORPH_CLOSE,k)
_,contours,_ = cv2.findContours(closing,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
rects = []
for contour in contours:
    x,y,w,h = cv2.boundingRect(contour)
    if(h/w<2 and w/h<2 and h>50 and h<150 and w<150 and w>50):
        rects.append((x,y,w,h))
if(len(rects)==1):
    print("rank icon")
    x,y,w,h = rects[0]
    x1 = max(x-20,0)
    y1 = max(y-20,0)
    x2 = min(x+w+20,icon.shape[1])
    y2 = min(y+h+20,icon.shape[0])
    rank = icon[y1:y2,x1:x2]
plt.imshow(rank[:,:,::-1])
```

```
rank icon
```

[499]: <matplotlib.image.AxesImage at 0x7f510f933ba8>



7 recognizeRank

[500]:
```
#load cnn model and predict
pass
```

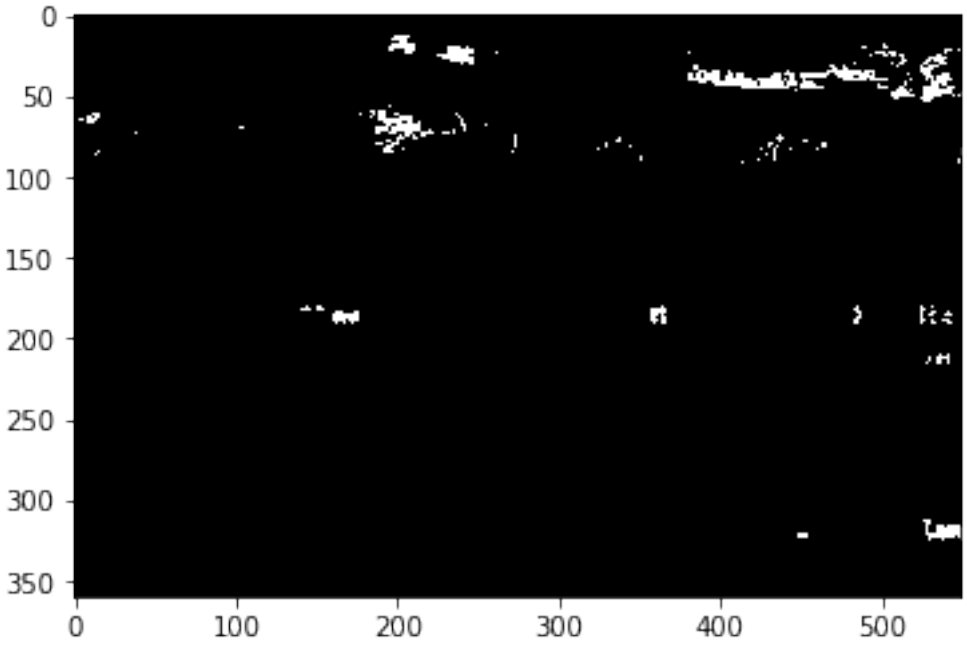
8 locateRate

[501]:
```
im = cv2.imread('2.jpg')
im = cv2.resize(im,(1280,720))
im = cv2.GaussianBlur(im,(3,3),0)
plt.imshow(im[:,:,::-1])
```

[501]: <matplotlib.image.AxesImage at 0x7f510f662fd0>

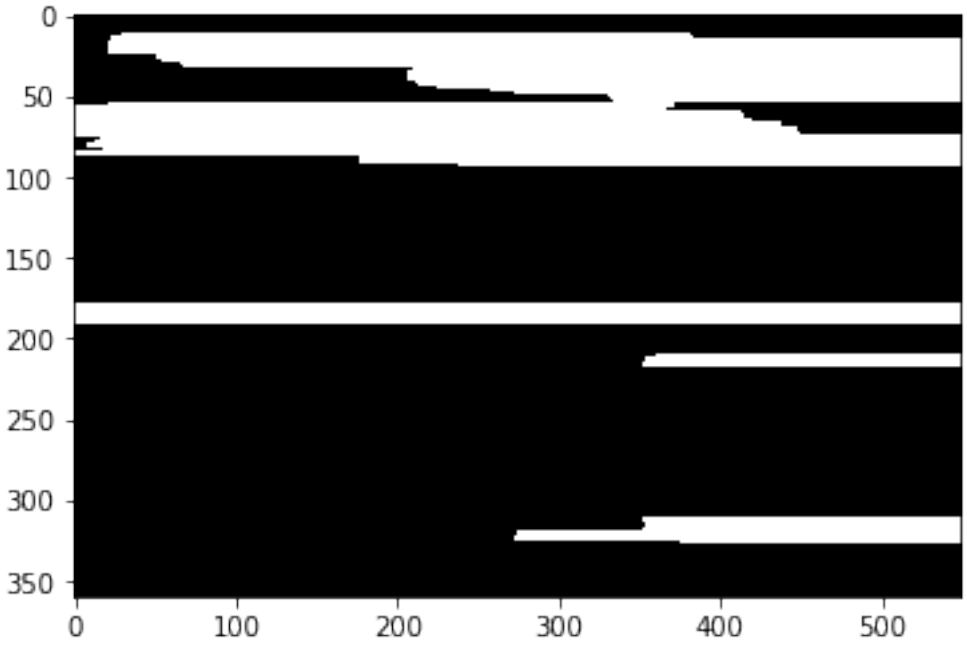
[502]:
```
#rgb2hsv,find yellow region
h, w, c = im.shape
w2 =int(0.93*w)
part = im[h//2:,w//2:w2,:]
hsv = cv2.cvtColor(part, cv2.COLOR_BGR2HSV)
hsv_low = np.array([20, 150, 100])
hsv_high = np.array([35, 255, 255])
mask = cv2.inRange(hsv, hsv_low, hsv_high)
plt.imshow(mask[:,:],cmap='gray')
```

[502]: <matplotlib.image.AxesImage at 0x7f510f5c99b0>



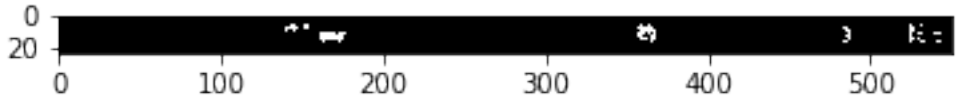
[503]:
```
#morphology op
k = cv2.getStructuringElement(cv2.MORPH_RECT,(350,3))
dilated = cv2.dilate(mask,k)
plt.imshow(dilated,cmap='gray')
