**Use the concept of Data Augmentation to increase the data size from a single image.**

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras.preprocessing.image import ImageDataGenerator, img\_to\_array, load\_img

import numpy as np

import matplotlib.pyplot as plt

import os

from PIL import Image

# Create a directory to save augmented images

os.makedirs("augmented\_images", exist\_ok=True)

# Upload an image manually in Google Colab

from google.colab import files

uploaded = files.upload()

image\_path = list(uploaded.keys())[0]

# Load the uploaded image

image = load\_img(image\_path) # Load image as PIL object

image\_array = img\_to\_array(image) # Convert to array

image\_array = np.expand\_dims(image\_array, axis=0) # Expand dimensions to match input format

# Define Data Augmentation parameters

datagen = ImageDataGenerator(

rotation\_range=40,

width\_shift\_range=0.2,

height\_shift\_range=0.2,

shear\_range=0.2,

zoom\_range=0.2,

horizontal\_flip=True,

brightness\_range=[0.5, 1.5],

fill\_mode='nearest'

)

# Generate and visualize augmented images

num\_images = 10

fig, axes = plt.subplots(1, num\_images, figsize=(20, 5))

# Generate images

i = 0

for batch in datagen.flow(image\_array, batch\_size=1, save\_to\_dir="augmented\_images", save\_prefix="aug", save\_format="jpg"):

ax = axes[i]

ax.imshow(batch[0].astype('uint8'))

ax.axis('off')

i += 1

if i == num\_images:

break # Stop after generating required images

plt.show()