Code:

# %%

from PIL import Image

import cv2

from matplotlib import pyplot as plt

import numpy as np

filepath = 'image5.jpg'

img = cv2.imread(filepath)

h,w,c = img.shape

print(type(img))

print(img.shape)

plt.imshow(img)

plt.xlabel('Original image in BGR')

plt.savefig('BGR\_image.jpg')

plt.show()

# %%

#COLOR TO GRAYSCALE

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

img = cv2.cvtColor(img, cv2.COLOR\_RGB2GRAY)

#img = cv2.resize(img,(64,64), cv2.INTER\_LINEAR)

print(type(img))

print(img.shape)

plt.imshow(img, cmap= 'gray')

plt.xlabel('Original image in grayscale')

plt.savefig('Grayscale.jpg')

plt.show()

# %%

#ROTATE 90 DEGREE CCW

img = cv2.rotate(img, cv2.ROTATE\_90\_COUNTERCLOCKWISE)

print(type(img))

print(img.shape)

plt.imshow(img, cmap= 'gray')

plt.xlabel('90 degree CCW rotated image')

plt.savefig('90\_CCW.jpg')

plt.show()

# %%

#ARBITRARY ANGLE ROTATE

mat = cv2.getRotationMatrix2D((h/2, w/2), 45, 1)

img = cv2.warpAffine(img,mat,(h,w), borderValue=100 )

print(type(img))

print(img.shape)

plt.imshow(img, cmap= 'gray')

plt.xlabel('45 degree rotated image')

plt.savefig('arbitrary rotated.jpg')

plt.show()

# %%

#IMAGE TRANSLATION

tx = w/4; ty= h/4

mat = np.array([ [1,0,tx], [0,1,ty] ], dtype=np.float32)

img = cv2.warpAffine(img, mat, (h,w))

print(type(img))

print(img.shape)

plt.imshow(img, cmap= 'gray')

plt.xlabel('Translated Image')

plt.savefig('Translated.jpg')

plt.show()

# %%

#IMAGE BINARIZATION USING BUILT IN OTSU THRESHOLDING

img = cv2.imread(filepath)

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

img = cv2.cvtColor(img, cv2.COLOR\_RGB2GRAY)

val, img = cv2.threshold(img, 154,255, cv2.THRESH\_BINARY+cv2.THRESH\_OTSU)

print(val)

plt.imshow(img, cmap= 'gray')

plt.xlabel('Binary image using built-in OTSU method')

plt.savefig('Binary image.jpg')

plt.show()

Output:

