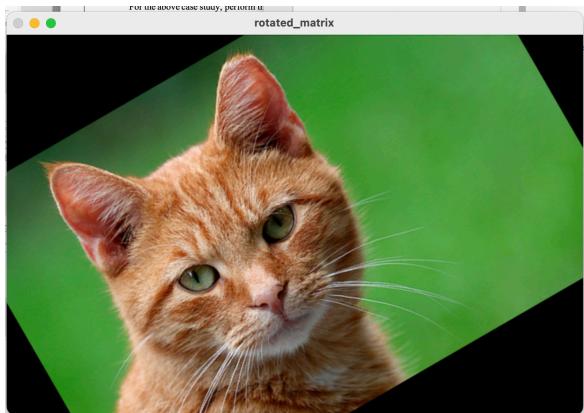


Activity 5



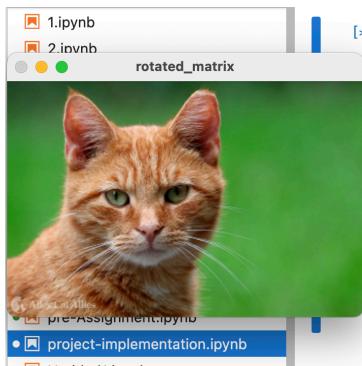
The screenshot shows a Jupyter Notebook interface. On the left, there is a thumbnail of a ginger cat image labeled 'rotated_matrix'. The main notebook cell contains Python code for performing various image transformations:

```
#Rotation
angle = 30
rows,cols = image.shape[:2]
rotation_matrix = cv2.getRotationMatrix2D((cols/2 , rows/2) , angle , 1)
rotated_matrix = cv2.warpAffine(image , rotation_matrix , (cols,rows))

#Scaling
scale_factor = 0.5
scaled_image = cv2.resize(image , None , fx=scale_factor , fy = scale_factor)

#Perspective transformation
pts1 = np.float32([[50,50] , [200,50] , [50,200] , [200,200]])
pts2 = np.float32([[0,0] , [250,0] , [0,250] , [250,250]])

show_img({'rotated_matrix':rotated_matrix})
```



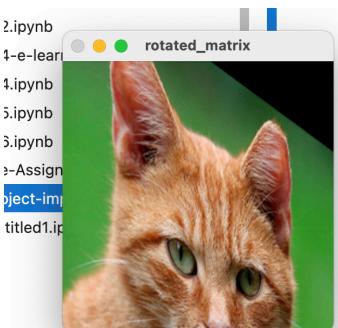
The screenshot shows a Jupyter Notebook interface. On the left, there is a thumbnail of a ginger cat image labeled 'rotated_matrix'. The main notebook cell contains Python code for scaling the image:

```
#Rotation
angle = 30
rows,cols = image.shape[:2]
rotation_matrix = cv2.getRotationMatrix2D((cols/2 , rows/2) , angle , 1)
rotated_matrix = cv2.warpAffine(image , rotation_matrix , (cols,rows))

#Scaling
scale_factor = 0.5
scaled_image = cv2.resize(image , None , fx=scale_factor , fy = scale_factor)

#Perspective transformation
pts1 = np.float32([[50,50] , [200,50] , [50,200] , [200,200]])
pts2 = np.float32([[0,0] , [250,0] , [0,250] , [250,250]])

show_img({'rotated_matrix':scaled_image})
```



The screenshot shows a Jupyter Notebook interface. On the left, there is a thumbnail of a ginger cat image labeled 'rotated_matrix'. The main notebook cell contains Python code for performing perspective transformation:

```
scale_factor = 0.5
scaled_image = cv2.resize(image , None , fx=scale_factor , fy = scale_factor)

#Perspective transformation
pts1 = np.float32([[50,100] , [200,30] , [50,200] , [200,200]])
pts2 = np.float32([[0,0] , [100,0] , [0,100] , [100,150]])
perspective_matrix = cv2.getPerspectiveTransform(pts1,pts2)
perspective_transformed_image = cv2.warpPerspective(image, perspective_matrix , (250,250))

show_img({'rotated_matrix':perspective_transformed_image})
```