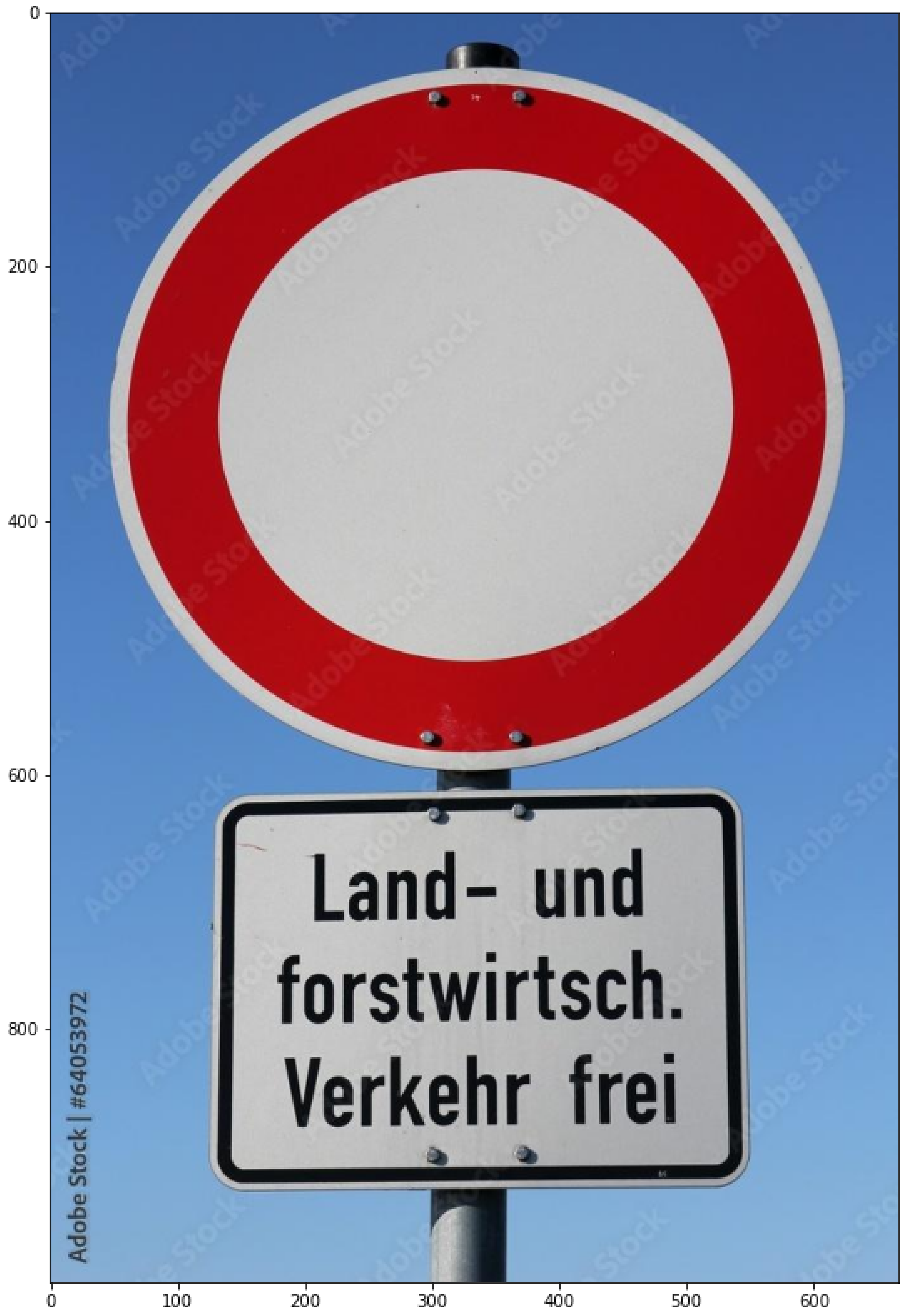
In [1]:
```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
import os
%matplotlib inline
```

In [2]:
```
def load(path):
    img=cv.imread(path)
    img=cv.cvtColor(img,cv.COLOR_BGR2RGB)
    return img
```

In [3]:
```
def display(img1,cmap="gray"):
    fig=plt.figure(figsize=(12,14))
    ax=fig.add_subplot()
    ax.imshow(img1,cmap="gray")
```

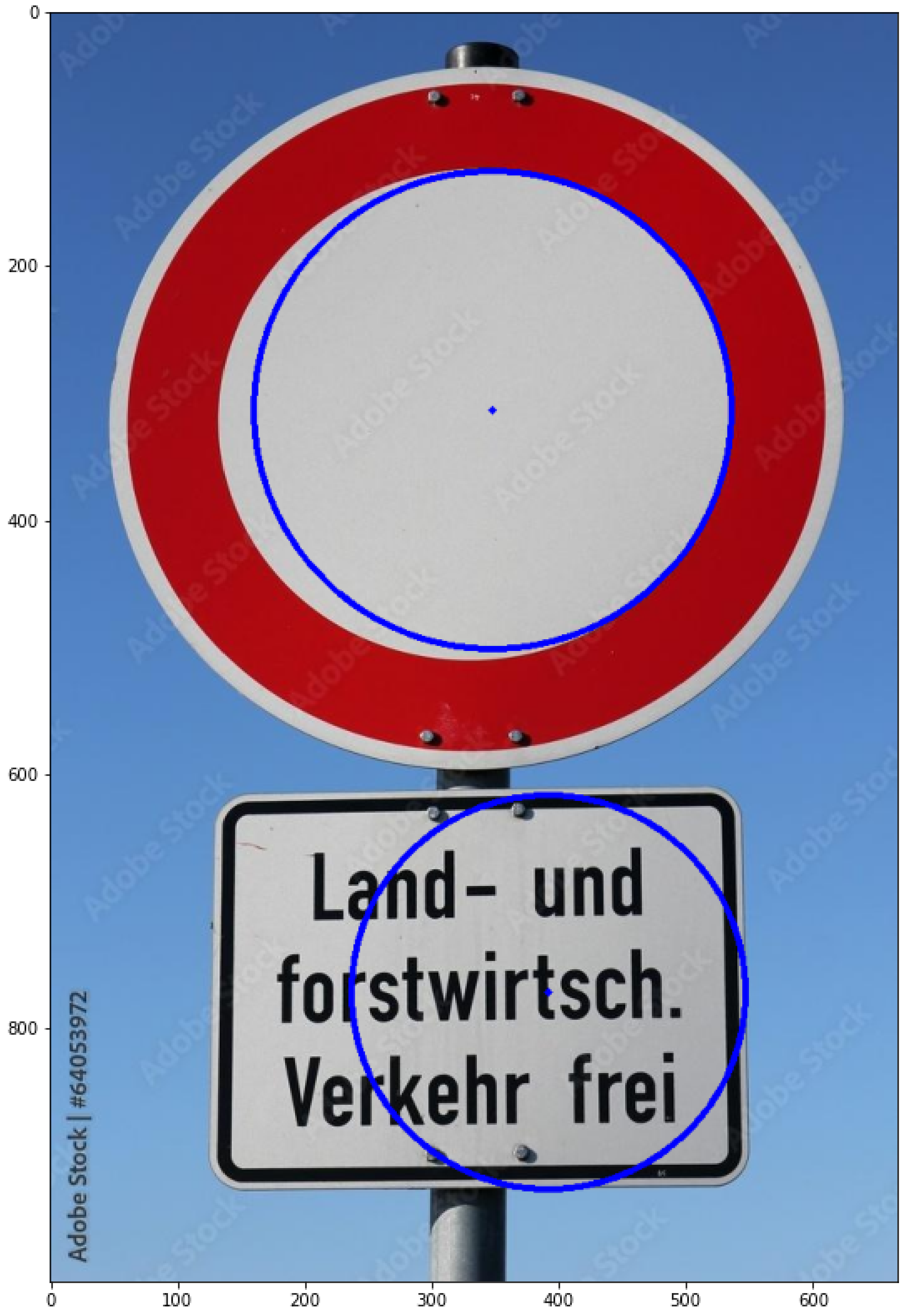
In [4]:
```
path="/Users/mehradhq/Computer_Vision/Research_2/dataset/train/Prohibition_Signs/45.jpeg"
img=load(path)
print ("this is our initial image")
display(img)
```

```
this is our initial image
```



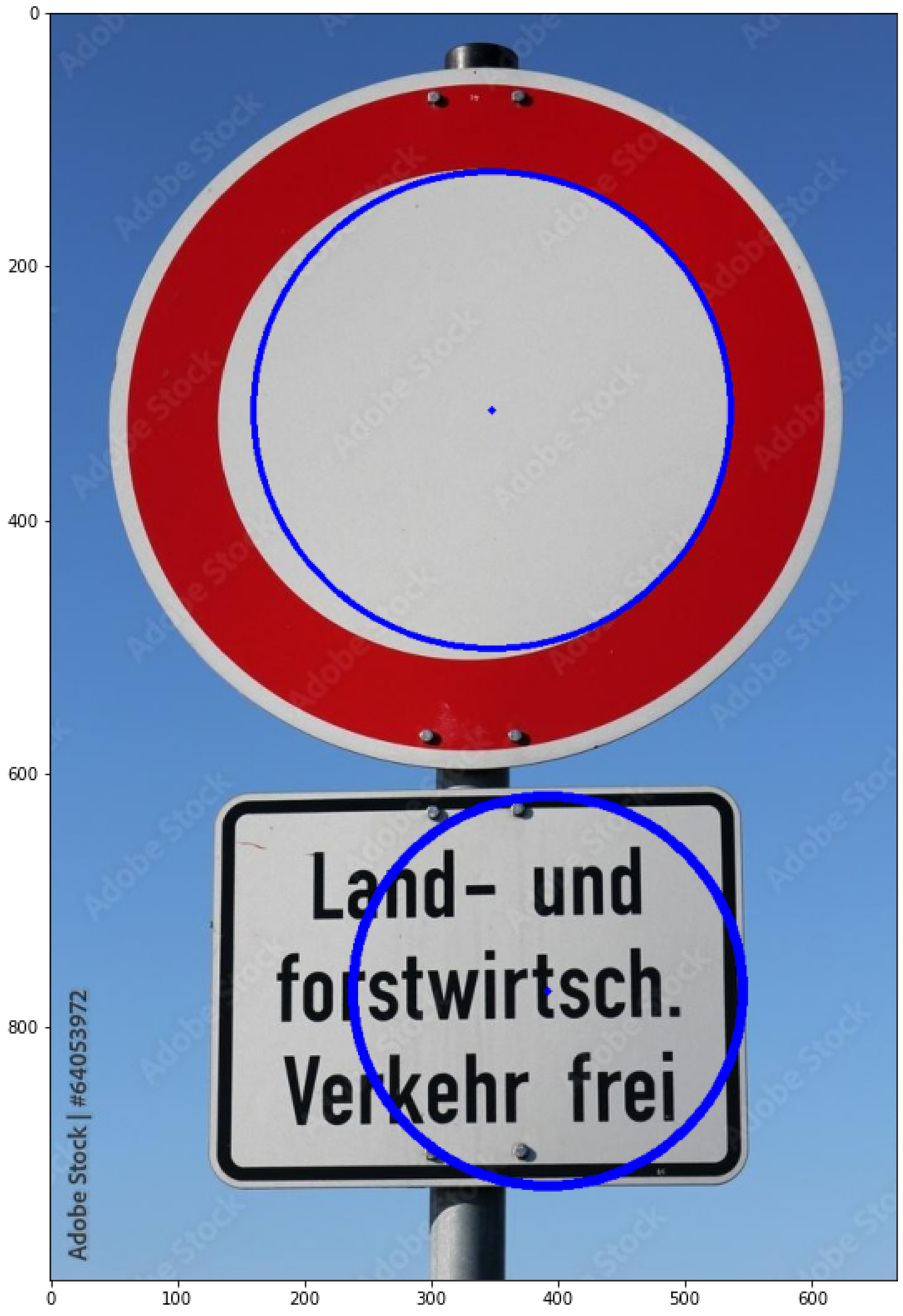
In [5]:

```
#hough tranformation only accepts gray-scale images
img_gray=cv.cvtColor(img,cv.COLOR_BGR2GRAY)
rows=img.shape[0]
circles = cv.HoughCircles(img_gray, cv.HOUGH_GRADIENT, 1, rows/6, param1=100, param2=50, minRadius=5

if circles is not None:
    circles = np.uint16(np.around(circles))
    for i in circles[0, :]:
        center = (i[0], i[1])
        # circle center
        cv.circle(img, center, 1, (0, 0, 255), 3)
