In [9]:
```
import cv2
from matplotlib import pyplot as plt
import numpy as np
import imutils
import easyocr
from ultralytics import YOLO
```

1. Read in Image, Grayscale

In [10]:
```
img = cv2.imread('../data/good1.jpg')
plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
```

Out[10]:
```
<matplotlib.image.AxesImage at 0x17cacf500>
```

2. Import the pre-trained license detector model and find the license location

In [11]:
```
license_plate_detector = YOLO('license_plate_detector.pt')
```

In [12]:
```
license_plates = license_plate_detector(source=img, conf=0.60)
```

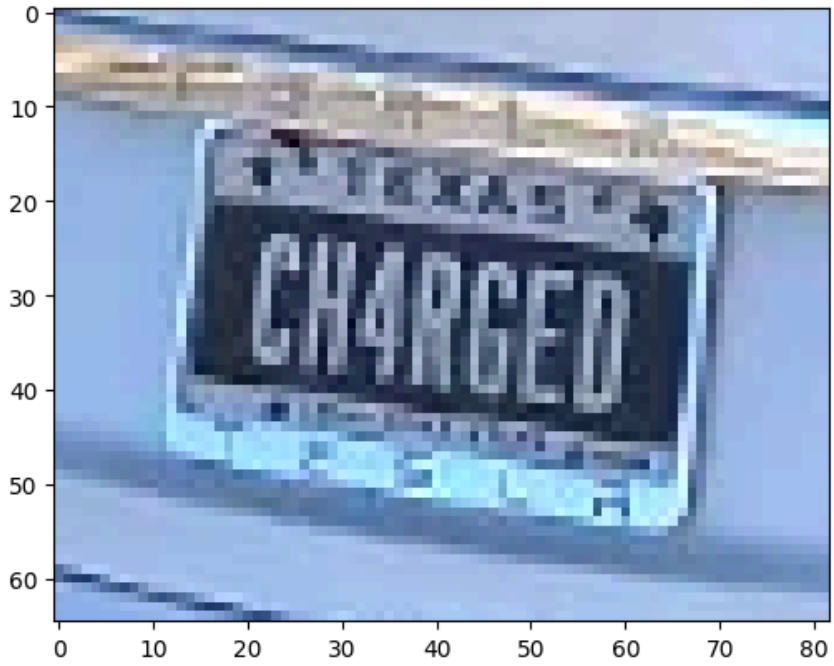
```
0: 320x640 1 license_plate, 49.8ms
Speed: 2.3ms preprocess, 49.8ms inference, 0.8ms postprocess per image at shape (1, 3, 320, 640)
```

In [13]:
```
for license_plate in license_plates[0].boxes.data.tolist():
            x1, y1, x2, y2, score, class_id = license_plate
```

3. Crop the image with the coordinates found above

In [14]:
```
 # crop license plate
license_plate_crop = img[int(y1):int(y2), int(x1): int(x2), :]
plt.imshow(cv2.cvtColor(license_plate_crop,cv2.COLOR_BGR2RGB))
```

Out[14]:
```
<matplotlib.image.AxesImage at 0x2a0f85e20>
```



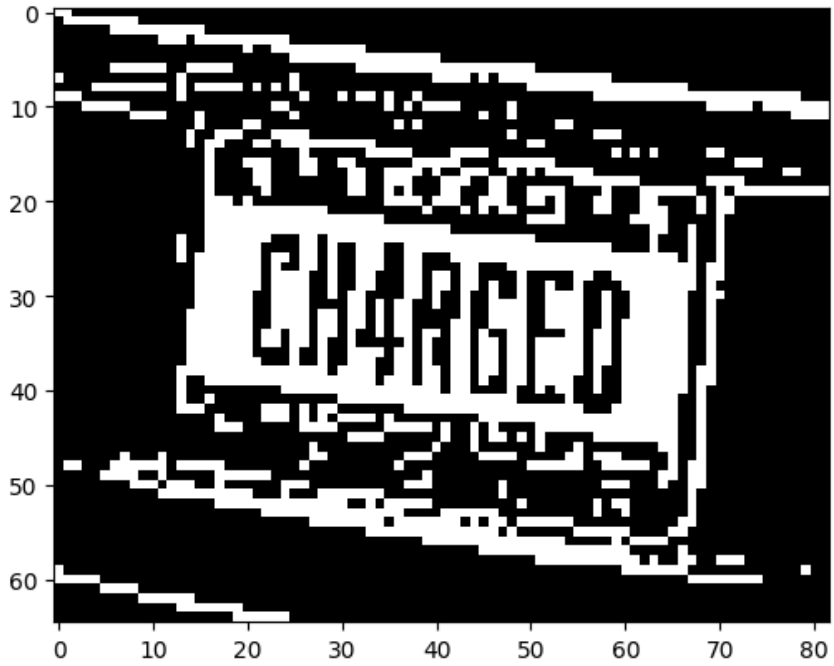
4. Treat the cropped image

In [15]:
```
blurred = cv2.GaussianBlur(license_plate_crop, (3, 3), 0)  # Apply Gaussian blur
sharpened = cv2.addWeighted(license_plate_crop, 3, blurred, -1.95, 0)# Sharpen the image

gray = cv2.cvtColor(sharpened, cv2.COLOR_BGR2GRAY)  # Convert to grayscale
_, binary = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)  # Thresholding

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

Out[15]:
```
<matplotlib.image.AxesImage at 0x2a0ecef60>
```



5. Predict the text contained in the treated image

In [16]:
```
reader = easyocr.Reader(['en'])
result = reader.readtext(sharpened, detail=0)
result
```

Out[16]:
```
['ChArGED']
```