MODEL BUILDING LIBRARIES

DATE	16 Nov 2022
TEAM ID	PNT2022TMID16818
PROJECT NAME	AI-powered Nutrition Analyzer for Fitness Enthusiasts
MARK	6

Model Building

- Importing The Model Building Libraries

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

- Initializing The Model

```
model = Sequential()
```

- Adding 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
layerclassifier.add(MaxPooling2D(pool_size=(2, 2)))
# Flattening the layers
classifier.add(Flatten())
```

- Adding Dense Layers

```
classifier.add(Dense(units=128, activation='relu')) classifier.add(Dense(units=5, activation='softmax'))
```

```
#summary of
our model
classifi
er.summa ry()
```

```
Model: "sequential_1"

 Layer (type)                      Output Shape              Param #
=================================================================
conv2d (Conv2D)                  (None, 62, 62, 32)           896
max_pooling2d (MaxPooling2D )      (None, 31, 31, 32)         0
conv2d_1 (Conv2D)                (None, 29, 29, 32)           9248
max_pooling2d_1 (MaxPooling
2D)                                (None, 14, 14, 32)         0
flatten (Flatten)                (None, 6272)                 0
dense (Dense)                    (None, 128)                  802944
dense_1 (Dense)                  (None, 5)                    645

=================================================================
Total params: 813,733
Trainable params: 813,733
Non-trainable params: 0
```

- Configure The Learning Process

```
# Compiling the CNN
# categorical_crossentropy for more than 2
classifier.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['acc
```

- Train The Model

```
#Fitting the model
classifier.fit_generator(generator=x_train,steps_per_epoch = len(x_train),epochs=20, valid
```

```
Epoch 1/20
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: UserWarning: `Model.
```

824/824 [========================== ====] Epoch 2/20	- 21 s	16ms/st ep	– loss:	0.617 2	accuracy:
824/824 [========================== ====]	- 13 s	15ms/st ep	- loss:	0.411 5	accuracy:
Epoch 3/20 824/824 [============================	- 13 s	16ms/st ep	- loss:	0.376 6	accuracy:
Epoch 4/20 824/824 [========================== ====]	- 13 s	16ms/st ep	- loss:	0.348 4	accuracy:
Epoch 5/20					
824/824 [========================== ====] Epoch 6/20	- 13 s	16ms/st ep	– loss:	0.324 3	accuracy:
824/824 [========================== ====]	- 13 s	16ms/st ep	- loss:	0.324 0	accuracy:

Epoch 7/20 824/824 [=========================	- 13s	16ms/step	- loss:	0.2887	accuracy:
Epoch 8/20 824/824 [=============================]	- 13s	16ms/step	- loss:	0.2728	accuracy:
Epoch 9/20 824/824 [=============================]	- 13s	16ms/step	- loss:	0.2717	accuracy:
Epoch 10/20					
824/824 [=============================]	- 14s	17ms/step	- loss:	0.2365	accuracy:
Epoch 11/20 824/824 [=============================]	- 13s	15ms/step	- loss:	0.2301	accuracy:
Epoch 12/20 824/824 [=========================	- 13s	15ms/step	- loss:	0.2083	accuracy:
Epoch 13/20					
824/824 [=============================]	- 13s	15ms/step	- loss:	0.2049	accuracy:
Epoch 14/20					
824/824 [=============================]	- 12s	15ms/step	- loss:	0.1930	accuracy:

Epoch 15/20 824/824 [========================== ====]	- 13s	15ms/step	- loss:	0.1807	accuracy:
Epoch 16/20 824/824 [========================== ====]	- 13s	15ms/step	- loss:	0.1712	accuracy:
Epoch 17/20					
824/824 [========================== ====]	- 13s	15ms/step	- loss:	0.1599	accuracy:
Epoch 18/20 824/824 [========================== ====]	- 13s	15ms/step	- loss:	0.1619	accuracy:
Epoch 19/20 824/824 [========================== ====]	- 13s	15ms/step	- loss:	0.1505	accuracy:
Epoch 20/20					
824/824 [========================== ====]	- 12s	15ms/step	- loss:	0.1211	accuracy:

```
<keras.callbacks.History at 0x7fd655833d90>
```

- Saving The Model

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

- Testing The Model

```
#Predict the results from
tensorflow.keras.models import
load_modelfrom keras.preprocessing import
image model = load_model("nutrition.h5")

from tensorflow.keras.utils import
img_to_array#loading of the image img =
load_img(r'/content/Sample_Images/Test_Image1.jpg',grayscale=False,target_siz e= (64,#image to array
x = img_to_a rray(img
)
#changin g
the shape
x = np.expand_dims(x,axis = 0)
```

```
predict_x=model.predict(x)
     classes_x=np.argmax(predi
     ct_x,axis=-1)classes_x
```

```
1/1 [==============================] - 0s 18ms/step array([0])
```

```
index=['APPLES', 'BANANA', 'ORANGE','PINEAPPLE','WATERMELON']
result=str(index[c lasses_x[0]]) result
```

```
'APPLES'
```

Colab HYPERLINK "https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links" HYPERLINK "https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links"paid HYPERLINK "https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links" HYPERLINK

"https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links"products - Cancel HYPERLINK "https://colab.research.google.com/cancel-subscription" HYPERLINK "https://colab.research.google.com/cancel-subscription"contracts HYPERLINK "https://colab.research.google.com/cancel-subscription" HYPERLINK "https://colab.research.google.com/cancel-subscription"here