        # circle outline
        radius = i[2]
        cv.circle(img, center, radius, (0, 0, 255), 3)

display(img)
```



In [6]:

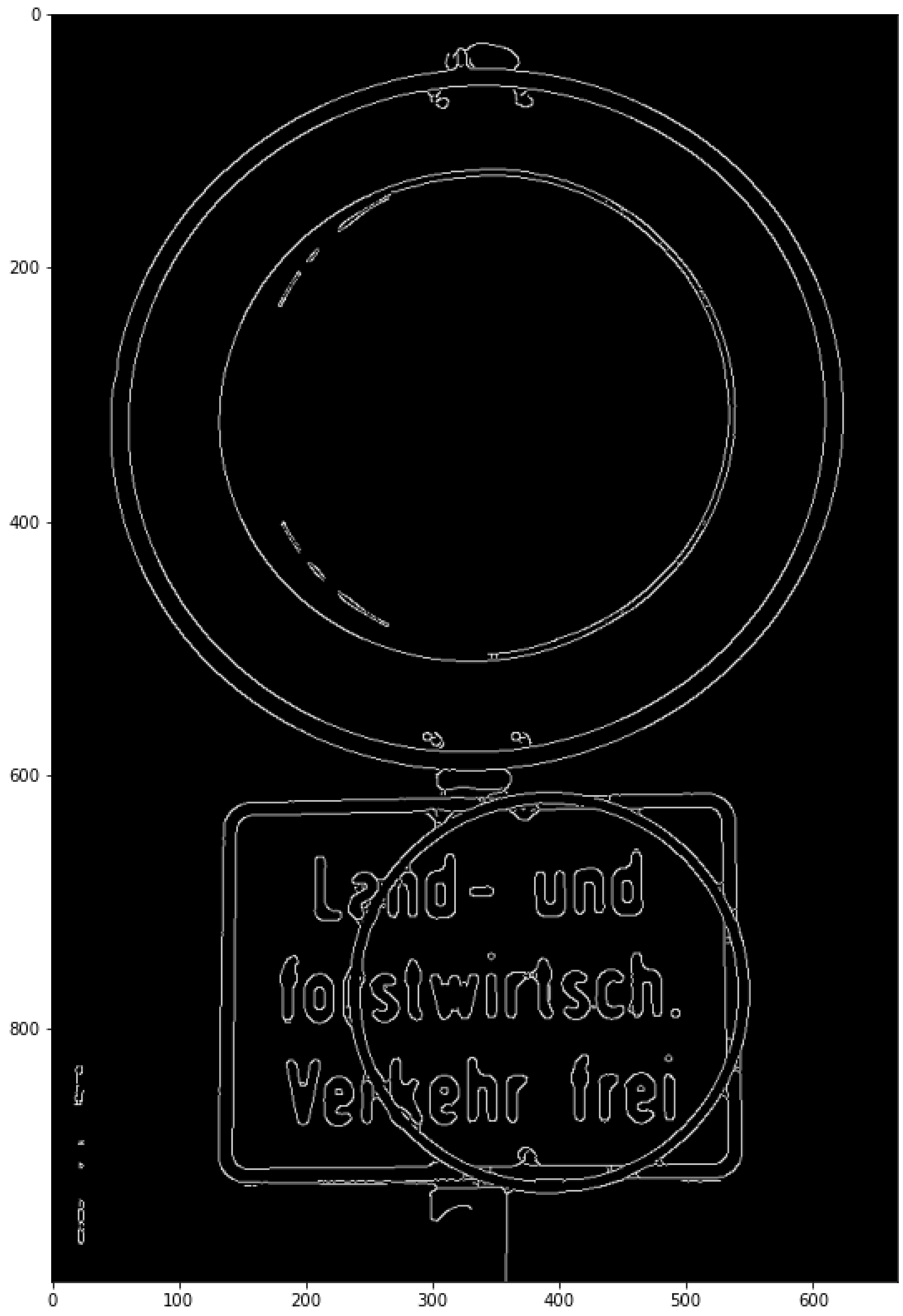
```
#hough tranformation only accepts gray-scale images
img_gray=cv.cvtColor(img,cv.COLOR_BGR2GRAY)
rows=img.shape[0]
circles = cv.HoughCircles(img_gray, cv.HOUGH_GRADIENT, 1, rows, param1=100, param2=50, minRadius=50,

if circles is not None:
    circles = np.uint16(np.around(circles))
    for i in circles[0, :]:
        center = (i[0], i[1])
