

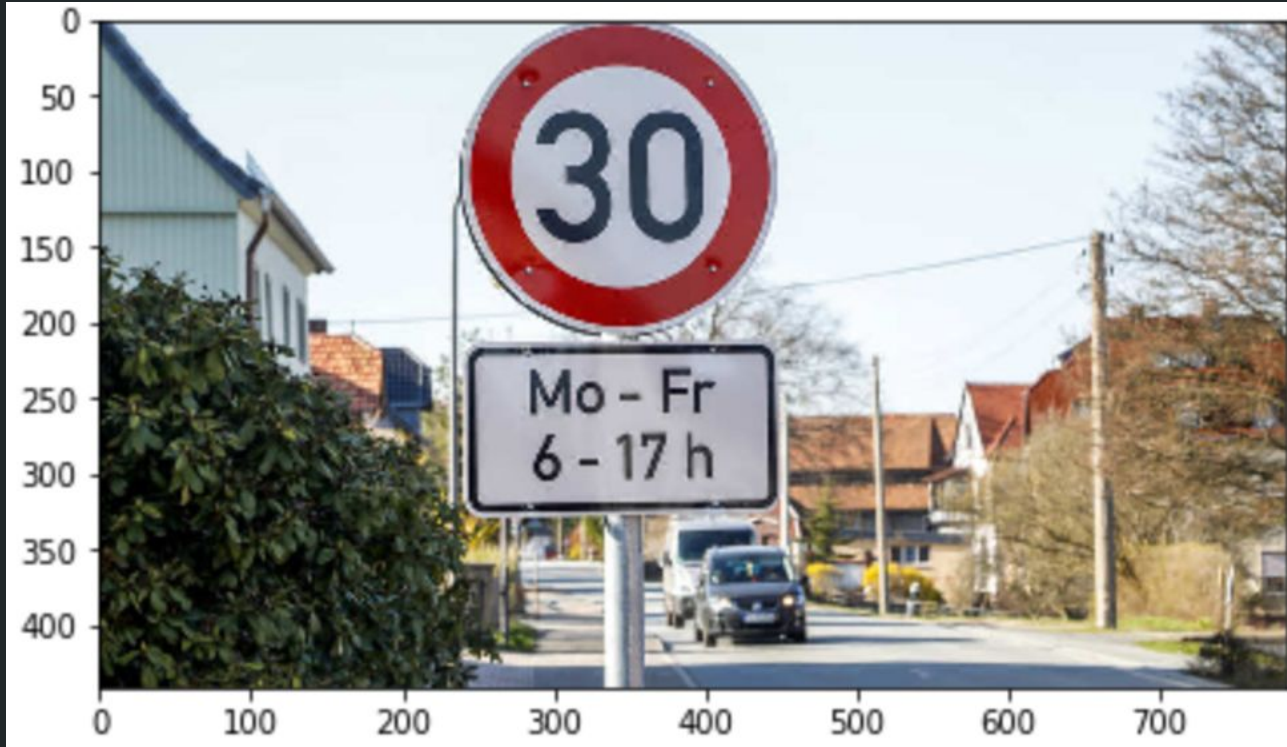
Verkehrsschilderkennung

Alina Göttig, Leon Budimovic & Alexander-Robert Keller

Inhalt

- Potentielle Schilder herausfiltern:
 - Mit dem `cv2.SimpleBlobDetektor`
 - Mit der Contour und `cv2.fitEllipse`
- Klassifizierung der potentiellen Schilder:
 - Matching mithilfe des Spektrums
 - Mit Template Matching
 - Kombination

Bild einlesen



Kantendetektionsversuche

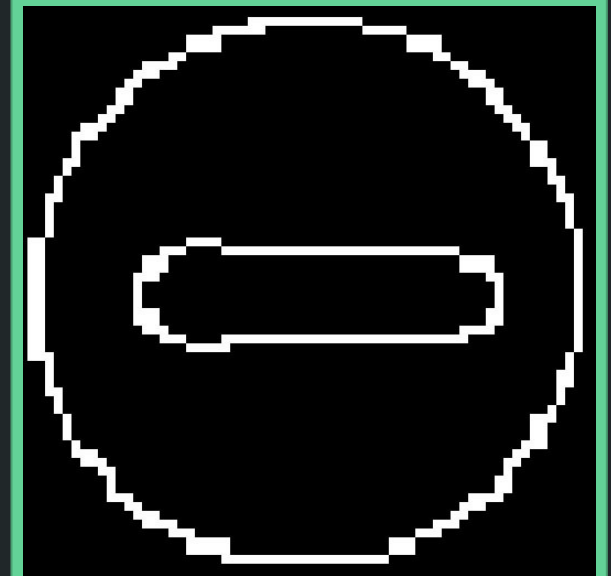
Bild



1. Gradient



2. Canny



Gradient

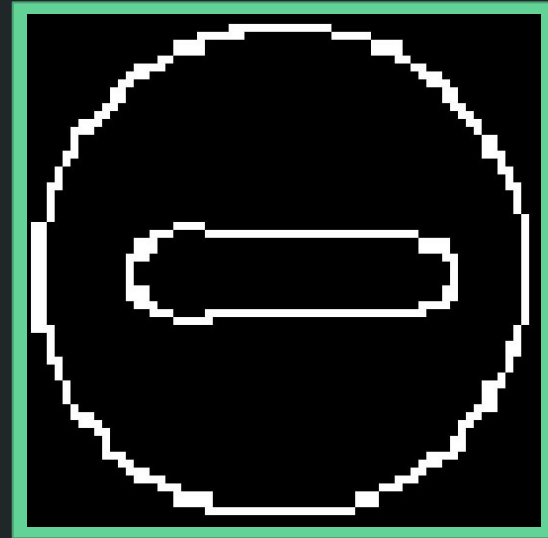


Canny-Kantendetektion

Versuch 1:



Final:



Sigma: 3.0

Canny-Kantendetektion



1x Dilate

— $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ →

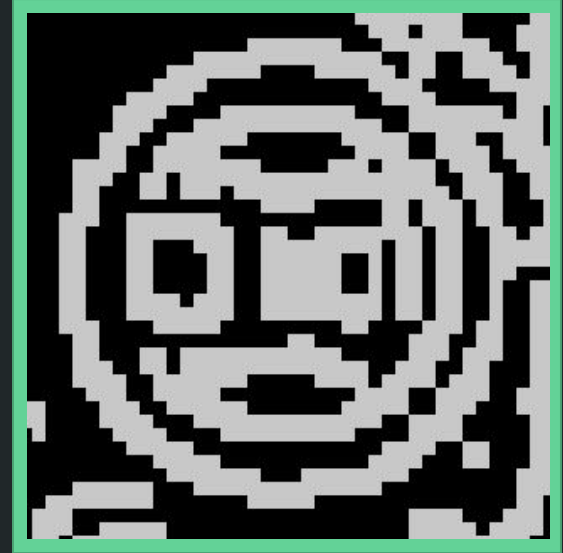


Canny-Kantendetektion



1x Dilate

$$\begin{array}{c} [1 \ 1 \ 1] \\ [1 \ 1 \ 1] \\ [1 \ 1 \ 1] \end{array} \rightarrow$$



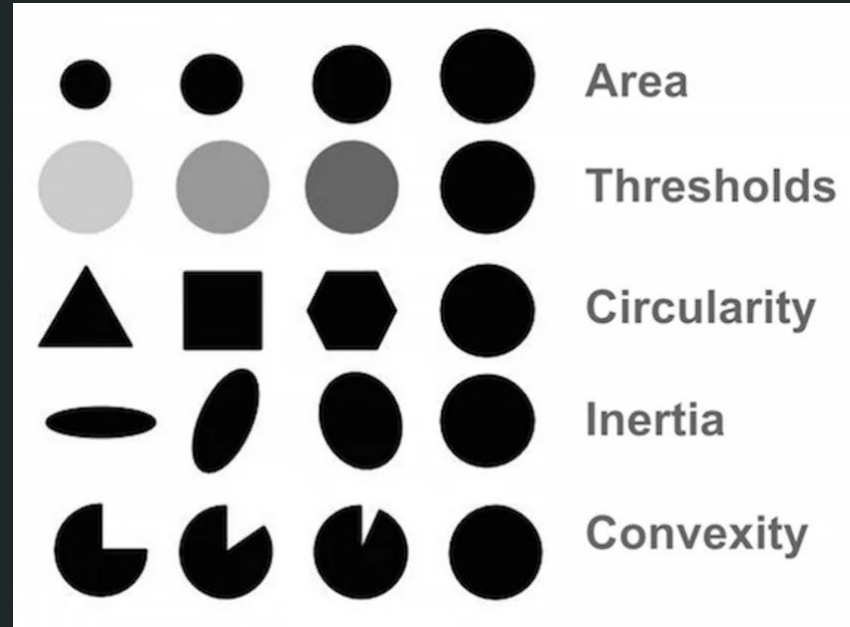
Erkennung von Formen mit dem **OpenCV SimpleBlobDetector**

```
p = cv2.SimpleBlobDetector_Params()
```

```
d = cv2.SimpleBlobDetector_create(p)
```

```
# liefert Koordinaten & Durchmesser
```

```
blobs = d.detect( image )
```



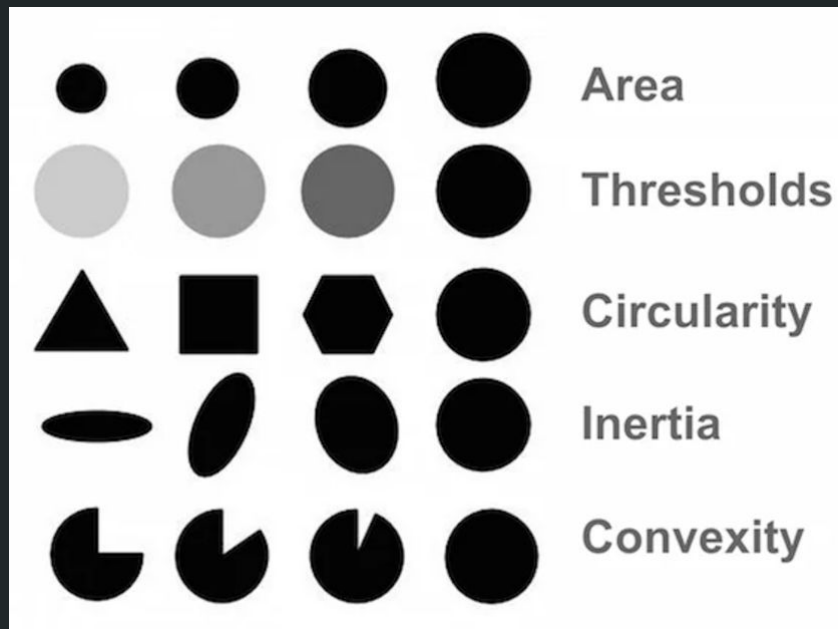
SimpleBlobDetector Codebeispiel

```
params = cv2.SimpleBlobDetector_Params()

params.filterByColor = False
params.filterByArea = True
params.minArea = minArea #20000 #1000
params.maxArea = 100000000
params.filterByCircularity = True
params.minCircularity = minCircularity
params.maxCircularity = maxCircularity
params.filterByConvexity = True
params.minConvexity = 0.7
params.filterByInertia = True
#params.minInertiaRatio = 0.8
params.minInertiaRatio = 0.3
params.minThreshold = 1
params.maxThreshold = 255

detector = cv2.SimpleBlobDetector_create(params)

keypoints = detector.detect(img)
```



