Geometric Image Transformation

By Prakhar Gupta(16UCS131)

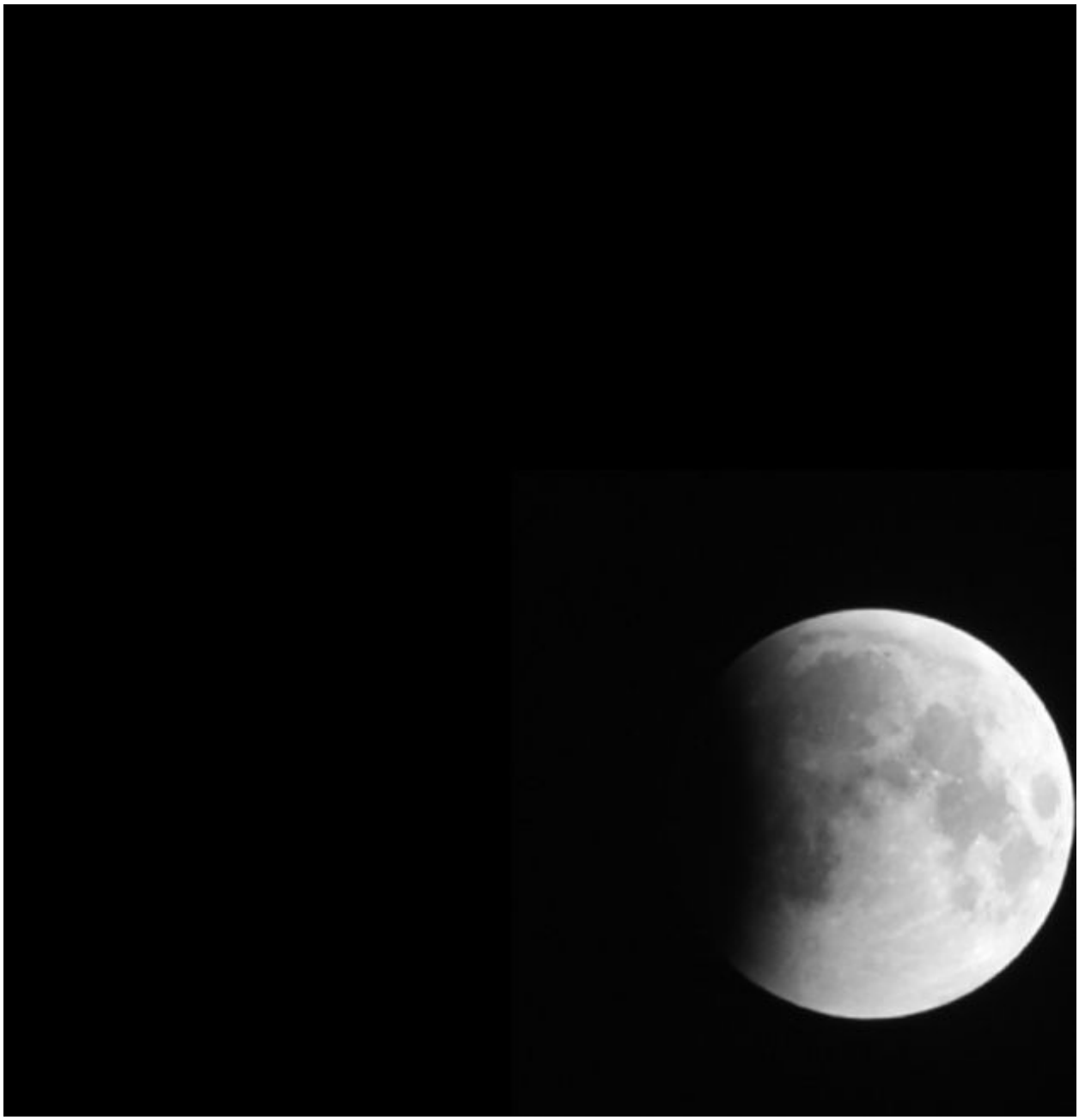
Q1. Translating An Image

Code:

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

img = cv2.imread('moon.jpg',cv2.IMREAD_GRAYSCALE)
cv2.imshow('image',img)

BLACK =[0,0,0]

img2 = cv2.copyMakeBorder(img,100,100,100,100,cv2.BORDER_CONSTANT,value=BLACK)
cv2.imshow('padimage',img2)

height,width = img2.shape[:2]

tx = 200
ty = 175
T = np.float32([[1,0,tx],
    [0,1,ty]])

f_img = cv2.warpAffine(img2,T,(width,height))

cv2.imshow('Final Image',f_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Input Image:

Output Image:



Observation:

Image was initially going out of bounds by translation so we had to add an offset.