        # circle center
        cv.circle(img, center, 1, (0, 0, 255), 3)
        # circle outline
        radius = i[2]
        cv.circle(img, center, radius, (0, 0, 255), 3)

display(img)
```



In [7]:

```
#median_blur
img_blur=cv.medianBlur(img,11)
#canny
med_val=np.median(img_blur)
lower=int(max(0,0.7*med_val))
```

```
upper=int(min(255,1.3*med_val))
edges=cv.Canny(img_blur,lower,upper)
display(edges)
```



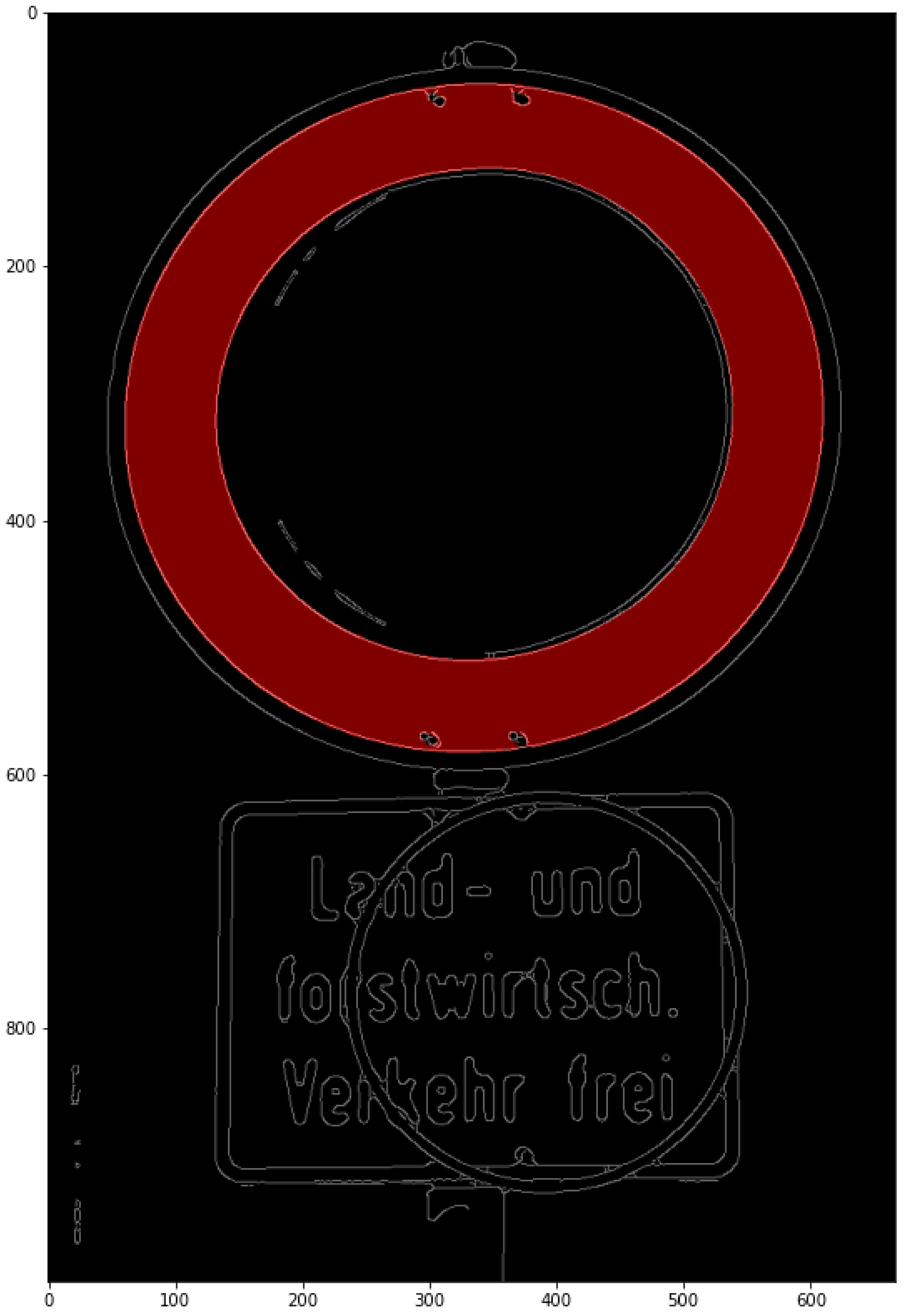
In [8]:

```
#inorder to blend to images the shapes must be equal so because our initial image has 3 channels, our edges must h
edges=cv.cvtColor(edges,cv.COLOR_GRAY2RGB)
```

In [9]:

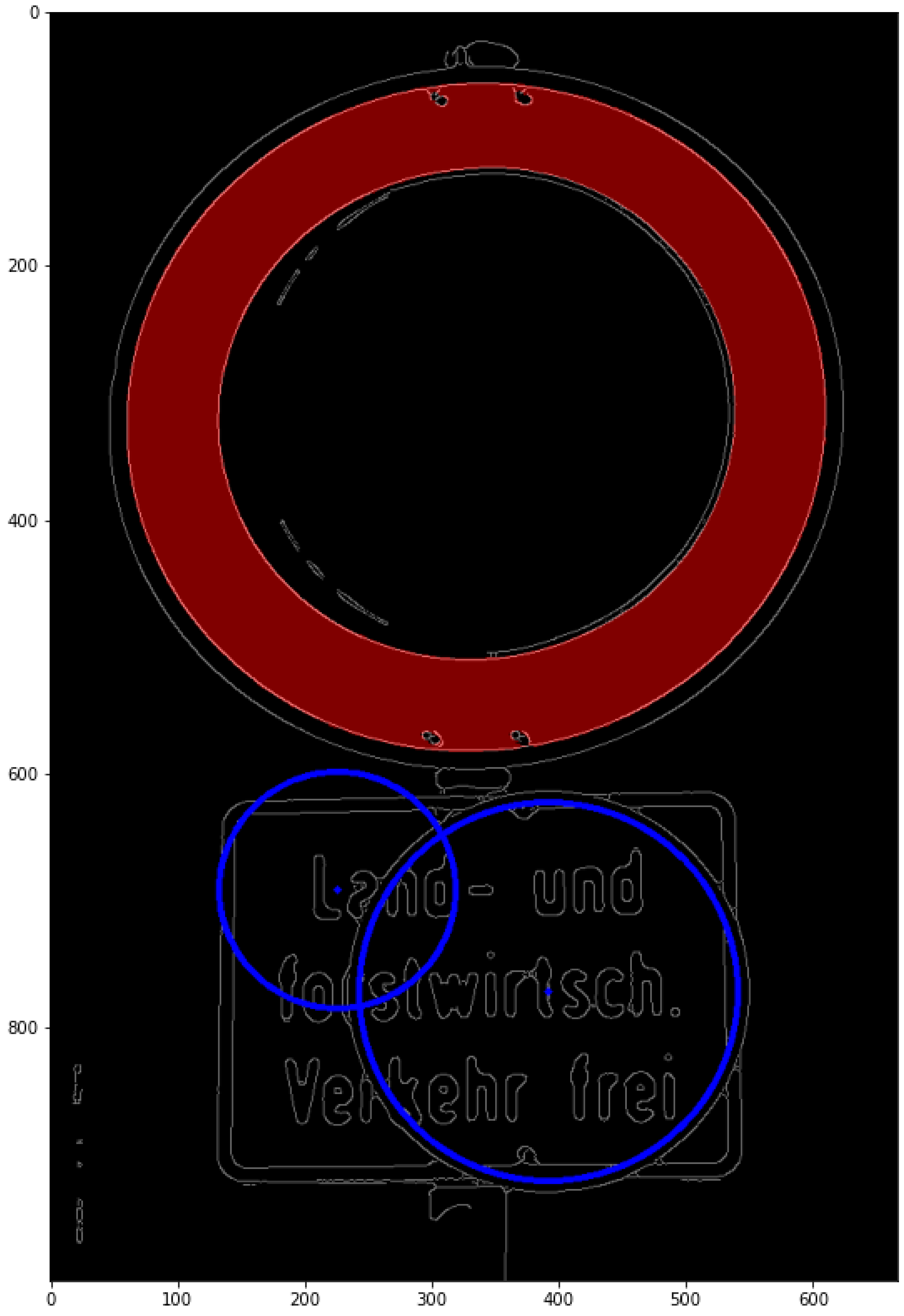
```
#masking
#In masking we first convert our image to HSV. Then given that the red channel in HSV contains hues from 0-10 and
#We get our ultimate mask by combining these two masks.
img_hsv=cv.cvtColor(img, cv.COLOR_RGB2HSV)
```

```
# lower mask (0-10)
lower_red = np.array([0,50,50])
upper_red = np.array([10,255,255])
mask0 = cv.inRange(img_hsv, lower_red, upper_red)

# upper mask (170-180)
lower_red = np.array([170,50,50])
upper_red = np.array([180,255,255])
mask1 = cv.inRange(img_hsv, lower_red, upper_red)

# join my masks
mask = mask0+mask1

# set my output img to zero everywhere except my mask
output_img = img.copy()
output_img[np.where(mask==0)] = [0,0,0]
output_img[np.where(mask!=0)] = [255,0,0]

#plt.imshow(output_img)
#use blending for each image to get the red parts with a more emphasis in the initial image.
blended=cv.addWeighted(output_img,0.5,edges,0.5,0)
display(blended)
```



In [10]:

```
#hough tranformation on the masked image
img_gray=cv.cvtColor(blended,cv.COLOR_BGR2GRAY)
rows=blended.shape[0]
circles = cv.HoughCircles(img_gray, cv.HOUGH_GRADIENT, 1, rows/6, param1=100, param2=50, minRadius=5

if circles is not None:
    circles = np.uint16(np.around(circles))
    for i in circles[0, :]:
        center = (i[0], i[1])
        # circle center
        cv.circle(blended, center, 1, (0, 0, 255), 3)
        # circle outline
        radius = i[2]
    cv.circle(blended, center, radius, (0, 0, 255), 3)

display(blended)
```


Land- und
forstwirtsch.
Verkehr frei