Data Collection and Preprocessing Phase

Date	20 August 2024
Team ID	Sneha S
Project Title	mushroom
Maximum Marks	6 Marks

Preprocessing Template

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Section	**Description**
Data Overview	The dataset contains images of wild mushrooms categorized as **edible** or **poisonous**. Each image is a color photo (RGB) collected from open-source repositories. The goal is to train a convolutional neural network (CNN) that can classify new mushroom images as edible or poisonous.
Resizing	All images were resized to a uniform **128 × 128 pixels** to ensure consistent input dimensions for the CNN and to reduce computational cost during training.
Normalization	Pixel values were scaled from the original 0–255 range to **0–1** using rescale=1./255. Normalization accelerates training and helps the model converge more smoothly.
Data Augmentation	Real-time augmentation was applied to increase dataset diversity and improve model generalization. Techniques included random horizontal flips, small rotations (±20°), width and height shifts (±10%), and zoom up to 20%.
Denoising	The images were relatively clean, so no dedicated denoising filter was necessary. (Optional denoising methods such as Gaussian blur or Non-local Means could be added if noise is detected.)

Edge Detection	Edge detection was not required because convolutional layers in the CNN automatically learn edge features.
Color Space Conversion	All images were kept in the **RGB** color space, which is standard for training CNNs on natural images.
Image Cropping	Explicit cropping was not performed. Resizing operations inherently center and crop when needed to maintain aspect ratio.
Batch Normalization	Batch Normalization layers were included after convolutional layers in the CNN to stabilize learning, speed convergence, and improve overall accuracy.

Data Preprocessing Code Screenshots

Loading Data

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator


train_datagen = ImageDataGenerator(
    rescale=1.0/255.0,              # normalize pixel values
    rotation_range=30,              # random rotations
    width_shift_range=0.2,          # horizontal shift
    height_shift_range=0.2,         # vertical shift
    zoom_range=0.2,                 # zoom in/out
    brightness_range=[0.8, 1.2],    # vary brightness
    horizontal_flip=True,           # random horizontal flips
    vertical_flip=True              # random vertical flips
)


test_datagen = ImageDataGenerator(rescale=1.0/255.0)


train_dir = "/kaggle/input/mushroom/Dataset/train"
test_dir  = "/kaggle/input/mushroom/Dataset/test"

# Train generator
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),   # resize images
    batch_size=32,
    class_mode="binary"       # use 'categorical' if more than 2 classes
)
```

Resizing	target_size=(224, 224)
Normalization	rescale=1./255
Data Augmentation	shear_range=0.2, zoom_range=0.2,

	horizontal_flip=True
Denoising	-
Edge Detection	-
Color Space Conversion	-
Image Cropping	-
Batch Normalization	-