Erkennung von Kreisen/Ellipsen

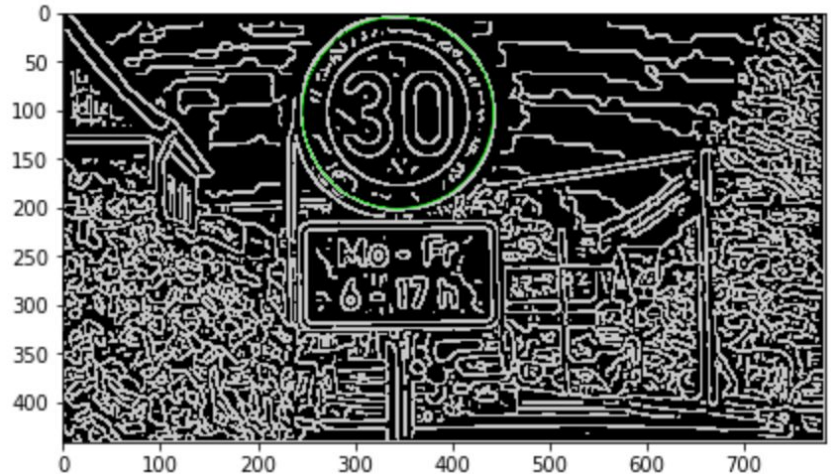
Gradient

Number of Circular Blobs: 1



Canny

Number of Circular Blobs: 1



Mehrfache Erkennung



Durchgang verboten
Ø 280 mm



Richtung links oder
Richtung rechts 280 mm



Baustellen 380 mm



Richtung rechts/unten o.
Richtung links/unten 280 mm



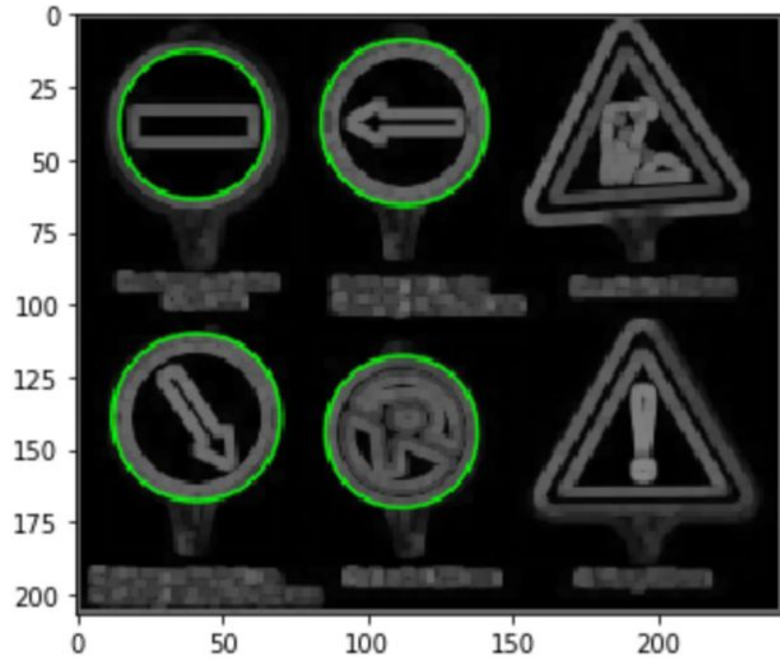
Parkverbot 280 mm



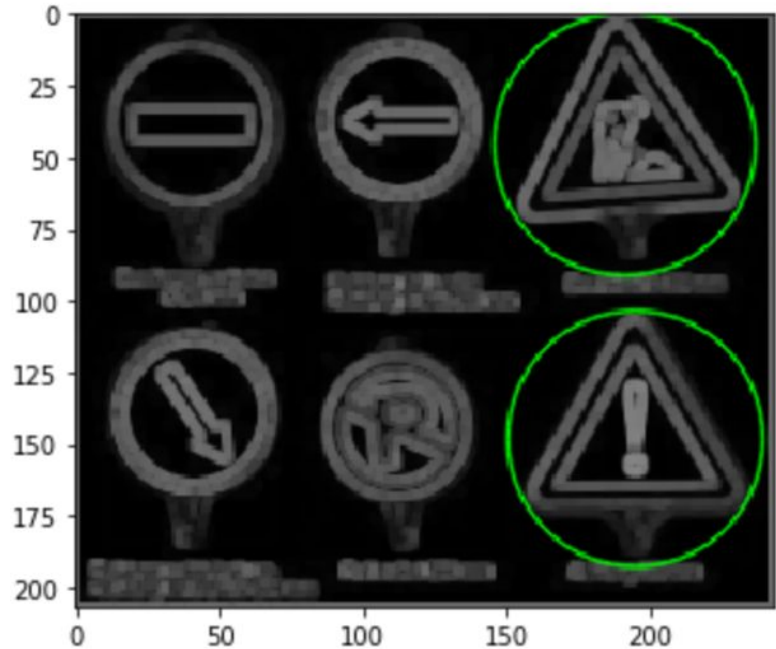
Achtung 380 mm

Mehrfache Erkennung (Gradient)

Number of Circular Blobs: 4

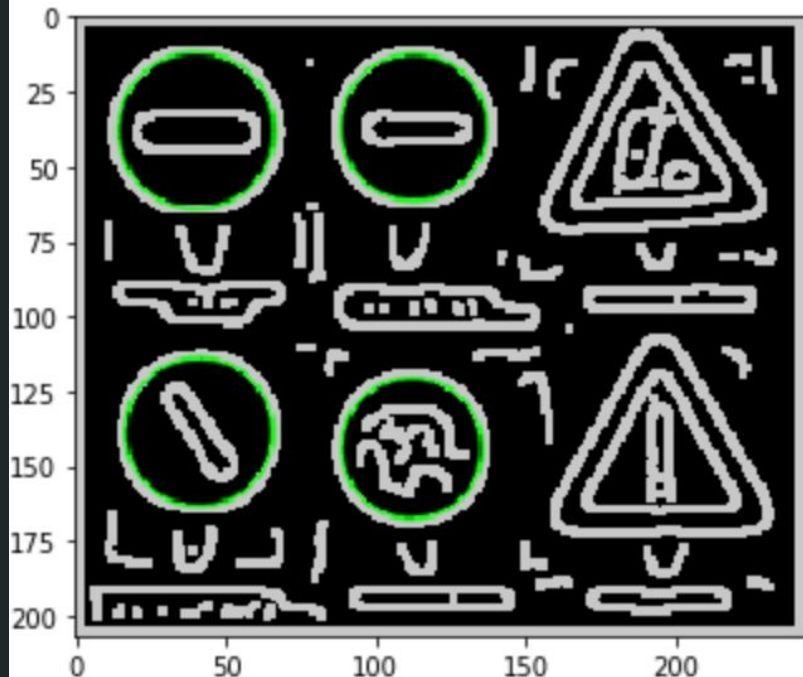


Number of Triangular Blobs: 2



Mehrfache Erkennung (Canny)

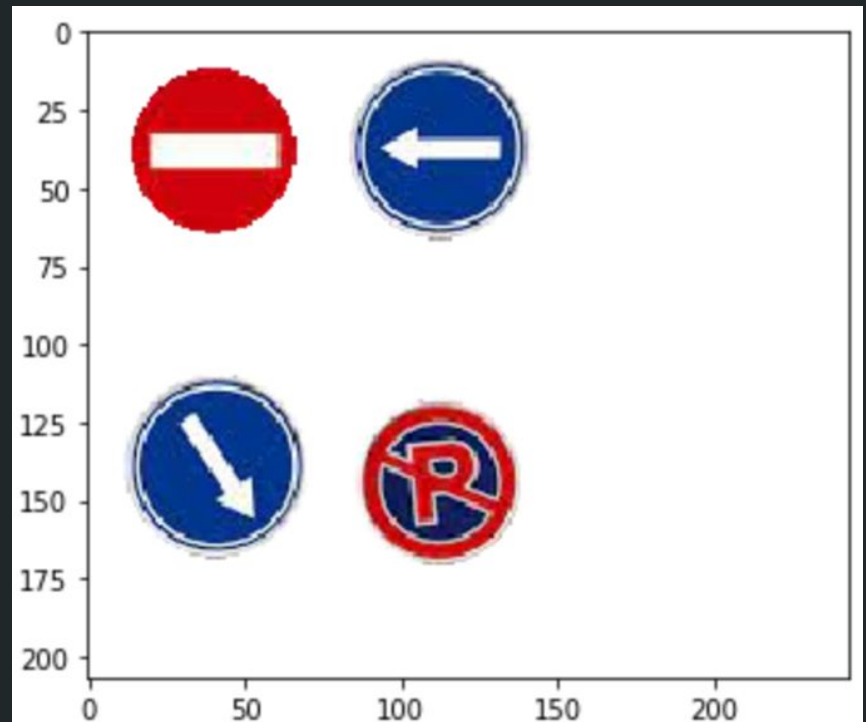
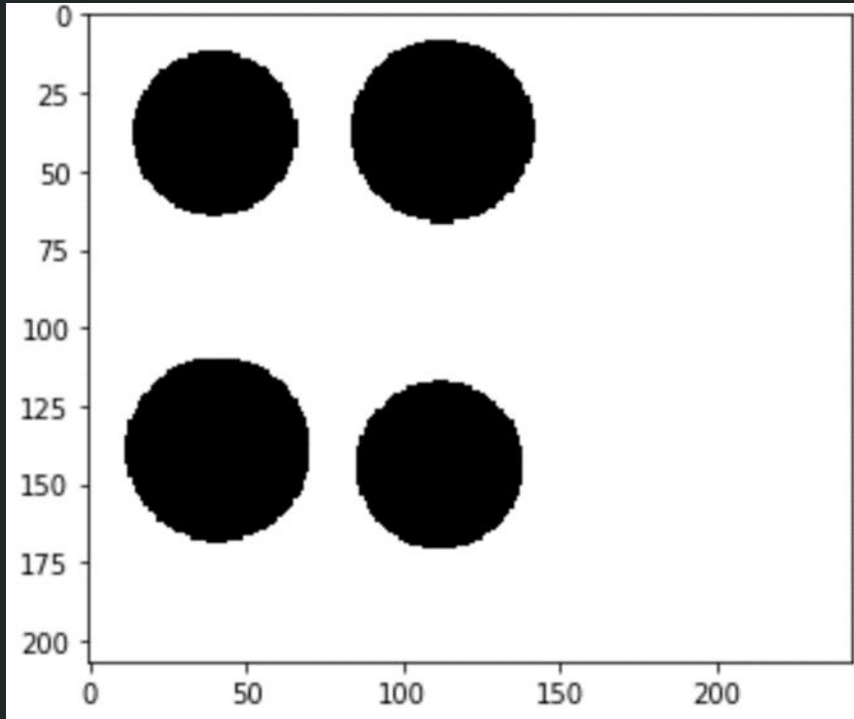
Number of Circular Blobs: 4



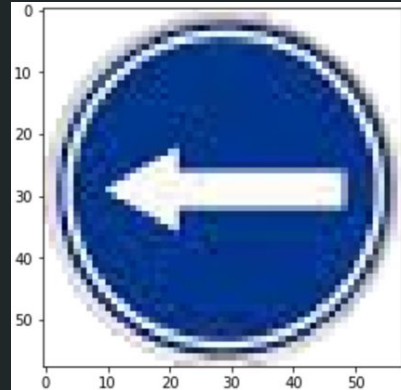
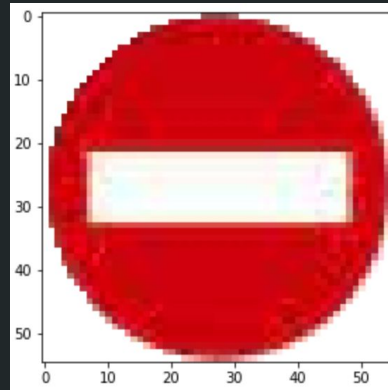
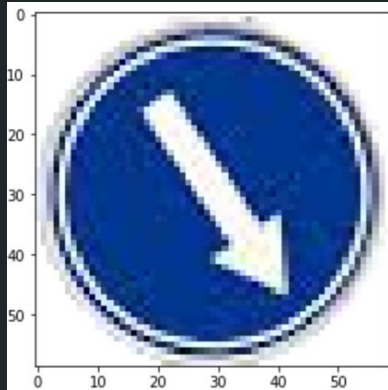
Number of Triangular Blobs: 2



Blobs aus dem Hintergrund ausschneiden



Erkannte Blobs in eigene Images

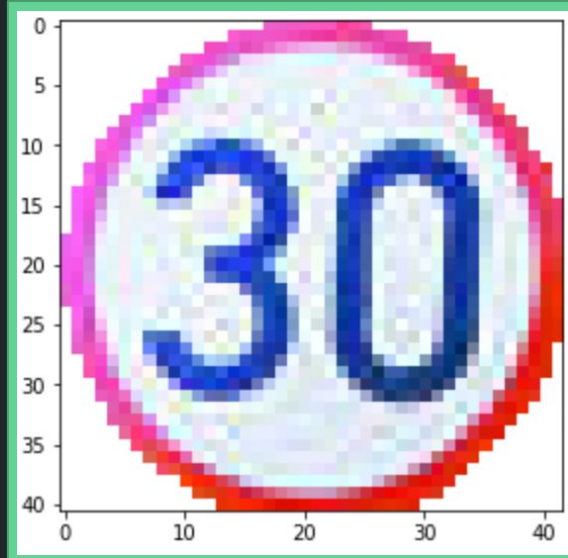


Vergleich

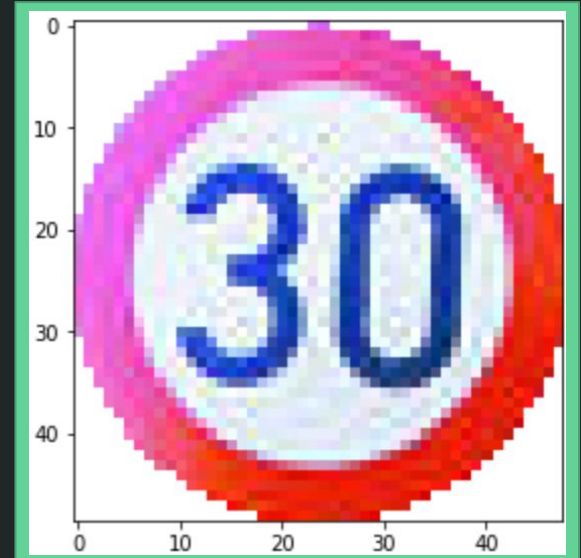
Bild



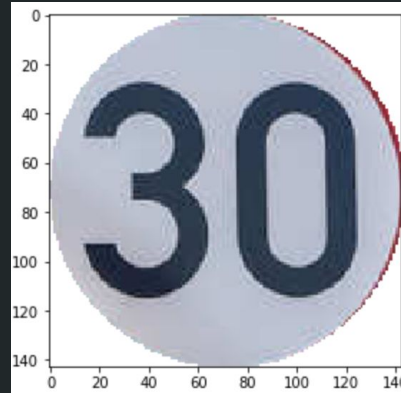
1. Gradient



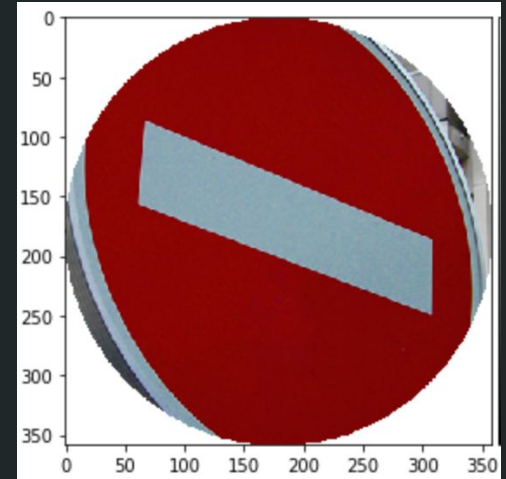
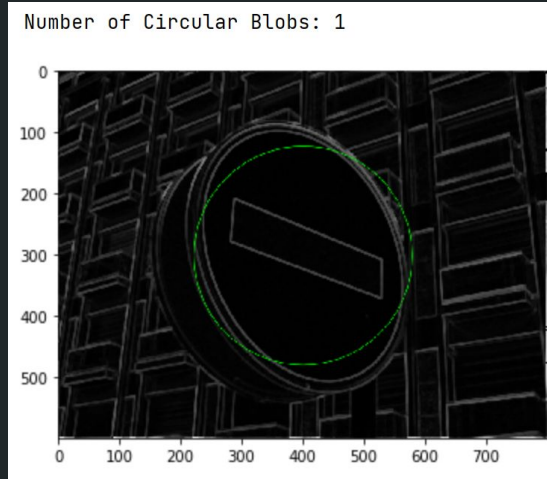
2. Canny



