Data Collection and Preprocessing Phase

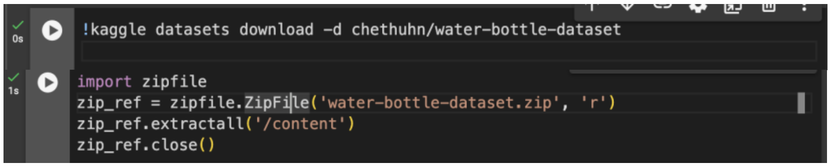
Date	20 July 2024
Team ID	SWTID1720082030
Project Title	Hydration Essentials: Classifying Water Bottle Image
Maximum Marks	6 Marks

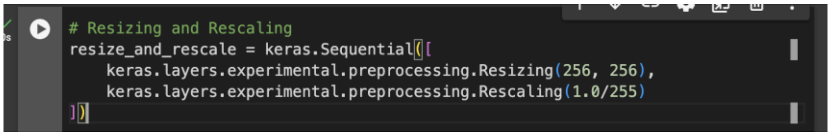
Preprocessing Template

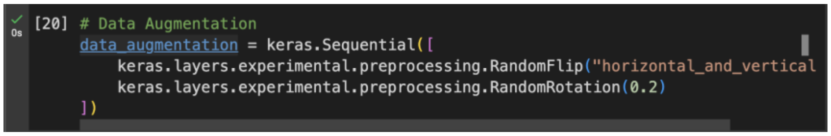
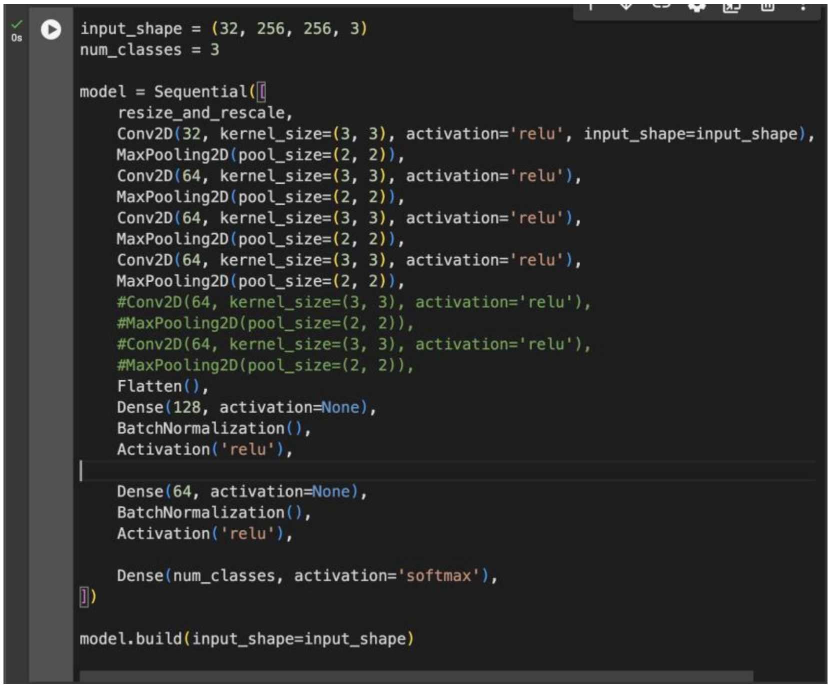
The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network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

Section	**Description**
Data Overview	There are three types of image datasets we have used here , full-water, half-water, and overflowi
Resizing	Add a resizing layer to the model. Resize all input images to a size of 256x256 pixels.
Normalization	Min-max normalization scales numerical values to a specific range, typically between 0 and 1. In this case, dividing each pixel value by 255 maps the original pixel values (which range from 0 to 255) to a new range of 0 to 1

Data Augmentation	The provided code implements two common data augmentation techniques Random flipping**:** Images are randomly flipped horizontally or vertically to improve the model's ability to recognise objects regardless of their orientation. Random rotation: Images are randomly rotated by up to 20 degrees to increase the model's robustness to variations in object orientation.
Batch Normalization	The provided code incorporates Batch Normalization (BN) at two specific points: **1.** After the first Dense layer (128 units)**:** Here, BN is applied to the output of the first fully-connected layer. This layer transforms the input data from a spatial representation (images) to a flattened vector. BN helps to normalize the activations in this vector, improving the training process for the subsequent layers. **2.** After the second Dense layer (64 units): Similar to the first instance, BN normalizes the activations in the second fully-connected layer before feeding them to the final output layer.
Data Preprocessing Code Screenshots	

Loading Data	
	`!kaggle datasets download -d chethuhn/water-bottle-dataset`
	`import zipfile`
	`zip_ref = zipfile.ZipFile('water-bottle-dataset.zip', 'r')`
	`zip_ref.extractall('/content')`
	`zip_ref.close()`

Resizing, Normalisation	
	`# Resizing and Rescaling`
	`resize_and_rescale = keras.Sequential([`
	`    keras.layers.experimental.preprocessing.Resizing(256, 256),`
	`    keras.layers.experimental.preprocessing.Rescaling(1.0/255)`
	`])`

Data Augmentation	
Batch Normalization	

```
[20] # Data Augmentation
     data_augmentation = keras.Sequential([
         keras.layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical
         keras.layers.experimental.preprocessing.RandomRotation(0.2)
     ])
```

```
input_shape = (32, 256, 256, 3)
num_classes = 3

model = Sequential([
    resize_and_rescale,
    Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=input_shape),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, kernel_size=(3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, kernel_size=(3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, kernel_size=(3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    #Conv2D(64, kernel_size=(3, 3), activation='relu'),
    #MaxPooling2D(pool_size=(2, 2)),
    #Conv2D(64, kernel_size=(3, 3), activation='relu'),
    #MaxPooling2D(pool_size=(2, 2)),
    Flatten(),
    Dense(128, activation=None),
    BatchNormalization(),
    Activation('relu'),

    Dense(64, activation=None),
    BatchNormalization(),
    Activation('relu'),

    Dense(num_classes, activation='softmax'),
])

model.build(input_shape=input_shape)
```