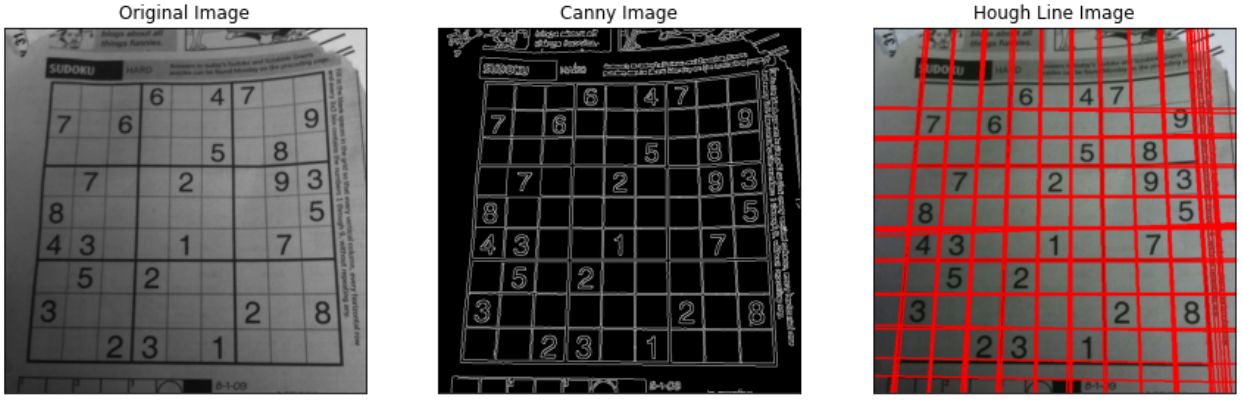
Name : Jegakumaran P.

Index Number : 190280N

In [ ]:
```
import cv2 as cv
import numpy as np
img=cv.imread(r'Images/sudoku.png',cv.IMREAD_COLOR)
assert img is not None

gray=cv.cvtColor(img,cv.COLOR_BGR2GRAY)
edges=cv.Canny(gray,20,120,apertureSize=3)
lines=cv.HoughLines(edges,1,np.pi/180,175)

for line in lines:
    rho,theta=line[0]
    a=np.cos(theta)
    b=np.sin(theta)
    x0,y0=a*rho,b*rho
    x1,y1=int(x0+1000*(-b)),int(y0+1000*(a))
    x2,y2=int(x0-1000*(-b)),int(y0-1000*(a))
    cv.line(img,(x1,y1),(x2,y2),(0,0,255),2)

fig,ax=plt.subplots(1,3,sharex='all',sharey='all',figsize=(15,15))
ax[0].imshow(cv.cvtColor(gray,cv.COLOR_BGR2RGB))
ax[0].set_title("Original Image")
ax[0].set_xticks([]) , ax[0].set_yticks([])
ax[1].imshow(cv.cvtColor(edges,cv.COLOR_BGR2RGB))
ax[1].set_title("Canny Image")
ax[1].set_xticks([]) , ax[1].set_yticks([])
ax[2].imshow(cv.cvtColor(img,cv.COLOR_BGR2RGB))
ax[2].set_title("Hough Line Image")
ax[2].set_xticks([]) , ax[2].set_yticks([])
```

