**Translation Noise**

from PIL import Image

import numpy as np

def add\_translation\_noise(image, max\_translation):

width, height = image.size

# Generate random translation values

translation\_x = np.random.randint(-max\_translation, max\_translation + 1)

translation\_y = np.random.randint(-max\_translation, max\_translation + 1)

# Create the translation matrix

translation\_matrix = (1, 0, translation\_x, 0, 1, translation\_y)

# Apply the translation to the image

translated\_image = image.transform((width, height), Image.AFFINE, translation\_matrix)

return translated\_image

# Example usage

image = Image.open('/kaggle/input/image-test/test.jpg') # Load your image here

# image = image.resize((28, 28)) # Resize the image to 28x28 if needed

max\_translation = 5 # Maximum translation in pixels

noisy\_image = add\_translation\_noise(image, max\_translation)

# Save the original and noisy images

image.save('original\_image.jpg')

noisy\_image.save('noisy\_image.jpg')



Figure 2 Translation Noise

Figure 1 Original Image

**Translation Noise with Pixel-Level Noise**

from PIL import Image

import numpy as np

def add\_translation\_noise(image, max\_translation, noise\_intensity):

width, height = image.size

# Generate random translation values

translation\_x = np.random.randint(-max\_translation, max\_translation + 1)

translation\_y = np.random.randint(-max\_translation, max\_translation + 1)

# Create the translation matrix

translation\_matrix = (1, 0, translation\_x, 0, 1, translation\_y)

# Apply the translation to the image

translated\_image = image.transform((width, height), Image.AFFINE, translation\_matrix)

# Add pixel-level noise

noisy\_image = np.array(translated\_image)

noise = np.random.normal(scale=noise\_intensity, size=noisy\_image.shape).astype(np.int)

noisy\_image = np.clip(noisy\_image + noise, 0, 255).astype(np.uint8)

return Image.fromarray(noisy\_image)

# Example usage

image = Image.open('/kaggle/input/image-test/test.jpg') # Load your image here

max\_translation = 5 # Maximum translation in pixels

noise\_intensity = 20 # Adjust the noise intensity to control the amount of visible noise

noisy\_image = add\_translation\_noise(image, max\_translation, noise\_intensity)

# Save the original and noisy images

image.save('original\_image3.jpg')

noisy\_image.save('noisy\_image3.jpg')



Figure : Transplation+Pixel Level Noise

Figure Orignal Image