Schwierigkeit: Falsche Detektion

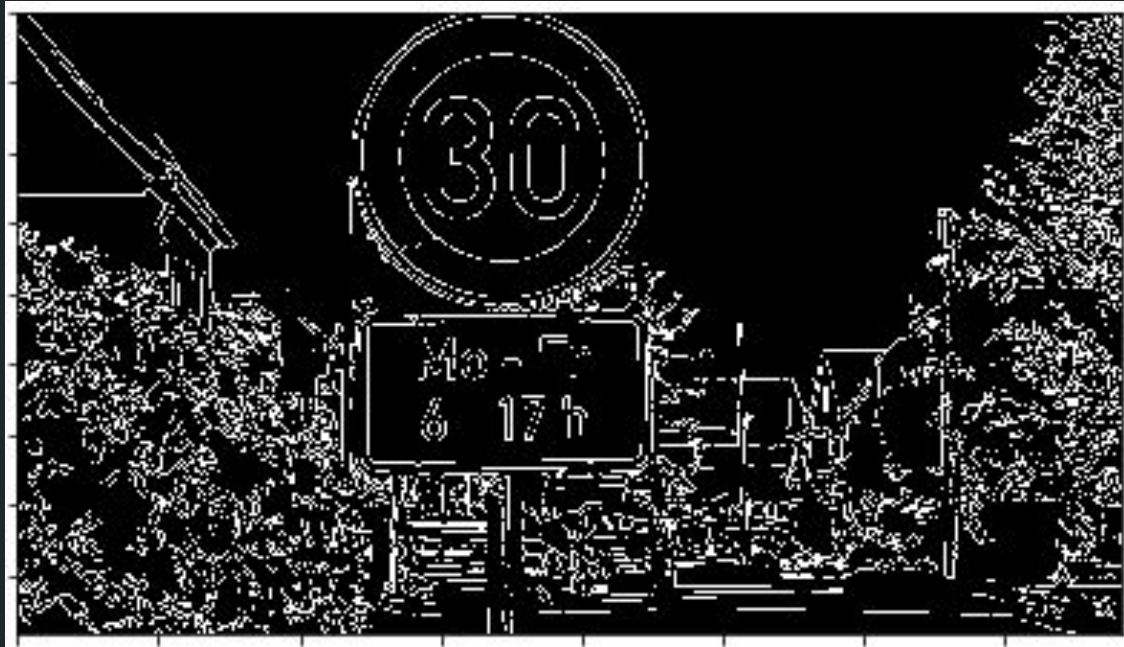


Schwierigkeit: Verzerrung



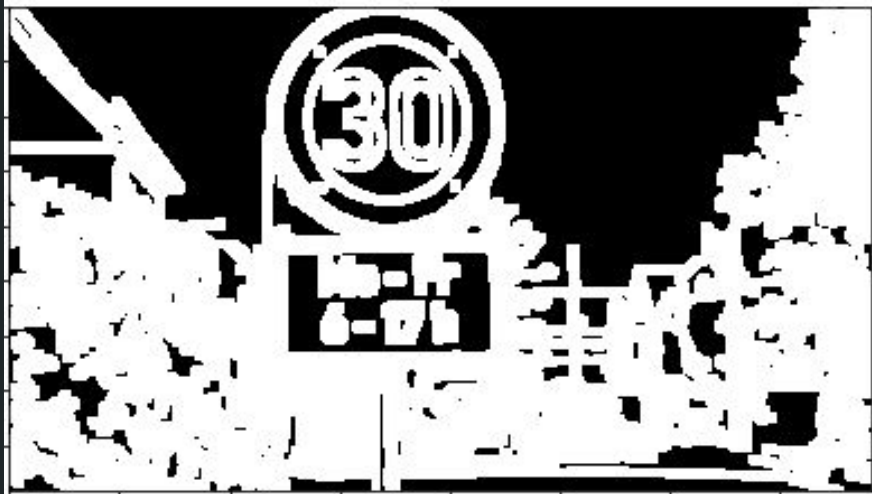
Erkennung der Bildregion über Konturen

Canny



Erkennung der Bildregion über Konturen

Dilate

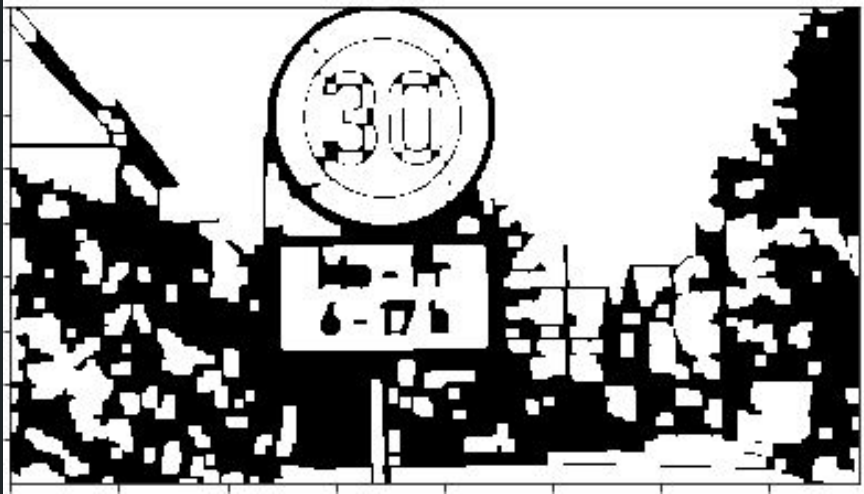


Erode



Erkennung der Bildregion über Konturen

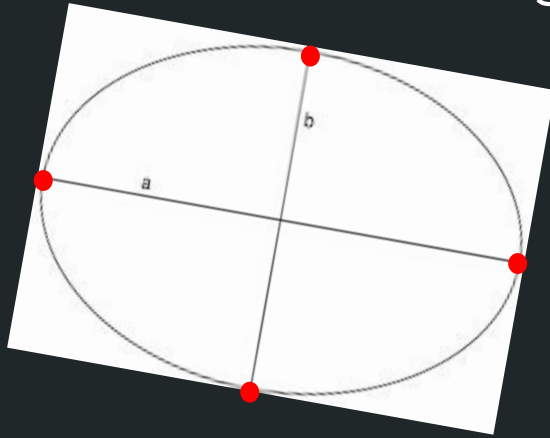
Invert



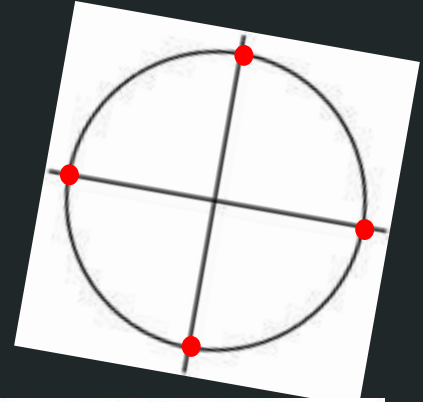
find Contours and fit Ellipse



Erkennung der Bildregion über Konturen



