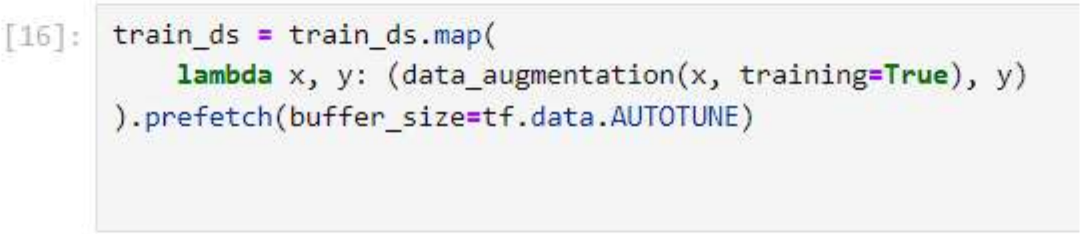
Model Development Phase Template

Date	20 July 2024
Team ID	SWTID1720163161
Project Title	Hydration Essentials: Classifying Water Bottle Images
Maximum Marks	10 Marks

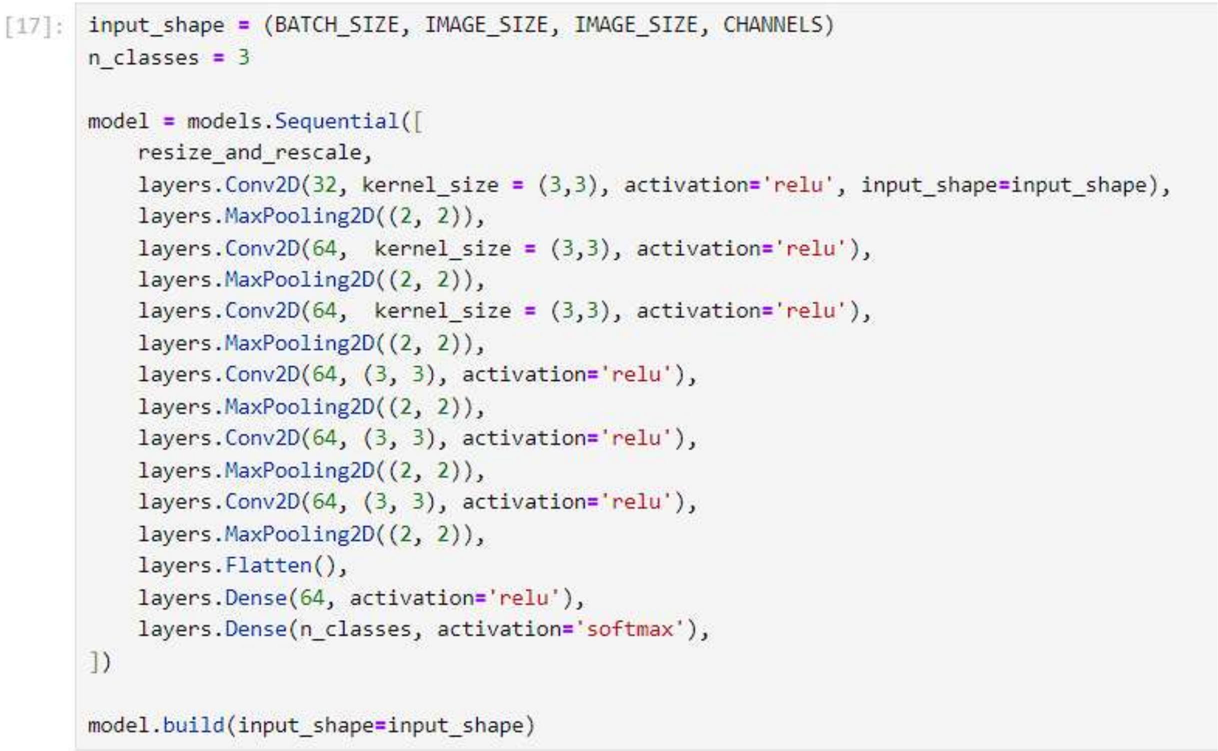
Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

Initial Model Training Code (5 marks):



```
[16]: train_ds = train_ds.map(
          lambda x, y: (data_augmentation(x, training=True), y)
      ).prefetch(buffer_size=tf.data.AUTOTUNE)
```



