**Machine Vision System Practical 3**

**Programming for Normal & Perspective Transformation**

**in OpenCV using Python**

In this Practical, we will learn

1. How to Transform an Image in 2D plane i.e. Translation and Rotation.

&

1. How to Transform an Image in 3D plane i.e. Perspective transformation.

**1] Normal Transformation of an Image:**

* **Importing OpenCV & Numpy libraries:**

The code starts by importing the OpenCV library i.e. cv2 as cv, which is used for computer vision tasks, including reading and processing images. We also import numpy library as np for matrix calculations for transformation.

* **Reading the Original Image:**

Here we read our image with the help of cv.imread() function by giving it the proper path where the image is stored. And then we display the image with any name suitable using cv.imshow() function.

* **Writing a Function for Translation:**

To translate the image, we first write a function named translate that takes three parameters:

1. Image
2. Translation along x
3. Translation along y

This function creates a translation matrix using these above parameters and also the dimensions of the image. This Translation matrix and dimensions are then used to *Warp* the image with the help of function cv.warpAffine().

* **Writing a Function for Rotation:**

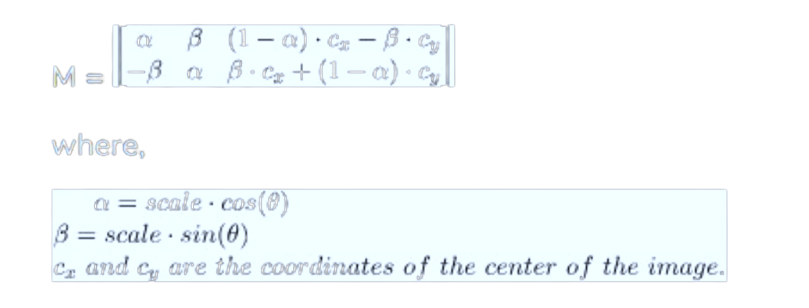
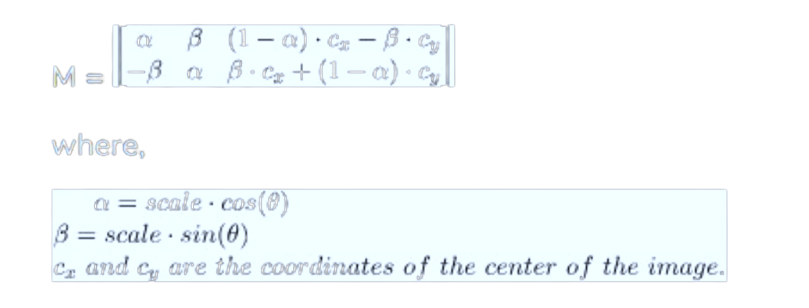
To rotate the image, we also write a function named rotate that takes three parameters:

1. Image
2. Angle of Rotation
3. Point at which the images is to be rotated (*It is defined as the Midpoint*)

This function creates a rotational matrix with the help of cv.getRotationMatrix2D() and giving it the above parameters and also the dimensions of the image. This Rotation matrix and dimensions are then used to *Warp* the image with the help of function cv.warpAffine().

This is a type of affine transformation. An affine transformation preserves lines and parallelism. These transformation matrix are taken by wrapaffine() function as parameter and the rotated image will be returned.

Here,



* **Main Program:**

Now we call our **Translate** function by giving it our original image that is to be translated with x & y values accordingly:

* -x --> Left
* -y --> Up
* x --> Right
* y --> Down

And also the **Rotate** function by giving it our original image that is to be rotated with the angle of rotation

And finally we display the translated & rotated image by using cv.imshow() function.

**Code:**

import cv2 as cv

import numpy as np

img = cv.imread('Photos/kkwagh.jpg')

cv.imshow('College', img)

# Translation

def translate(img, x, y):

    transMat = np.float32([[1,0,x],[0,1,y]])

    dimensions = (img.shape[1], img.shape[0])

    return cv.warpAffine(img, transMat, dimensions)

translated = translate(img, -25, 25)

cv.imshow('Translated', translated)

# Rotation

def rotate(img, angle, rotPoint=None):

    (height,width) = img.shape[:2]

    if rotPoint is None:

        rotPoint = (width//2,height//2)

    rotMat = cv.getRotationMatrix2D(rotPoint, angle, 1.0)

    dimensions = (width,height)

    return cv.warpAffine(img, rotMat, dimensions)

rotated = rotate(img, -45)