Perspective Transform



Perspective Transform



Erkennung der Bildregion über Konturen



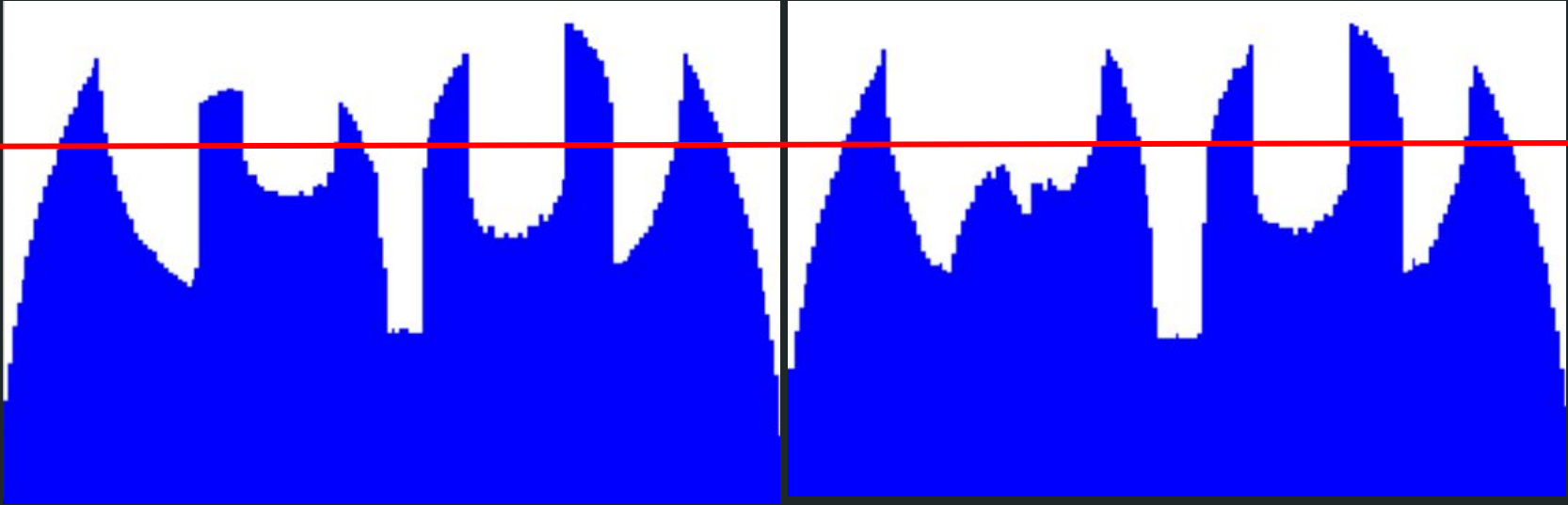
Perspective Transform



Matching mithilfe des Spektrums



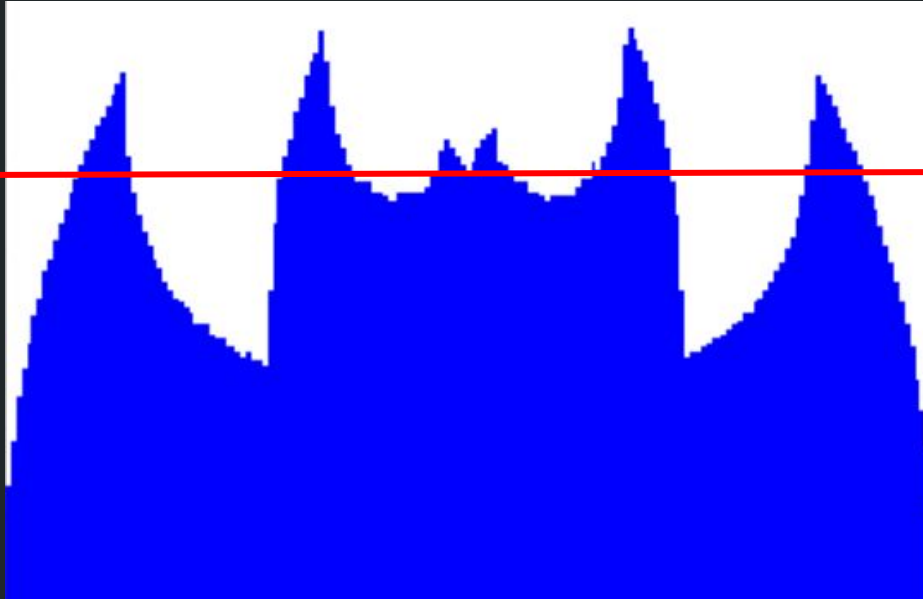
Vergleich durch Thresholding



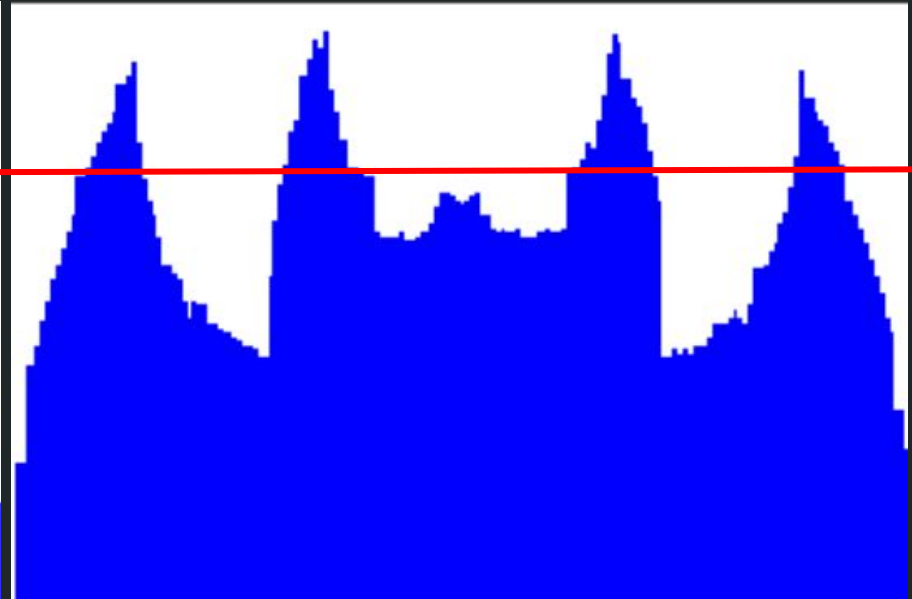
50er-Schild

30er-Schild

Schwierigkeiten



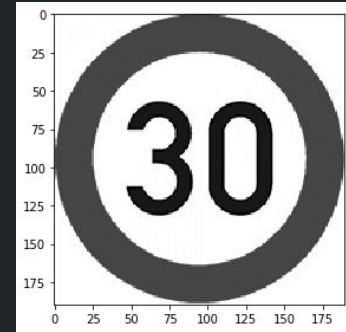
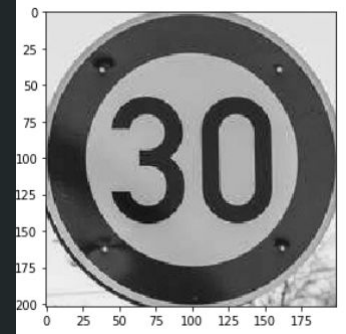
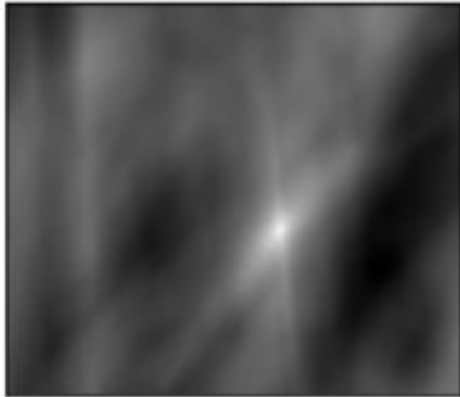
80er-Schild