```

[503]: <matplotlib.image.AxesImage at 0x7f510f5a6a58>



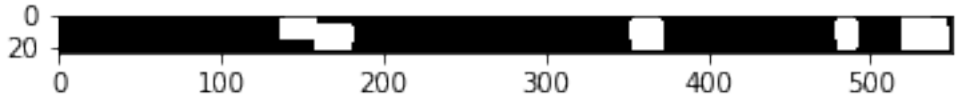
[504]:
```
_,contours,_ = cv2.findContours(dilated,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
rects =[]
for contour in contours:
    x,y,width,height = cv2.boundingRect(contour)
    if(width==(w2-w/2) and height<50 and height>10):
        y = max(y-5,0)
        height = min(height+10,h/2-y)
        rects.append((x,y,width,height))
x,y,w,h = rects[0]
bar = mask[y:y+h,x:x+w]
bar2 = part[y:y+h,x:x+w]
plt.imshow(bar,cmap='gray')
```

[504]: <matplotlib.image.AxesImage at 0x7f510f50fba8>



[505]:
```
k = cv2.getStructuringElement(cv2.MORPH_RECT,(10,10))
dilated = cv2.dilate(bar,k)
plt.imshow(dilated,cmap='gray')
```

[505]: <matplotlib.image.AxesImage at 0x7f510f4e5208>

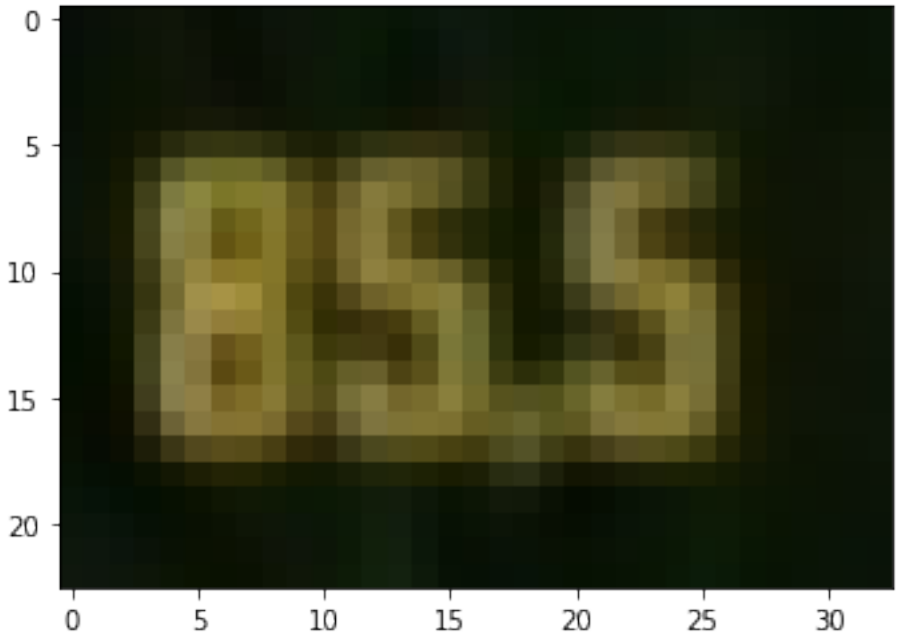


[506]:
```
#return right block
_,contours,_ = cv2.findContours(dilated,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
tmp=0
tmpRect=(0,0,0,0)
for contour in contours:
    x,y,w,h = cv2.boundingRect(contour)

    if(x>550 or x<300):
        continue
    if(x>tmp):
```

```
        tmp=x
        tmpRect = (x,y,w,h)
x,y,w,h = tmpRect
y = 0
x = max(x-2,0)
h = bar.shape[0]
w = min(w+4,bar.shape[1]-x)
ratio = w/h
if(ratio>0.5 and ratio<3):
    block = bar2[y:y+h,x:x+w]
plt.imshow(block[:,:,::-1])
```

[506]: <matplotlib.image.AxesImage at 0x7f510f4bb780>



9 recognizeRate

[507]:
```
#load rate_sum model and predict
pass
```