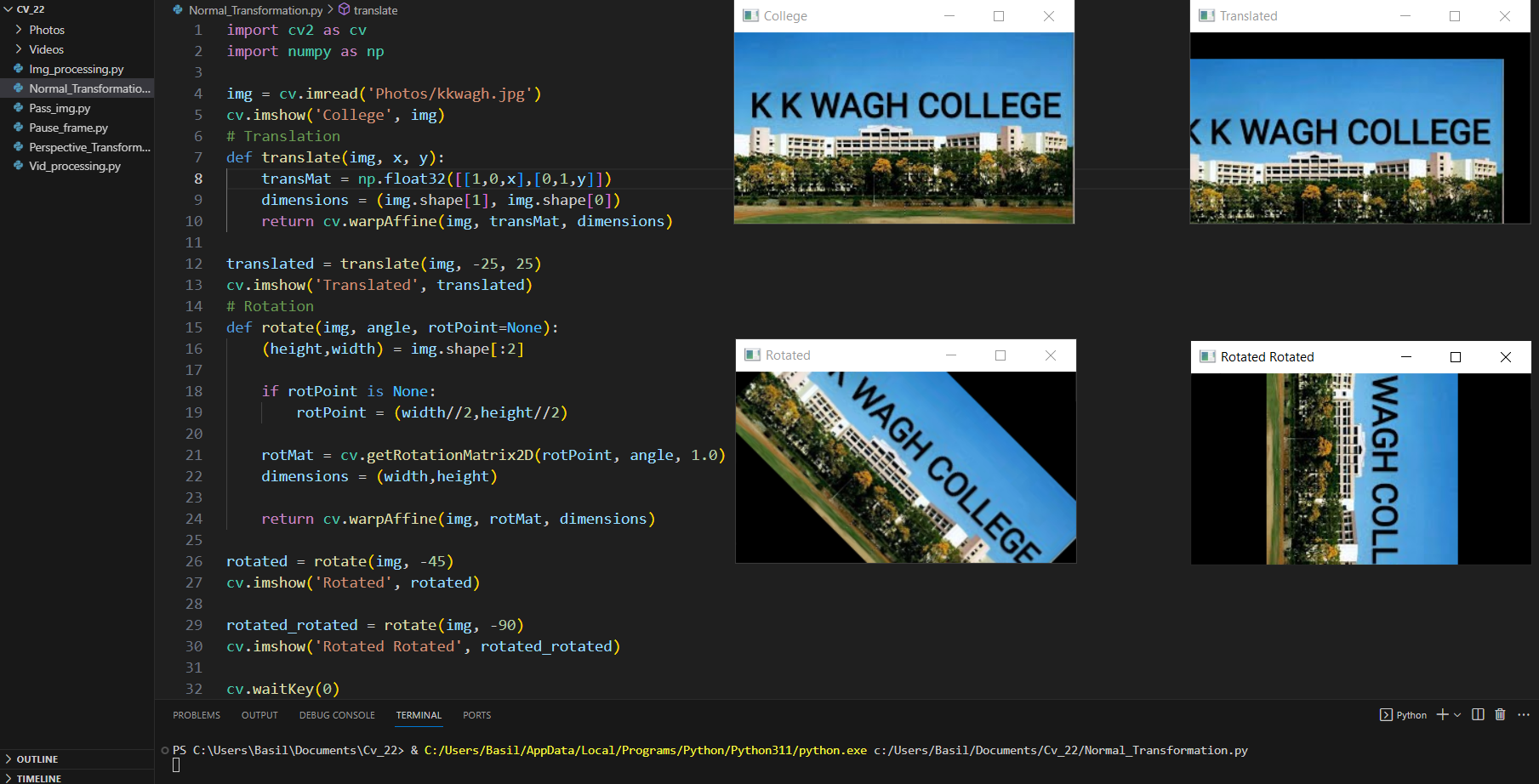
cv.imshow('Rotated', rotated)

rotated\_rotated = rotate(img, -90)

cv.imshow('Rotated Rotated', rotated\_rotated)

cv.waitKey(0)

**Output:**

****

**2] Perspective Transformation of an Image:**

* **Importing OpenCV & Numpy libraries:**

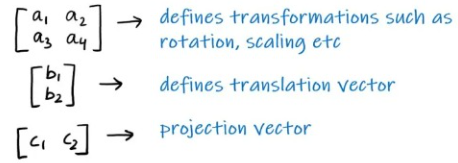
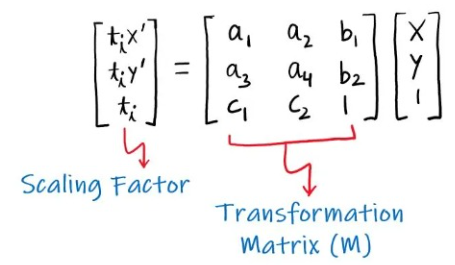
The code starts by importing the OpenCV library i.e. cv2 as cv, which is used for computer vision tasks, including reading and processing images. We also import numpy library as np for matrix calculations for transformation.

* **Reading the Original Image:**

Here we read our image with the help of cv.imread() function by giving it the proper path where the image is stored. And then we display the image with any name suitable using cv.imshow() function.

* **Perspective Transformation:**

In Perspective Transformation, the straight lines remain straight even after the transformation. To apply a perspective transformation, we need a 3×3 perspective transformation matrix. We need four points on the input image and corresponding four points on the output image.



* **Main Program:**

First we define **pts1** and **pts2**. **pts1** is an array of four points on the input image and **pts2** is an array of corresponding four points on the output image. Then we have to Compute the perspective transform matrix M using cv.getPerspectiveTransform(pts1, pts2) function. This matrix M and dimensions are then used to *Warp* the image with the help of function cv.warpPerspective().

And finally we display the Transforms image by using cv.imshow() function.

**Code:**

import cv2 as cv

import numpy as np

img = cv.imread('Photos/kkwagh.jpg')

cv.imshow('College', img)

rows,cols,ch = img.shape

# define four points on input image

pts1 = np.float32([[56,65],[368,52],[28,387],[389,390]])

# define the corresponding four points on output image

pts2 = np.float32([[100,50],[300,0],[0,300],[300,300]])

M = cv.getPerspectiveTransform(pts1,pts2)

dst = cv.warpPerspective(img,M,(cols, rows))

cv.imshow('Transformed Image', dst)

cv.waitKey(0)

**Output:**

