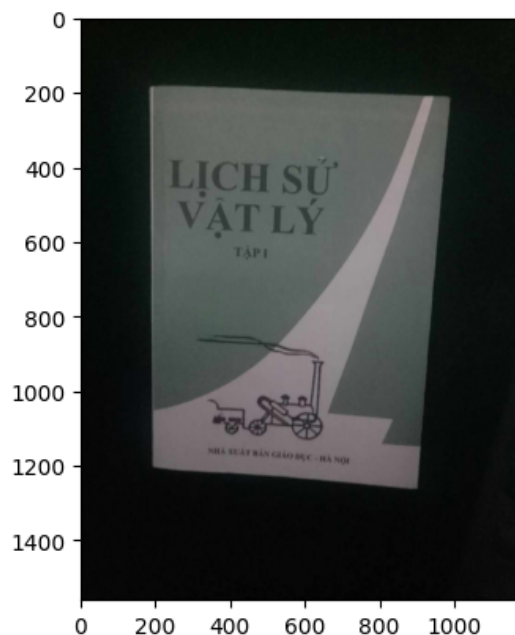


```
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

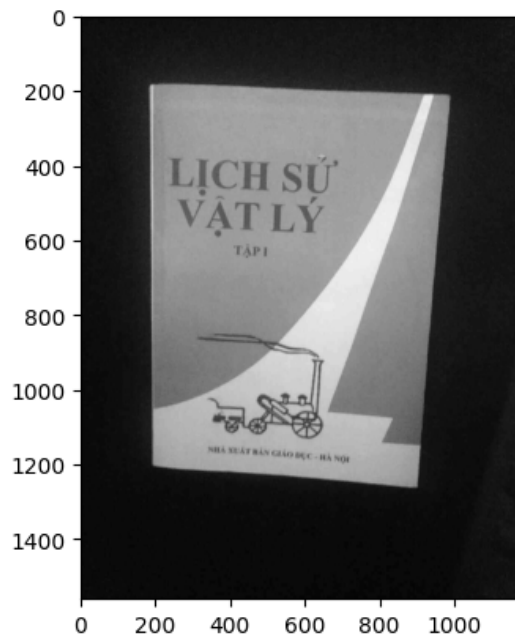
```
# Khởi tạo biến ảnh
img = cv2.imread('book.jpg')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # Convert thành gam màu chuẩn mắt người
plt.imshow(img) # Hiện ảnh original
```

<matplotlib.image.AxesImage at 0x7845f7845180>



```
gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY) # Convert thành gray
plt.imshow(gray, cmap='gray') # Hiện ảnh vừa chia ngưỡng, dưới dạng gray
```

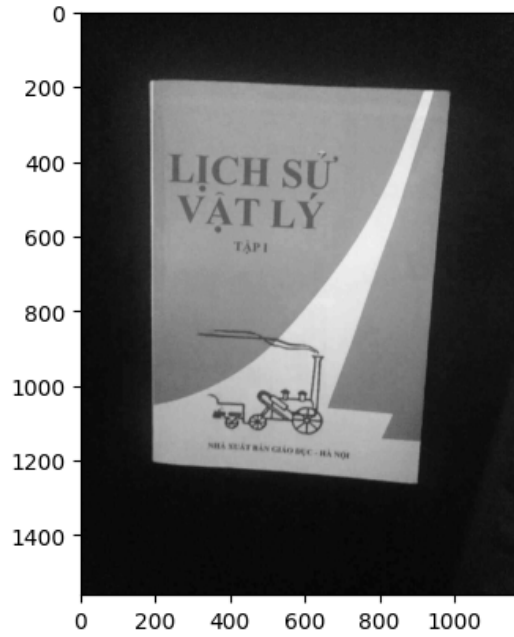
<matplotlib.image.AxesImage at 0x7845f7648d60>



```
gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY) # Convert thành gray
_, thresh = cv2.threshold(gray, 70, 255, cv2.THRESH_BINARY) # Chia ngưỡng binary sao cho chữ số "Tên cuốn sách" hiện rõ
plt.imshow(thresh, cmap='gray') # Hiện ảnh vừa chia ngưỡng, dưới dạng gray
```

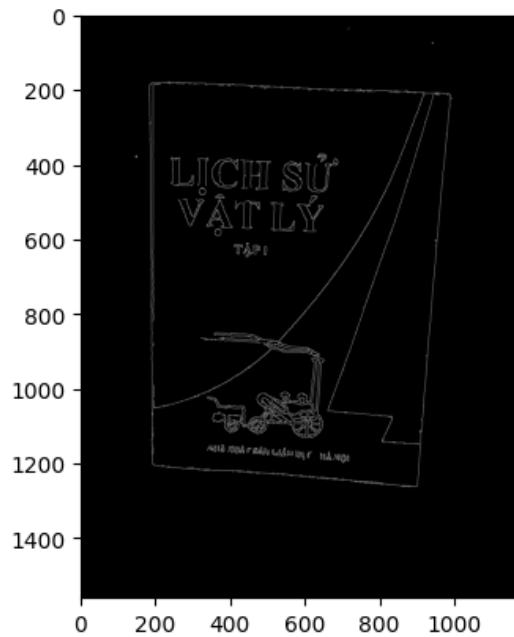
```
plt.imshow(gray, cmap= gray ) # hiện ảnh vừa của người, với ứng gray
```

```
<matplotlib.image.AxesImage at 0x7845f6e4ada0>
```



```
edge = cv2.Canny(gray, 50,100) # Lấy đường biên
plt.imshow(edge, cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7845f6fb9810>
```



```
# Tìm contours trong ảnh đường biên
contours, hierarchy = cv2.findContours(edge.copy(), cv2.RETR_CCOMP, cv2.CHAIN_APPROX_SIMPLE)
contours = sorted(contours, key = cv2.contourArea, reverse = True)[:10]
test = img.copy()
```

```
# Loop over our contours to find the license plate contour
plate_contour = None
for c in contours:
    # approximate the contour
    peri = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, 0.02 * peri, True)
```

```

# if our approximated contour has four points, we can assume that we have found our license plate
if len(approx) == 4:
    plate_contour = approx
    break
# Lấy hình chữ nhật giới hạn cho đường viền biển số xe
rect = cv2.minAreaRect(plate_contour)
box = cv2.boxPoints(rect)
box = np.int0(box)

# Lấy các chiều cho ma trận
width = int(rect[1][0])
height = int(rect[1][1])

src_pts = box.astype("float32")
dst_pts = np.array([[0, height-1],
                    [0, 0],
                    [width-1, 0],
                    [width-1, height-1]], dtype="float32")
# Get the perspective transform matrix and warp the image to get a top-down view of it
M = cv2.getPerspectiveTransform(src_pts, dst_pts)
warped = cv2.warpPerspective(img, M, (width, height))

# Convert color space for matplotlib
warped_rgb = cv2.cvtColor(warped, cv2.COLOR_BGR2RGB)
plt.imshow(warped_rgb)

```

<matplotlib.image.AxesImage at 0x7845f70928c0>