Q2. Rotating An Image

Code:

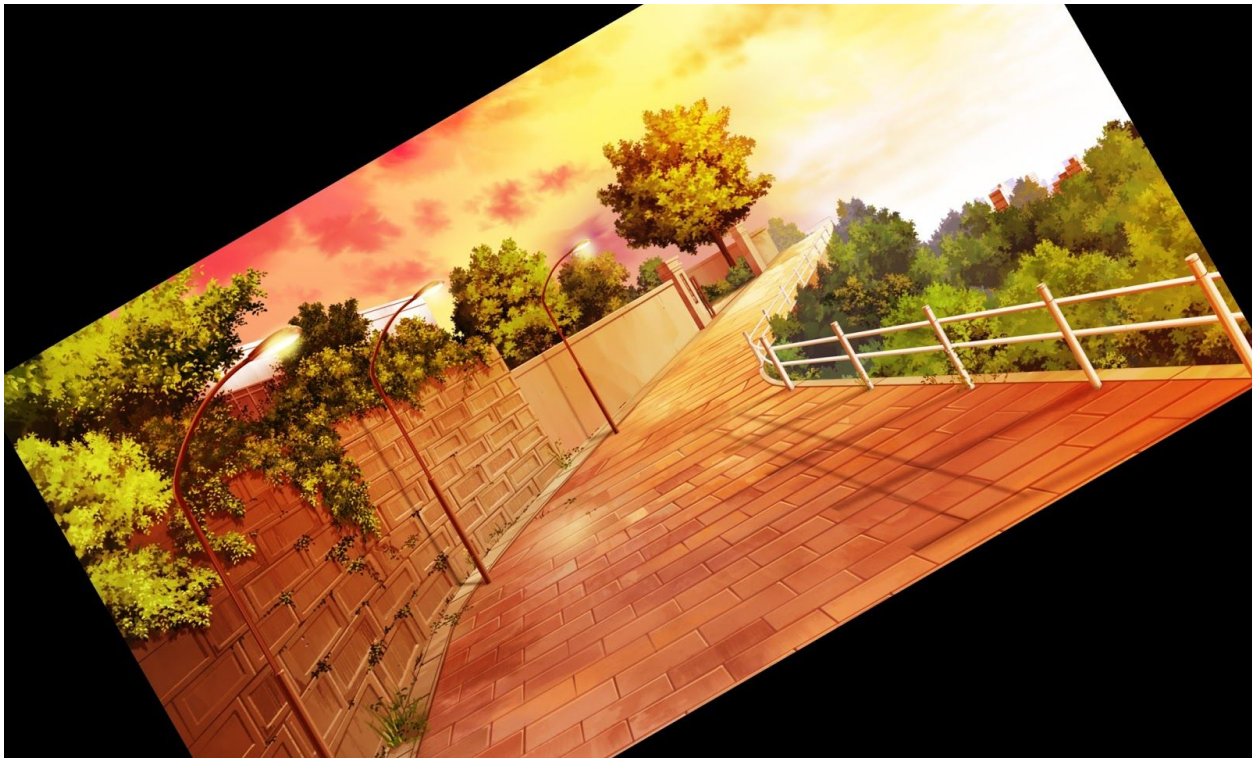
```
import cv2
import numpy as np
import matplotlib.pyplot as plt

img = cv2.imread('pic.png',1)
cv2.imshow('image',img)

BLACK =[0,0,0]

img2 = cv2.copyMakeBorder(img,100,100,100,100,cv2.BORDER_CONSTANT,value=BLACK)
cv2.imshow('padimage',img2)

rows,columns,channels = img2.shape