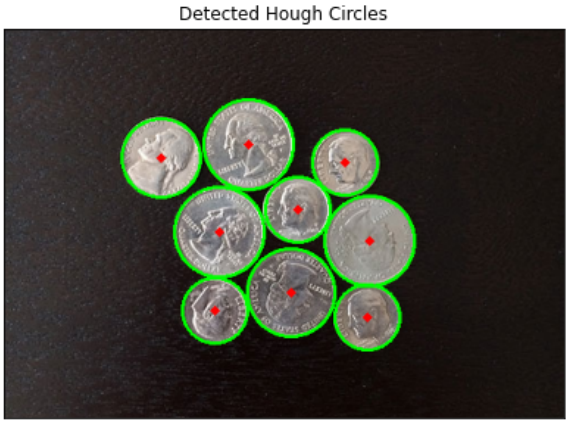
Out[ ]:
```
([], [])
```



In [ ]:
```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
img=cv.imread(r'Images/coins.jpg',cv.IMREAD_COLOR)
assert img is not None
fig,ax=plt.subplots(1,2,sharex='all',sharey='all',figsize=(15,15))
ax[0].imshow(cv.cvtColor(img,cv.COLOR_BGR2RGB))
ax[0].set_title("Original Image")
ax[0].set_xticks([]) , ax[0].set_yticks([])
gray=cv.cvtColor(img,cv.COLOR_BGR2GRAY)

circles=cv.HoughCircles(gray,cv.HOUGH_GRADIENT,1,50,param1=200,param2=65,minRadius=15,maxRadius
circles=np.uint16(np.around(circles))
for i in circles[0,:]:
```

```
    cv.circle(img,(i[0],i[1]),i[2],(0,255,0),2) #Draw the outer circle
    cv.circle(img,(i[0],i[1]),2,(0,0,255),3) #Draw the center of the circle

ax[1].imshow(cv.cvtColor(img,cv.COLOR_BGR2RGB))
ax[1].set_title("Detected Hough Circles")
ax[1].set_xticks([]) , ax[1].set_yticks([])
```

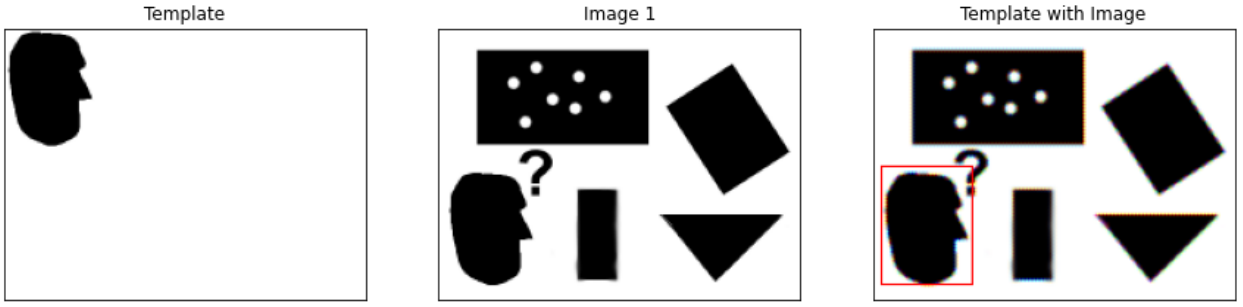
Out[ ]:
```
([], [])
```



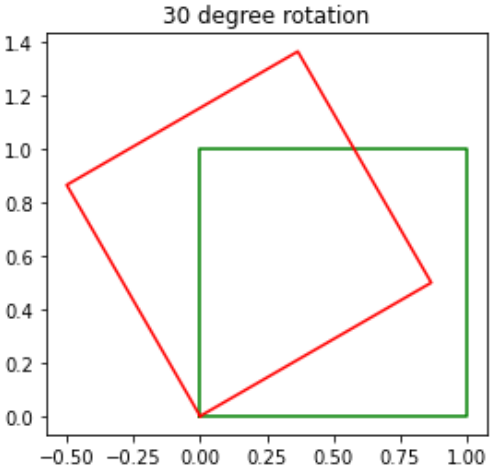
In [ ]:
```
import cv2 as cv
import numpy as np

img=cv.imread(r'Images/pic1.png',cv.IMREAD_REDUCED_GRAYSCALE_2)
temp1=cv.imread(r'Images/templ.png',cv.IMREAD_REDUCED_GRAYSCALE_2)
assert img is not None
assert temp1 is not None

img_edges=cv.Canny(img,50,250)
temp1_edges=cv.Canny(temp1,50,250)
alg=cv.createGeneralizedHoughGuil()
alg.setTemplate(temp1_edges)
alg.setAngleThresh(100000)
alg.setScaleThresh(40000)
alg.setPosThresh(1000)
alg.setAngleStep(1)
alg.setScaleStep(0.1)
alg.setMinScale(0.9)
alg.setMaxScale(1.1)
positions,votes=alg.detect(img_edges)
out=cv.cvtColor(img,cv.COLOR_BAYER_BG2BGR)
for x,y,scale,orientation in positions[0]:
    halfHeight=temp1.shape[0]/2.*scale
    halfWidth=temp1.shape[1]/2.*scale
    p1=(int(x-halfWidth),int(y-halfHeight))
    p2=(int(x+halfWidth),int(y+halfHeight))
    print("x = {}, y = {}, scale = {}, orientation = {}, p1 = {}, p2 = {}".format(x,y,scale,ori
    cv.rectangle(out,p1,p2,(0,0,255))

fig,ax=plt.subplots(1,3,sharex='all',sharey='all',figsize=(15,15))
ax[0].imshow(cv.cvtColor(temp1,cv.COLOR_BGR2RGB))
ax[0].set_title("Template")
ax[0].set_xticks([]) , ax[0].set_yticks([])
ax[1].imshow(cv.cvtColor(img,cv.COLOR_BGR2RGB))
ax[1].set_title("Image 1")
ax[1].set_xticks([]) , ax[1].set_yticks([])
ax[2].imshow(cv.cvtColor(out,cv.COLOR_BGR2RGB))
ax[2].set_title("Template with Image")
ax[2].set_xticks([]) , ax[2].set_yticks([])
```

```
x = 29.0, y = 109.0, scale = 1.0, orientation = 0.0, p1 = (4, 76), p2 = (54, 141)
```

