

# MODEL BUILDING LIBRARIES

DATE	12 Nov 2022
TEAM ID	PNT2022TMID50029
PROJEET 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 Sequential
from tensorflow.keras.layers import Dense, Flatten, 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
layer
classifier.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
	(None, 14, 14, 32)	0
max_pooling2d_1 (MaxPooling2D)		
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 [=====] ====]	- 16ms/st 21 ep s	- loss:		0.617 2	accuracy:
Epoch 2/20 824/824 [=====] ====]	- 15ms/st 13 ep s	- loss:		0.411 5	accuracy:
Epoch 3/20 824/824 [=====] =====	16ms/st ep			0.376 6	
- 13 s		- loss:			accuracy:
Epoch 4/20 824/824 [=====] =====]	- 16ms/st 13 ep s	- loss:		0.348 4	accuracy:
Epoch 5/20					

824/824 [=====] ====]	- 16ms/st 13 ep s	- loss:		0.324 3	accuracy:
Epoch 6/20 824/824 [=====] =====]	- 16ms/st 13 ep s	- loss:		0.324 0	accuracy:

Epoch 7/20 824/824 [=====]	16ms/st ep		0.288 7	
- 13 s		- loss:		accuracy:
Epoch 8/20 824/824 [=====]	- 16ms/st 13 ep s	- loss:	0.272 8	accuracy:
Epoch 9/20 824/824 [=====]	- 16ms/st 13 ep s	- loss:	0.271 7	accuracy:
Epoch 10/20				
824/824 [=====]	- 17ms/st 14 ep s	- loss:	0.236 5	accuracy:
Epoch 11/20 824/824 [=====]	- 15ms/st 13 ep s	- loss:	0.230 1	accuracy:
Epoch 12/20 824/824 [=====]	15ms/st ep		0.208 3	
- 13 s		- loss:		accuracy:
Epoch 13/20				
824/824 [=====]	- 15ms/st 13 ep s	- loss:	0.204 9	accuracy:
Epoch 14/20				
824/824 [=====]	- 15ms/st 12 ep s	- loss:	0.193 0	accuracy:

Epoch 15/20 824/824 [=====] =====]	- 15ms/st 13 ep s	- loss:	0.180 7	accuracy:
Epoch 16/20 824/824 [=====] =====]	- 15ms/st 13 ep s	- loss:	0.171 2	accuracy:
Epoch 17/20 824/824 [=====] =====]	- 15ms/st 13 ep s	- loss:	0.159 9	accuracy:
Epoch 18/20 824/824 [=====] =====]	- 15ms/st 13 ep s	- loss:	0.161 9	accuracy:
Epoch 19/20 824/824 [=====] =====]	- 15ms/st 13 ep s	- loss:	0.150 5	accuracy:
Epoch 20/20 824/824 [=====] =====]	- 15ms/st 12 ep s	- loss:	0.121 1	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_size = (64,#image to array
x = img_to_array(img
)
#changing
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](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](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](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"](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"](https://colab.research.google.com/cancel-subscription) HYPERLINK

["https://colab.research.google.com/cancel-subscription"](https://colab.research.google.com/cancel-subscription)[contracts](#)  
HYPERLINK

["https://colab.research.google.com/cancel-subscription"](https://colab.research.google.com/cancel-subscription) HYPERLINK

["https://colab.research.google.com/cancel-subscription"](https://colab.research.google.com/cancel-subscription)[here](#)