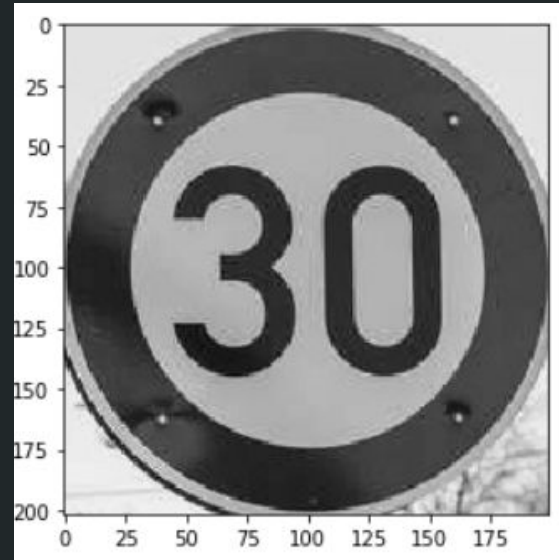
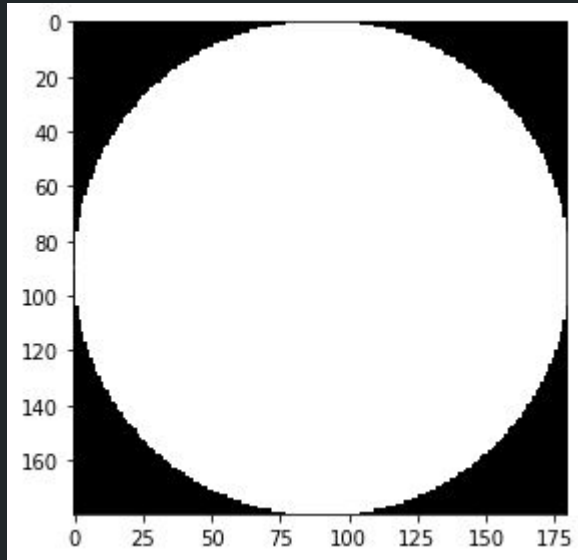
30er-Schild

Matching per Template Matching

cv2.TM_CCOEFF_NORMED

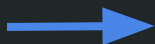


Matching mit Maske

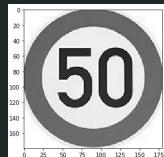
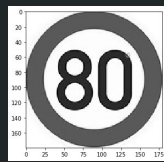
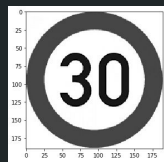
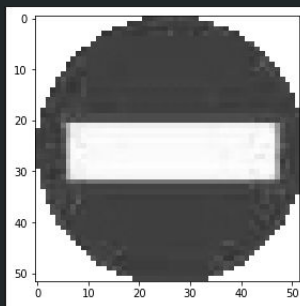
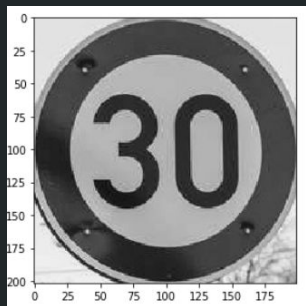


Matching Ergebnisse

Images



Templates



0.77

0.21

0.72

0.66

0.83

0.18

0.77

0.76

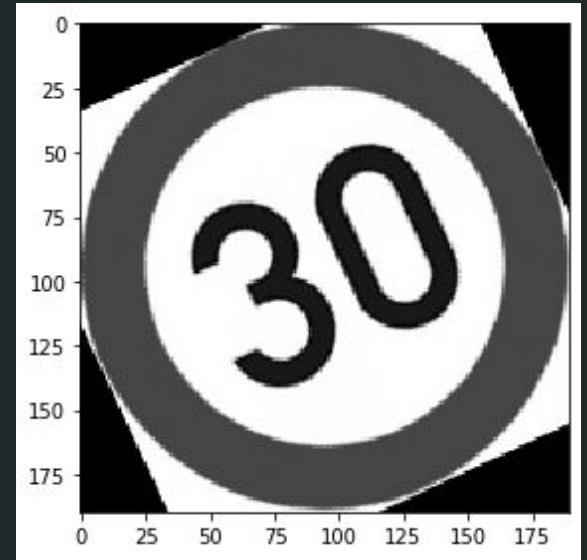
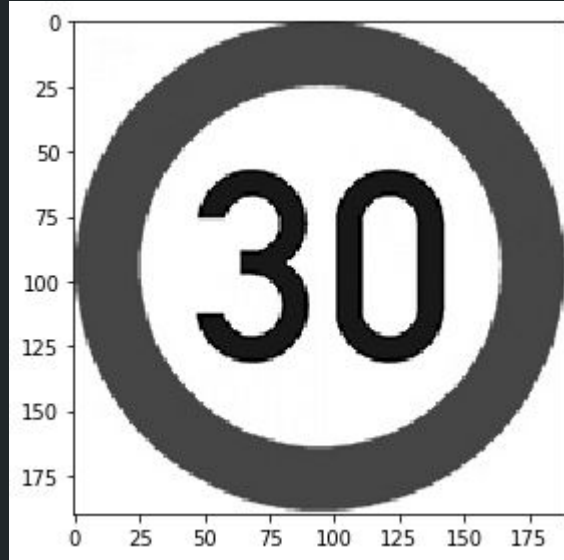
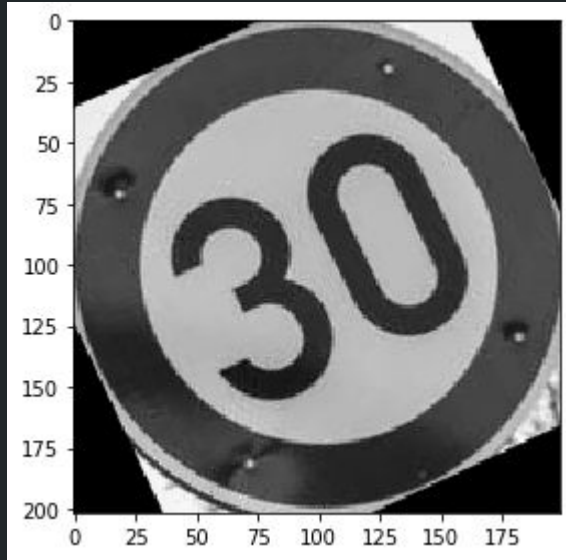
0.74

0.18

0.67

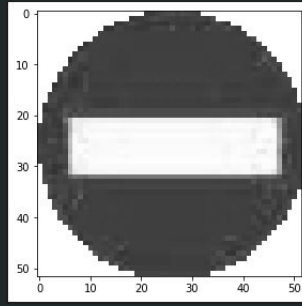
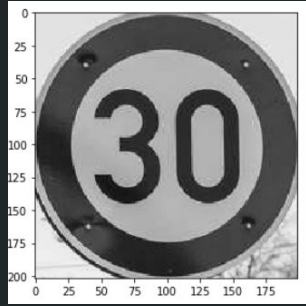
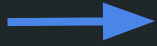
0.89

Rotations Problem

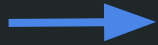


Matching Ergebnisse

Images



Erkanntes



Schild:

30er

No

80er

50er