Development Phase

Delivery of Sprint 2

Date	5 November 2022
Team ID	PNT2022TMID36413
Project Name	AI-poweredNutritionAnalyserforFitnessEnthusiasts65GP

Import libraries:

```
import numpy as np
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout

from keras.preprocessing.image import ImageDataGenerator
```

Initialize the model:

```
[ ] # Initializing the CNN
    classifier = Sequential()
```

CNN layers:

```
# Initializing the CNN
classifier = Sequential()

# First convolution layer and pooling
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Second convolution layer and pooling
classifier.add(Conv2D(32, (3, 3), activation='relu'))

# input_shape is going to be the pooled feature maps from the previous convolution layer
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Flattening the layers
classifier.add(Flatten())
```

Dense layer:

```
# Adding a fully connected layer
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=5, activation='softmax')) # softmax for more than 2
```

Configure learning process:

```
classifier.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

Train the model:

```
classifier.fit_generator(
        generator=x_train,steps_per_epoch = len(x_train),
        epochs=20, validation_data=x_test,validation_steps = len(x_test))# No of images in test set
```

Save model:

Saving our model

```
classifier.save('nutrition.h5')
```

Test the model:

Importing libraries for testing

```
#importing libraries for model testing

import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

Load model for testing

```
model = load_model('/content/nutrition.h5')
```

Prediction process

```
img = image.load_img('/content/Images.jpg', target_size=(64, 64))
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
pred = np.argmax(model.predict(x), axis=1)
index = ['APPLES', 'BANANA', 'ORANGE', 'PINEAPPLE', 'WATERMELON']
result = str(index[pred[0]])
x = result
print(x)
```

```
1/1 [==============================] - 0s 22ms/step
BANANA
```