Out[ ]: ([], [])

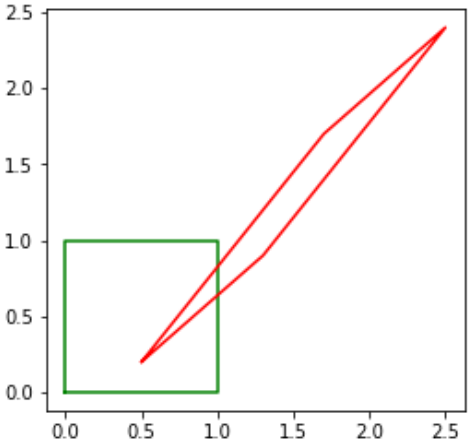


In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

a,b,c,d=[0,0,1],[0,1,1],[1,1,1],[1,0,1]
X=np.array([a,b,c,d]).T

theta=np.pi*30/180
s=1
tx,ty=0,0
H=np.array([[s*np.cos(theta),-s*np.sin(theta),tx],[s*np.sin(theta),s*np.cos(theta),ty],[0,0,1]]
Y=H@X
#a11,a12,a21,a22=0.8,1.2,0.7,1.5
#A=np.array([[a11,a12,tx],[a21,a22,ty],[0,0,1]])
#Y=A@X

x=np.append(X[0,:],X[0,0])
y=np.append(X[1,:],X[1,0])
fig,ax=plt.subplots(1,1)
ax.set_title("30 degree rotation")
ax.plot(x,y,color='g')
ax.set_aspect('equal')

x=np.append(Y[0,:],Y[0,0])
y=np.append(Y[1,:],Y[1,0])
ax.plot(x,y,color='r')
ax.set_aspect('equal')
plt.show()
```



In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

a,b,c,d=[0,0,1],[0,1,1],[1,1,1],[1,0,1]
X=np.array([a,b,c,d]).T
```

```
theta=np.pi*30/180
s=2
tx,ty=0.5,0.2
a11,a12,a21,a22=0.8,1.2,0.7,1.5
A=np.array([[a11,a12,tx],[a21,a22,ty],[0,0,1]])
Y=A@X

x=np.append(X[0,:],X[0,0])
y=np.append(X[1,:],X[1,0])
fig,ax=plt.subplots(1,1)
ax.set_title("")
ax.plot(x,y,color='g')
ax.set_aspect('equal')

x=np.append(Y[0,:],Y[0,0])
y=np.append(Y[1,:],Y[1,0])
ax.plot(x,y,color='r')
ax.set_aspect('equal')
plt.show()
```



In [ ]:
```
# Warping using the given homography
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt

img1=cv.imread(r"Images/graf/img1.ppm",cv.IMREAD_ANYCOLOR)
img4=cv.imread(r"Images/graf/img4.ppm",cv.IMREAD_ANYCOLOR)
H=[]
with open(r"Images/graf/H1to4p") as f:
    H=np.array([[float(h) for h in line.split()] for line in f])
im1to4=cv.warpPerspective(img4,np.linalg.inv(H),(2000,2000))

fig,ax=plt.subplots(1,3,figsize=(15,15))
ax[0].imshow(cv.cvtColor(img1,cv.COLOR_BGR2RGB))
ax[0].set_title("Image 1")
ax[0].set_xticks([]) , ax[0].set_yticks([])
ax[1].imshow(cv.cvtColor(img4,cv.COLOR_BGR2RGB))
ax[1].set_title("Image 2")
ax[1].set_xticks([]) , ax[1].set_yticks([])
ax[2].imshow(cv.cvtColor(im1to4,cv.COLOR_BGR2RGB))
ax[2].set_title("Image 1 Warped")
ax[2].set_xticks([]) , ax[2].set_yticks([])
plt.show()
```

Image 1

Image 2

Image 1 Warped