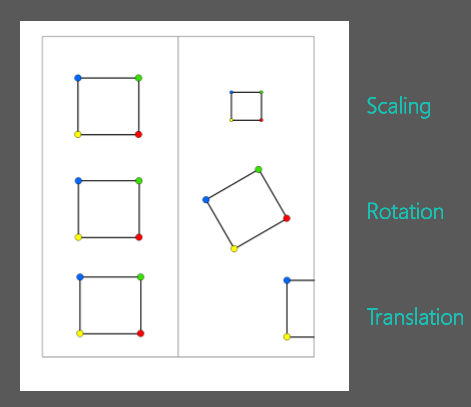
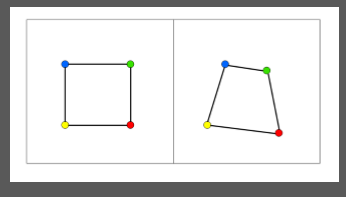
**Transformations:**

Bozulmaları ve perspektif sorunları düzeltmek için kullanılır.

* Affine (Afin / Latince Affines:birlikte bağlılık)
* Non – Affine / Projective Transform / Homography

**Affine:**

**Non – Affine:**

(Colliniarity: Eş Doğrusallık)

**Translations:**

**Translation Matrix:**

Tx - Represents the shift along the x--axis (horizontal)

Ty - Represents the shift along the y--axis (vertical)

The OpenCV function cv2.warpAffine to implement these translation

**Rotations:**

**Rotation Matrix:**

- the angle of rotation

cv2.getRotationMatrix2D(rotation\_center\_x, rotation\_center\_y, angle of rotation, scale)

import cv2

import numpy as np

image = cv2.imread('c:\\users\\egek\\images\\ataturk.jpg')

# Store height and width of the image

height, width = image.shape[:2]

quarter\_height, quarter\_width = height/4, width/4

# | 1 0 Tx |

# T = | 0 1 Ty |

# T is our translation matrix

T = np.float32([[1, 0, quarter\_width], [0, 1,quarter\_height]])

# We use warpAffine to transform the image using the matrix, T

img\_translation = cv2.warpAffine(image, T, (width, height))

cv2.imshow('Translation', img\_translation)

cv2.waitKey()

cv2.destroyAllWindows()

print T

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Rotations:

import cv2

import numpy as np

image = cv2.imread('c:\\users\\egek\\images\\ataturk.jpg')

height, width = image.shape[:2]

# Divide by two to rototate the image around its centre

rotation\_matrix = cv2.getRotationMatrix2D((width/2, height/2), 90, .5)

rotated\_image = cv2.warpAffine(image, rotation\_matrix, (width, height))

cv2.imshow('Rotated Image', rotated\_image)

cv2.waitKey()

cv2.destroyAllWindows()

#Other Option to Rotate

img = cv2.imread('c:\\users\\egek\\images\\ataturk.jpg')

rotated\_image = cv2.transpose(img)

cv2.imshow('Rotated Image - Method 2', rotated\_image)

cv2.waitKey()

cv2.destroyAllWindows()

----------------------------------------------------

# Horizontal flip.

flipped = cv2.flip(image, 1)

cv2.imshow('Horizontal Flip', flipped)

cv2.waitKey()

cv2.destroyAllWindows()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Scaling:

import cv2

import numpy as np

image = cv2.imread('c:\\users\\egek\\images\\ataturk.jpg')

# Make our image 3/4 of it's original size

image\_scaled = cv2.resize(image, None, fx=0.75, fy=0.75)

cv2.imshow('Scaling - Linear Interpolation', image\_scaled)

cv2.waitKey()

----------------------------------------------------

# Double the size of our image

img\_scaled = cv2.resize(image, None, fx=2, fy=2, interpolation = cv2.INTER\_CUBIC)

cv2.imshow('Scaling - Cubic Interpolation', img\_scaled)

cv2.waitKey()

---------------------------------------------------

#Skew the re-sizing by setting exact dimensions

img\_scaled = cv2.resize(image, (900, 400), interpolation = cv2.INTER\_AREA)

cv2.imshow('Scaling - Skewed Size', img\_scaled)

cv2.waitKey()

cv2.destroyAllWindows()