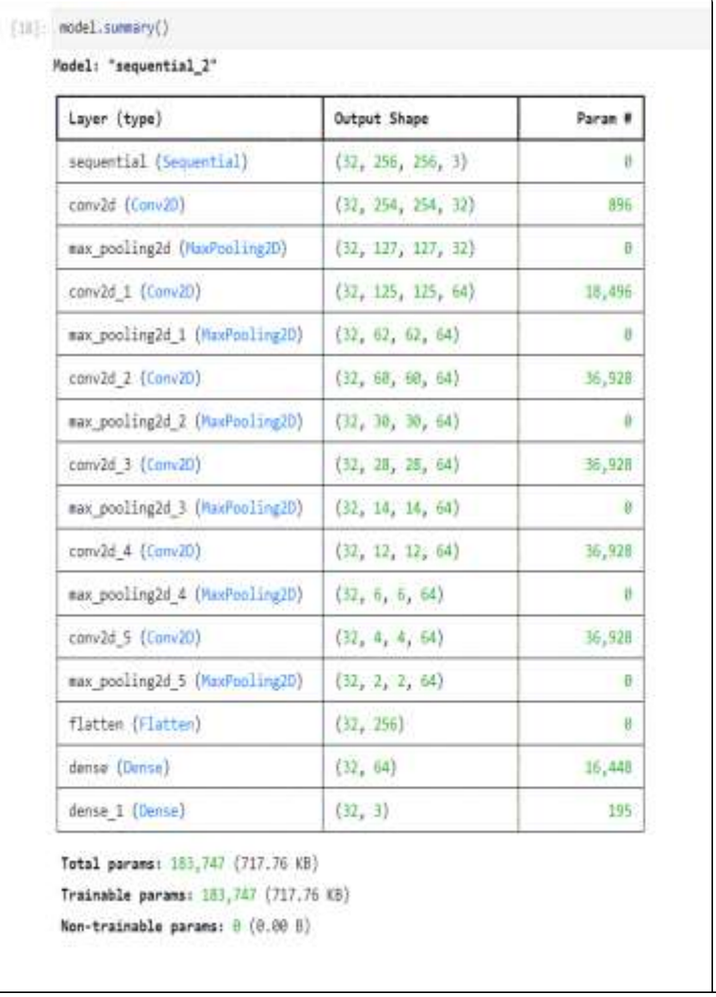
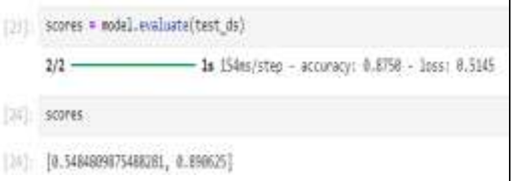
[17]:

```
input_shape = (BATCH_SIZE, IMAGE_SIZE, IMAGE_SIZE, CHANNELS)
n_classes = 3

model = models.Sequential([
    resize_and_rescale,
    layers.Conv2D(32, kernel_size = (3,3), activation='relu', input_shape=input_shape),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64,  kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64,  kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(n_classes, activation='softmax'),
])

model.build(input_shape=input_shape)
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
Model 1 (CNN)		

```
[18]: model.summary()

Model: "sequential_2"

┌──────────────────────────────────┬────────────────────────┬──────────┐
│ Layer (type)                     │ Output Shape           │  Param # │
├──────────────────────────────────┼────────────────────────┼──────────┤
│ sequential (Sequential)          │ (32, 256, 256, 3)      │        0 │
│ conv2d (Conv2D)                  │ (32, 254, 254, 32)     │      896 │
│ max_pooling2d (MaxPooling2D)     │ (32, 127, 127, 32)     │        0 │
│ conv2d_1 (Conv2D)                │ (32, 125, 125, 64)     │   18,496 │
│ max_pooling2d_1 (MaxPooling2D)   │ (32, 62, 62, 64)       │        0 │
│ conv2d_2 (Conv2D)                │ (32, 60, 60, 64)       │   36,928 │
│ max_pooling2d_2 (MaxPooling2D)   │ (32, 30, 30, 64)       │        0 │
│ conv2d_3 (Conv2D)                │ (32, 28, 28, 64)       │   36,928 │
│ max_pooling2d_3 (MaxPooling2D)   │ (32, 14, 14, 64)       │        0 │
│ conv2d_4 (Conv2D)                │ (32, 12, 12, 64)       │   36,928 │
│ max_pooling2d_4 (MaxPooling2D)   │ (32, 6, 6, 64)         │        0 │
│ conv2d_5 (Conv2D)                │ (32, 4, 4, 64)         │   36,928 │
│ max_pooling2d_5 (MaxPooling2D)   │ (32, 2, 2, 64)         │        0 │
│ flatten (Flatten)                │ (32, 256)              │        0 │
│ dense (Dense)                    │ (32, 64)               │   16,448 │
│ dense_1 (Dense)                  │ (32, 3)                │      195 │
└──────────────────────────────────┴────────────────────────┴──────────┘

 Total params: 183,747 (717.76 KB)
 Trainable params: 183,747 (717.76 KB)
 Non-trainable params: 0 (0.00 B)
```

```
[23]: scores = model.evaluate(test_ds)

2/2 ━━━━━━━━━━━━━━━━━━━━ 1s 154ms/step - accuracy: 0.8750 - loss: 0.5145

[24]: scores

[24]: [0.5484809875488281, 0.890625]
```