**Ex No:3**

**Date:**

**SYNTHETIC IMAGE GENERATION USING TRADITIONAL DATA AUGMENTATION**

**AIM**

To perform traditional image augmentation techniques (flip, rotation, brightness adjustment, and zoom) on a given image using OpenCV and visualize the results with Matplotlib.

**STEPS**

1. Import Required Libraries

cv2 for image processing.

numpy for numerical operations.

matplotlib.pyplot for displaying images.

2. Load the Image

Read the input image using cv2.imread().

Convert from BGR to RGB color format using cv2.cvtColor() for correct display.

3. Resize the Image

Use cv2.resize() to standardize the size to 224×224 pixels.

4. Display the Original Image

Use plt.imshow() to show the original image.

5. Define Augmentation Functions

Flip the image horizontally using cv2.flip().

Rotate the image by 30° using cv2.getRotationMatrix2D() and cv2.warpAffine().

Adjust Brightness using cv2.convertScaleAbs() with increased alpha and beta

values.

Zoom by cropping the image and resizing back to 224×224.

6. Apply Augmentations

Call the augmentation function and store the results.

7. Display Augmented Images

Use subplots to show all augmented images with titles.

**Program:**  
  
import cv2 # OpenCV for image processing

import numpy as np

import matplotlib.pyplot as plt

import os

# Load image safely

img\_path = '/content/Industrail visit.jpg' # Replace with your image file path

if not os.path.exists(img\_path):

raise FileNotFoundError(f"Image not found at path: {img\_path}")

original = cv2.imread(img\_path)

if original is None:

raise ValueError(f"Unable to read image. Check if {img\_path} is a valid image file.")

# Convert BGR to RGB

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

# Resize to 224x224

image = cv2.resize(original, (224, 224))

# Show Original Image

plt.figure(figsize=(5, 5))

plt.imshow(image)

plt.title("Original Image")

plt.axis("off")

plt.show()

# Define Traditional Augmentations

def traditional\_augmentations(image):

flipped = cv2.flip(image, 1) # Horizontal flip

# Rotate around center

center = (image.shape[1] // 2, image.shape[0] // 2)

M = cv2.getRotationMatrix2D(center, angle=30, scale=1.0)

rotated = cv2.warpAffine(image, M, (image.shape[1], image.shape[0]))

# Increase brightness

bright = cv2.convertScaleAbs(image, alpha=1.2, beta=30)

# Crop and zoom

cropped = image[30:194, 30:194] # (164x164)

zoomed = cv2.resize(cropped, (224, 224))

return [flipped, rotated, bright, zoomed]

# Apply Augmentations

augmented\_images = traditional\_augmentations(image)

titles = ["Flipped", "Rotated", "Brighter", "Zoomed"]

# Display Augmented Images

plt.figure(figsize=(15, 4))

for i, aug\_img in enumerate(augmented\_images):

plt.subplot(1, 4, i + 1)

plt.imshow(aug\_img)

plt.title(titles[i])

plt.axis("off")

plt.tight\_layout()

plt.show()

**Output:**



|  |  |
| --- | --- |
| COE (20) |  |
| Record (20) |  |
| VIVA (10) |  |
| Total (50) |  |

**Result:**

Successfully performed flip, rotation, brightness enhancement, and zoom on the original image to

create multiple augmented versions.