R2 = cv2.getRotationMatrix2D((columns/2,rows/2),120,1)
output2 = cv2.warpAffine(img2,R2,(columns,rows))
cv2.imwrite('rot120.jpg',output2)
cv2.imshow('output2',output2);

for i in range(1,9):
    R = cv2.getRotationMatrix2D((columns/2,rows/2),15*i,1)
    output = cv2.warpAffine(img2,R,(columns,rows))
    cv2.imshow('output1'+str(i),output)
    cv2.imwrite('rot15' + str(i)+'.jpg',output)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

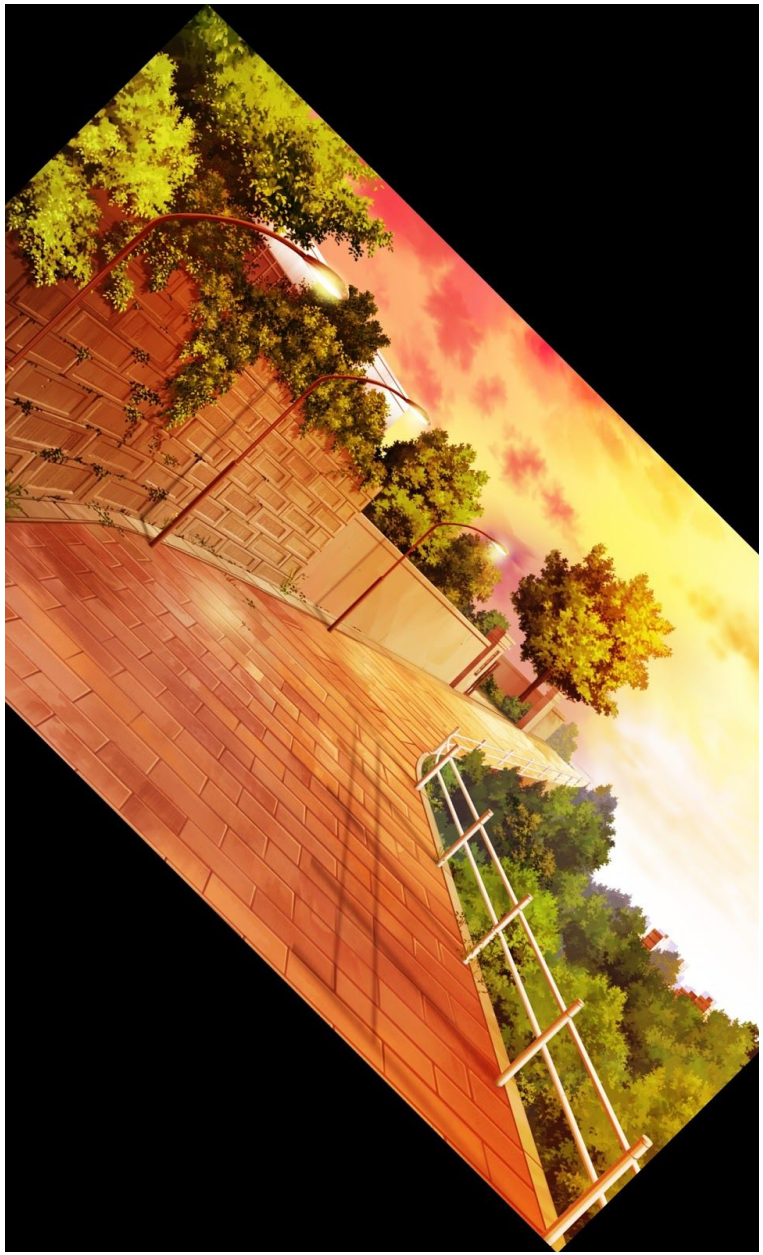
Input Image

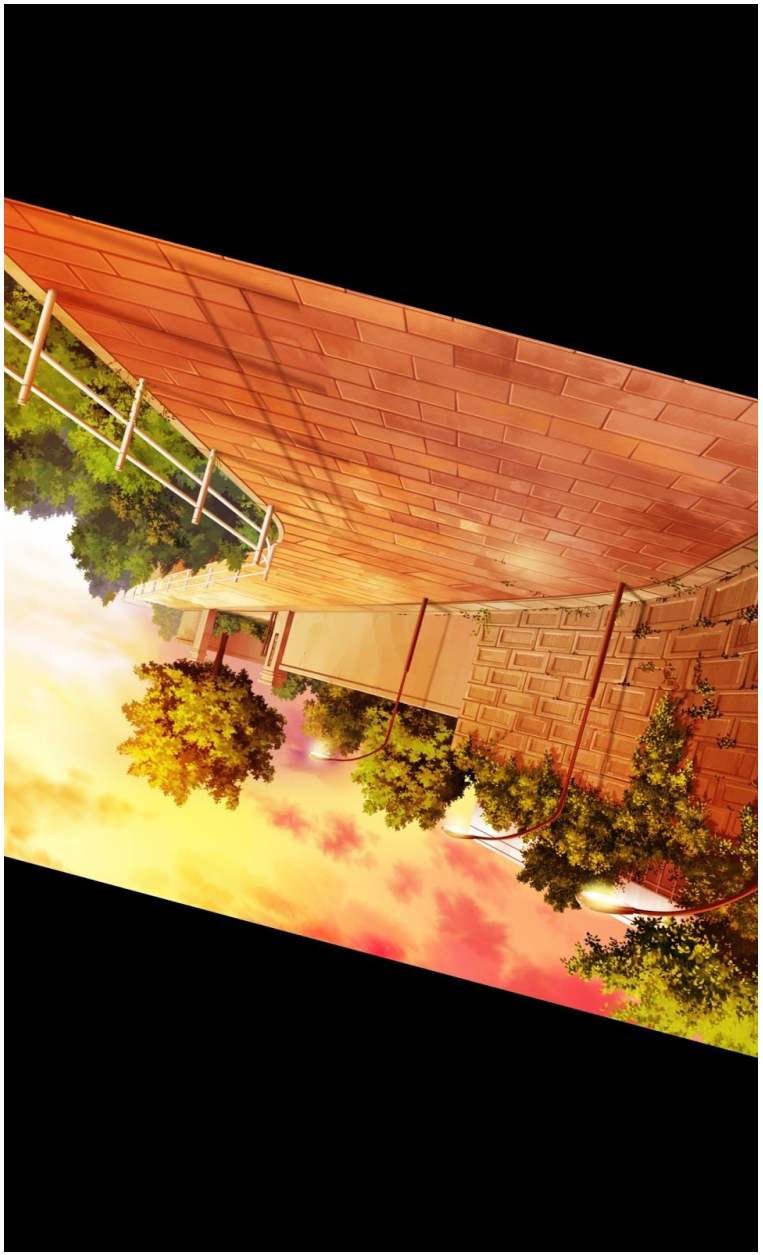
Output Image

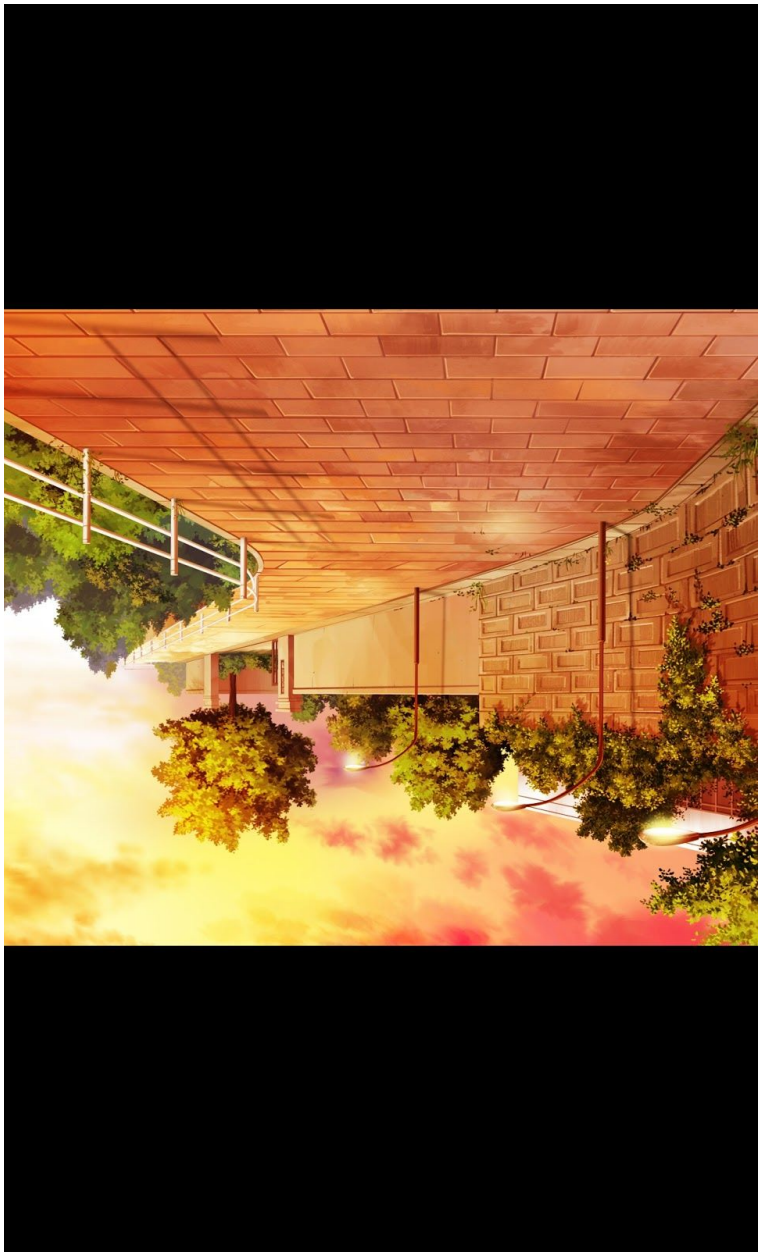
Direct

InDirect









Observation:

We can use “for” loop for visualizing the 15 degree change in images.