

```
In [1]: import cv2
from matplotlib import pyplot as plt
import numpy as np
import imutils
import easyocr
```

## 1. Read in Image, Grayscale

```
In [2]: img = cv2.imread('../data/generated1.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray, cv2.COLOR_BGR2RGB))
```

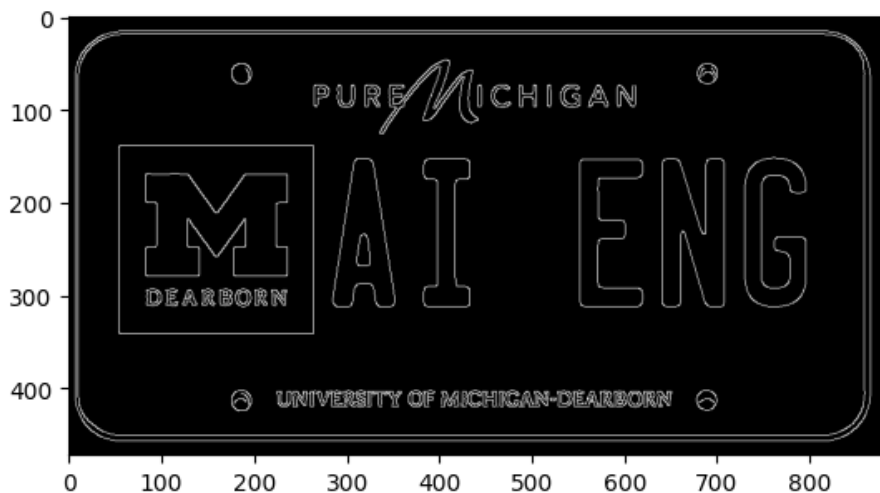
Out[2]: <matplotlib.image.AxesImage at 0x17ec98920>



## 2. Apply filter and find edges for localization

```
In [3]: bfilter = cv2.bilateralFilter(gray, 11, 17, 17) #Noise reduction
edged = cv2.Canny(bfilter, 20, 200) #Edge detection
plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))
```

Out[3]: <matplotlib.image.AxesImage at 0x17fe5aea0>



## 3. Find contours and apply masks

```
In [4]: keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours = imutils.grab_contours(keypoints)
```

```
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]
```

```
In [5]: location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour, 10, True)
    if len(approx) == 4:
        location = approx
        break
```

```
In [6]: contour
```

```
Out[6]: array([[ 55, 140]],
               [[ 54, 141]],
               [[ 54, 341]],
               [[ 55, 342]],
               [[264, 342]],
               [[264, 140]]], dtype=int32)
```

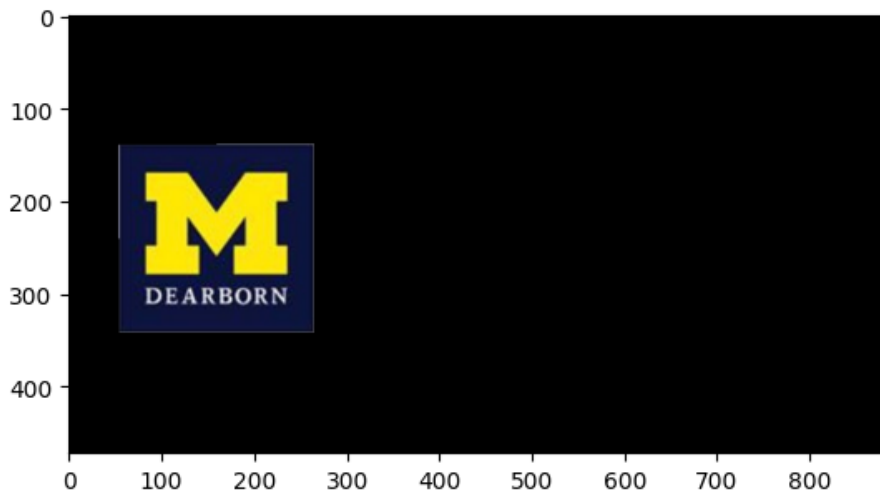
```
In [7]: location
```

```
Out[7]: array([[ 54, 141]],
               [[ 55, 342]],
               [[264, 342]],
               [[264, 140]]], dtype=int32)
```

```
In [8]: mask = np.zeros(gray.shape, np.uint8)
new_image = cv2.drawContours(mask, [location], 0, 255, -1)
new_image = cv2.bitwise_and(img, img, mask=mask)

plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_BGR2RGB))
```

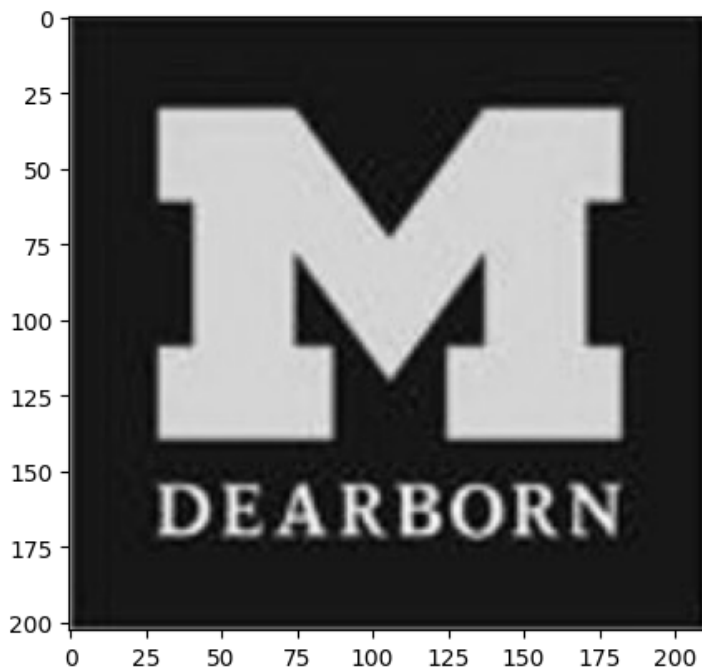
```
Out[8]: <matplotlib.image.AxesImage at 0x17fe59eb0>
```



```
In [9]: (x,y) = np.where(mask==255)
(x1, y1) = (np.min(x), np.min(y))
(x2, y2) = (np.max(x), np.max(y))
cropped_image = gray[x1:x2+1, y1:y2+1]
```

```
In [10]: plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
```

```
Out[10]: <matplotlib.image.AxesImage at 0x28a500290>
```



## 4. Use Easy OCR to Read Text

```
In [11]: reader = easyocr.Reader(['en'])
result = reader.readtext(cropped_image)
result
```

```
Out[11]: [[[24, 12], [192, 12], [192, 156], [24, 156]], 'M', 0.9956964164735496),
          [[26, 148], [190, 148], [190, 178], [26, 178]],
          'DEARBORN',
          0.9996900318258597]]
```

## 5. Render Result

```
In [12]: text = result[0][-2]
font = cv2.FONT_HERSHEY_SIMPLEX
res = cv2.putText(img, text=text, org=(approx[0][0][0], approx[1][0][1]+60), fontFace=font, fontSc
              thickness=2, lineType=cv2.LINE_AA)
res = cv2.rectangle(img, tuple(approx[0][0]), tuple(approx[2][0]), (0,255,0),3)
plt.imshow(cv2.cvtColor(res, cv2.COLOR_BGR2RGB))
```

```
Out[12]: <matplotlib.image.AxesImage at 0x28a57a480>
```

