Experiment Number : 3

Problem Statement: **Write Python code to perform Data Augmentation using affine transform**

NAME: Harshwardhan PatilROLL NO: 50

CLASS: IT-B BATCH: B1

CODE:

import cv2

import numpy as np

import matplotlib.pyplot as plt

def apply\_affine\_transform(image, scale=1.0, angle=0, tx=0, ty=0, shear=0):

    """

    Apply affine transformation to an image with scaling, rotation, translation, and shearing.

    Parameters:

        image: Input image (numpy array).

        scale: Scaling factor.

        angle: Rotation angle (in degrees).

        tx: Translation along the x-axis.

        ty: Translation along the y-axis.

        shear: Shearing factor.

    Returns:

        Transformed image.

    """

    (h, w) = image.shape[:2]

    center = (w // 2, h // 2)

    # Rotation and Scaling

    M = cv2.getRotationMatrix2D(center, angle, scale)

    M[0, 2] += tx

    M[1, 2] += ty

    # Shear transformation matrix

    shear\_matrix = np.array([

        [1, shear, 0],

        [0, 1, 0]

    ], dtype=np.float32)

    # Combine the transformations

    M = np.vstack([M, [0, 0, 1]])  # Convert to 3x3 matrix

    M = np.dot(shear\_matrix, M)    # Apply shearing

    # Warp the image

    transformed = cv2.warpAffine(image, M[:2], (w, h), flags=cv2.INTER\_LINEAR, borderMode=cv2.BORDER\_REFLECT\_101)

    return transformed

def plot\_images(images, titles, rows=2, cols=3):

    """

    Plot multiple images in a grid layout.

    """

    plt.figure(figsize=(15, 10))

    for i, (img, title) in enumerate(zip(images, titles)):

        plt.subplot(rows, cols, i + 1)

        plt.imshow(cv2.cvtColor(img, cv2.COLOR\_BGR2RGB))

        plt.title(title)

        plt.axis('off')

    plt.tight\_layout()

    plt.show()

# Main function to demonstrate various data augmentation techniques

if \_\_name\_\_ == "\_\_main\_\_":

    # Load the image

    image\_path = "image.jpg"

    image = cv2.imread(image\_path)

    if image is None:

        raise FileNotFoundError(f"Image not found at {image\_path}")

    # Augmentations

    augmented\_images = [

        image,  # Original

        apply\_affine\_transform(image, scale=1.5),             # Scaling

        apply\_affine\_transform(image, angle=45),             # Rotation

        apply\_affine\_transform(image, tx=50, ty=-30),        # Translation

        apply\_affine\_transform(image, shear=0.2),            # Shearing

        cv2.flip(image, 1)                                   # Horizontal Flip

    ]

    titles = [

        "Original",

        "Scaled (1.5x)",

        "Rotated (45°)",

        "Translated (tx=50, ty=-30)",

        "Sheared (shear=0.2)",

        "Horizontally Flipped"

    ]

    # Plot the augmented images

    plot\_images(augmented\_images, titles)

OUTPUT:

