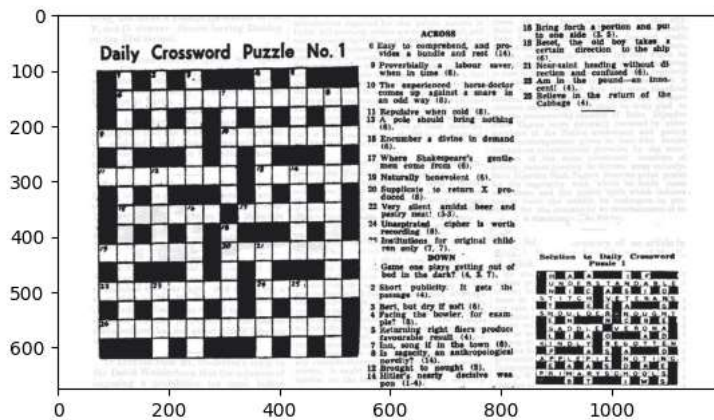


```
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

```
image=cv2.imread('/content/crosswordimage.jpg')
plt.imshow(image)
```

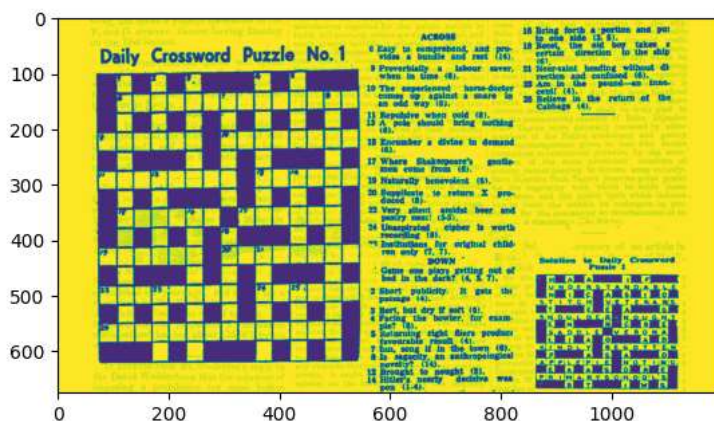


<matplotlib.image.AxesImage at 0x78fbca6a8bb0>



```
# grayscale color space
operatedImage=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
plt.imshow(operatedImage)
```

<matplotlib.image.AxesImage at 0x78fbca683880>



```
# modify the data type
# setting to 32 bit floating point
operatedImage = np.float32(operatedImage)
# 32 bit floats can provide sufficient precision
# without consuming excessive memory
```

```
# apply the cv2.cornerHarris method
# to detect the corners with appropriate
# Values as input parameters
# (Gray scale img,neighbourhood,kernel size and k value)
# k=0.07 make the algorithm more sensitive to corners,which you can adjust this parameter based on the characteristics
# and the desired corner detection results#Results are marked through the dilated corners
dest=cv2.cornerHarris(operatedImage,2,5,0.07)
```

```
# results are marked through the dilated corners
# increase the bright area portions
dest = cv2.dilate(dest,None)
```

```
# reverting back to the original image,
# with optimal threshold value
image[dest > 0.01 * dest.max()]=[0,0,255]
```

```
plt.imshow(image,cmap="gray")
```

