



**Department Of Creative Technologies.**

**Subject :**  
Linear Algebra.

**Submitted to:**  
Miss Nomana Abid.

**Submitted BY:**

221785	Muhammad Abdullah
221807	Muhammad Sohaib
221787	Syed Muhammad Taha
221849	Ahsan Ali khan
221841	Nayyar Abbas

**Project:** Take any two pictures of your own choice.transform the selected Figures using the following operators through MATLAB or Python.

1. Dilation of first figure by a factor of 3.

## Code:

```
import cv2
import numpy as n
import matplotlib.pyplot as p

image = cv2.imread('image1.jpg', cv2.IMREAD_COLOR)

# Define the dilation factor
kernel = n.ones((3, 3), n.uint8)

# Perform dilation
dilated_image = cv2.dilate(image, kernel, iterations=1)

x_range = n.arange(0, image.shape[1])
y_range = n.arange(0, image.shape[0])

# Displaying the original and dilated images using matplotlib in Cartesian plane
p.figure(figsize=(10, 6))

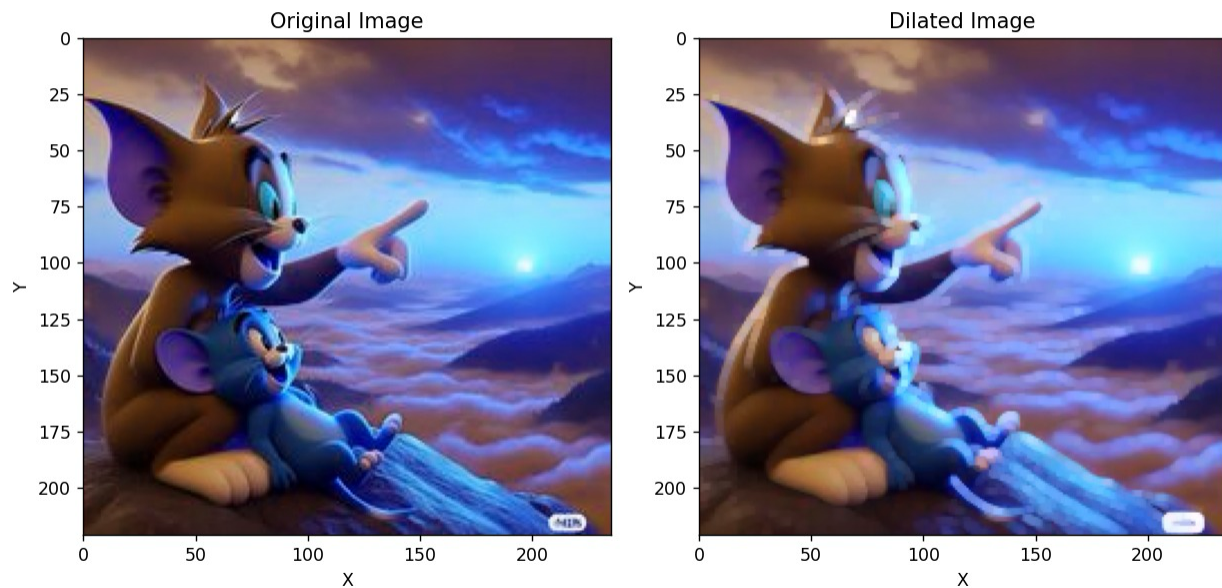
# Original Image
p.subplot(1, 2, 1)
p.imshow(image, cmap=None, extent=[0, image.shape[1], image.shape[0], 0])
p.title('Original Image')
p.xlabel('X')
p.ylabel('Y')

# Dilated Image
p.subplot(1, 2, 2)
p.imshow(dilated_image, cmap=None, extent=[0, dilated_image.shape[1],
dilated_image.shape[0], 0])
p.title('Dilated Image')
p.xlabel('X')
p.ylabel('Y')
```

```
p.tight_layout()
p.show()
```

## Output:

Figure 1



2. Clockwise Rotation of second figure about the origin with angle  $75^\circ$ .

## Code:

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

# Craeting Cartoon
# Body
body = [(0, 0), (2, -8), (2.5, -10), (3, -12), (2.5, -14), (0, -16),
        (-2.5, -14), (-3, -12), (-2.5, -10), (-2, -8), (0, 0)]
```

```

# Head
head = [(0, 0), (2, 2), (4, 4), (5, 7), (4, 10), (2, 12), (0, 13),
        (-2, 12), (-4, 10), (-5, 7), (-4, 4), (-2, 2), (0, 0)]

# Left Eye
left_eye = [(-1.5, 8), (-1, 8.5), (-0.5, 8), (-1, 7.5), (-1.5, 8)]

# Right Eye
right_eye = [(1.5, 8), (2, 8.5), (2.5, 8), (2, 7.5), (1.5, 8)]

# Mouth
mouth = [(-1, 5), (-0.5, 4.5), (0.5, 4.5), (1, 5), (0, 4), (-1, 5)]

# Function to plot and fill a shape
def plot_fill_shape(shape, edge_color='black', fill_color=None):
    x, y = zip(*shape)
    plt.plot(x, y, marker='o', color=edge_color)
    if fill_color:
        plt.fill(x, y, fill_color)

# Function to rotate a shape
def rotate_shape(shape, angle_degrees):
    angle_radians = np.radians(angle_degrees)
    cos_theta = np.cos(angle_radians)
    sin_theta = np.sin(angle_radians)
    return [(x * cos_theta - y * sin_theta, x * sin_theta + y * cos_theta) for x, y in
            shape]

# Rotate the cartoon by 75 degree clockwise
body_rotated = rotate_shape(body, -75)                #take angle negative for
clockwise rotation
head_rotated = rotate_shape(head, -75)
left_eye_rotated = rotate_shape(left_eye, -75)
right_eye_rotated = rotate_shape(right_eye, -75)
mouth_rotated = rotate_shape(mouth, -75)

# size of original and rotated cartoon figures
plt.figure(figsize=(10, 5))

# Plot original cartoon figure
plt.subplot(1, 2, 1)
plot_fill_shape(body, edge_color='red', fill_color='pink')
plot_fill_shape(head, edge_color='red', fill_color='pink')
plot_fill_shape(left_eye, edge_color='red', fill_color='pink')
plot_fill_shape(right_eye, edge_color='red', fill_color='pink')
plot_fill_shape(mouth, edge_color='red', fill_color='pink')
plt.gca().set_aspect('equal', adjustable='box')
plt.grid(True)

```

```

plt.title('Original Cartoon')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.xlim(-20, 20)
plt.ylim(-20, 20)

# Plot rotated cartoon figure
plt.subplot(1, 2, 2)
plot_fill_shape(body_rotated, edge_color='blue', fill_color='lightblue')
plot_fill_shape(head_rotated, edge_color='blue', fill_color='lightblue')
plot_fill_shape(left_eye_rotated, edge_color='blue', fill_color='lightblue')
plot_fill_shape(right_eye_rotated, edge_color='blue', fill_color='lightblue')
plot_fill_shape(mouth_rotated, edge_color='blue', fill_color='lightblue')
plt.gca().set_aspect('equal', adjustable='box')
plt.grid(True)
plt.title('After Rotating 75 degree')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.xlim(-20, 20)
plt.ylim(-20, 20)

plt.tight_layout()
plt.show()

```

## Output:

