Project Development Phase
Model Performance Test

Date	17 November 2022
Team ID	PNT2022TMID19068
Project Name	Fertilizers Recommendation System for Disease Prediction
Maximum Marks	10 Marks

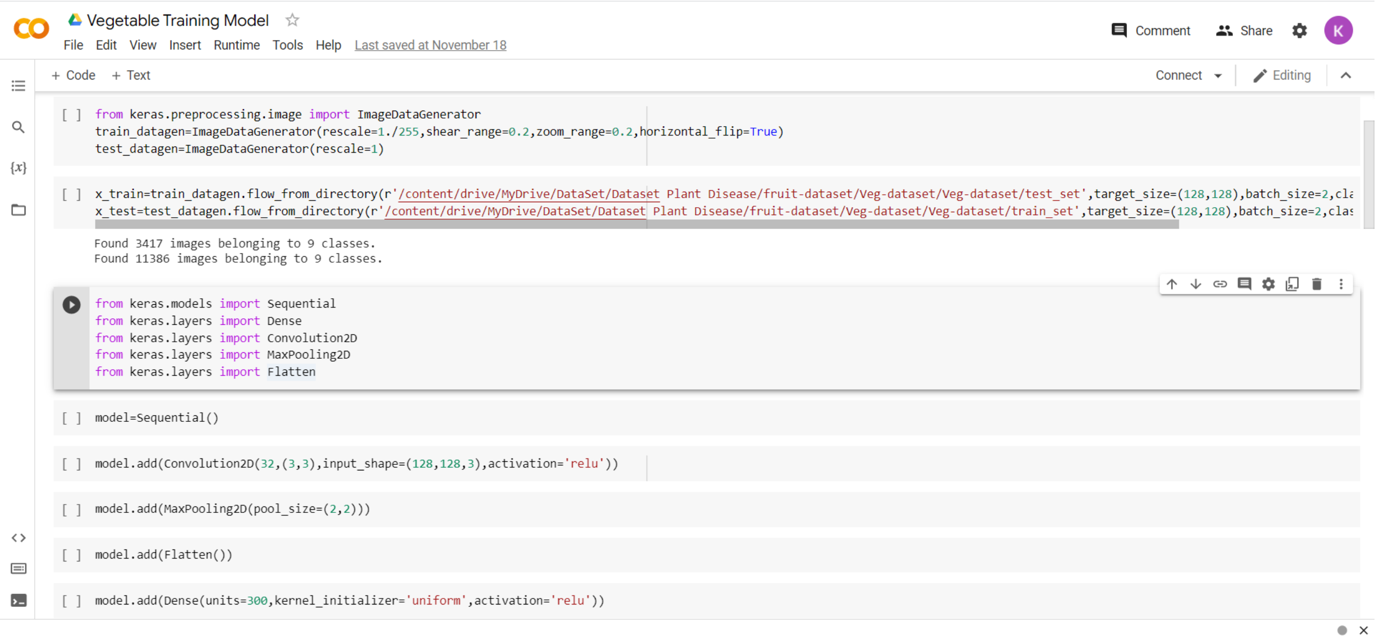
Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Total params: 10,227,737 Trainable params: 10,227,737 Non-trainable params: 0	Attached below
2.	Accuracy	Training Accuracy - 100 Validation Accuracy - 100	Attached below
3.	Confidence Score (Only Yolo Projects)	Class Detected - 15 Confidence Score - NILL	NILL

SCREENSHOTS :

Veg:



```
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1)
```

```
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/Veg-dataset/Veg-dataset/test_set',target_size=(128,128),batch_size=2,cla
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/Veg-dataset/Veg-dataset/train_set',target_size=(128,128),batch_size=2,clas
```

```
Found 3417 images belonging to 9 classes.
Found 11386 images belonging to 9 classes.
```

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
```

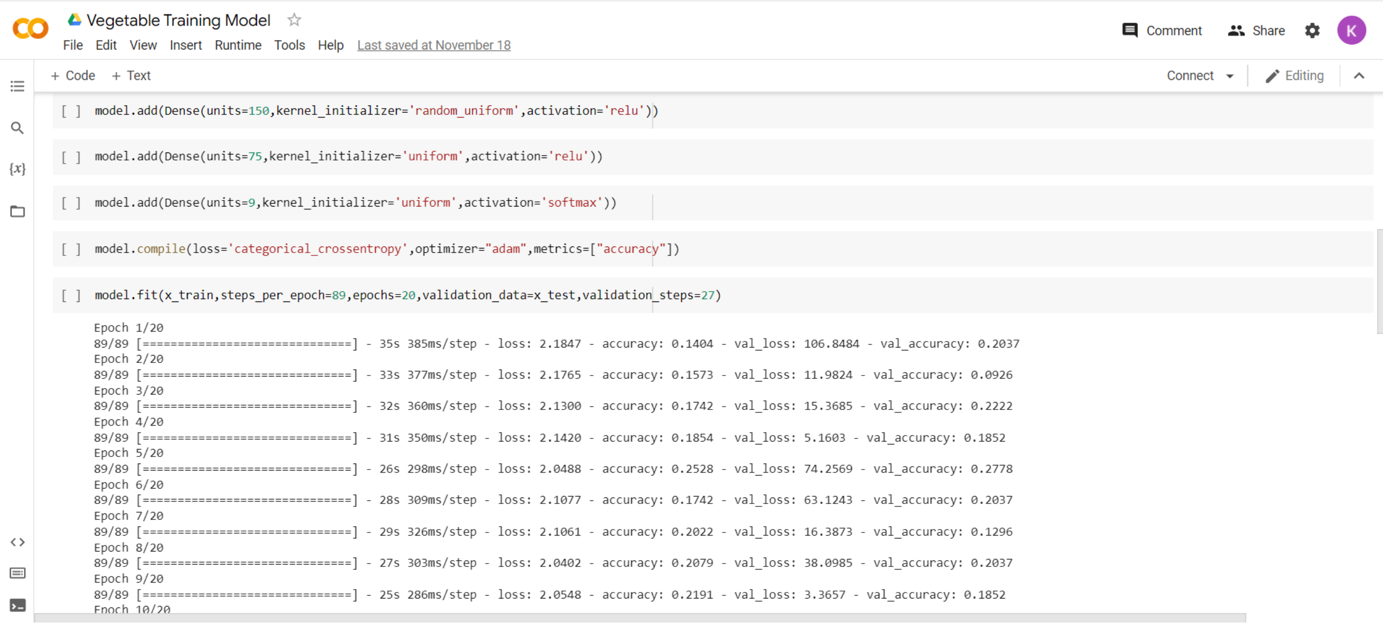
```
model=Sequential()
```

```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```

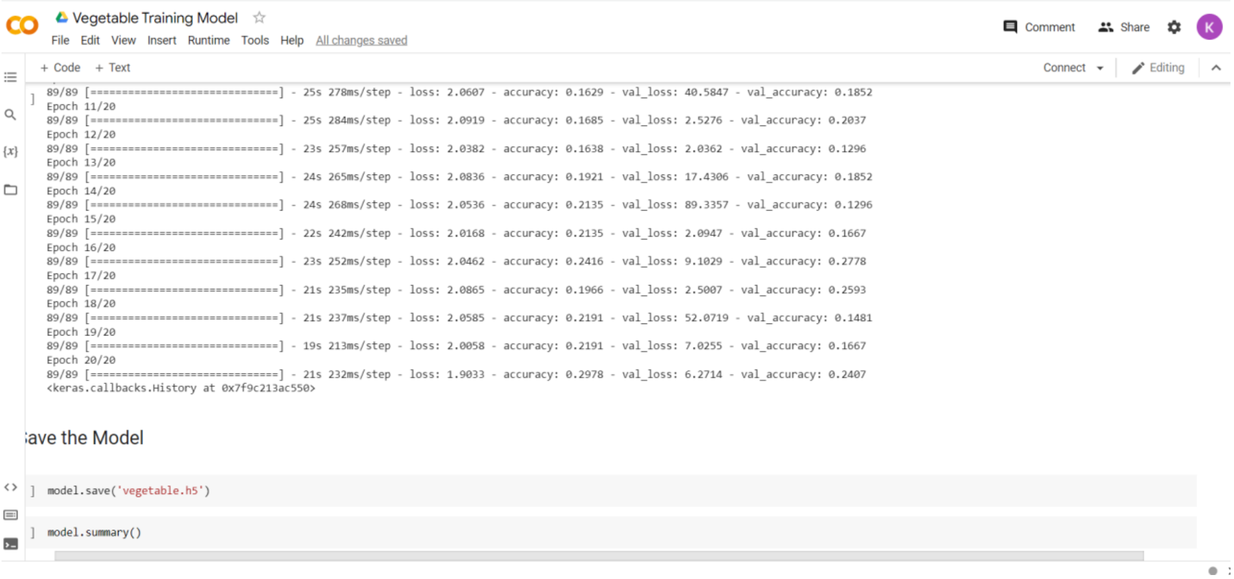
```
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.add(Flatten())
```

```
model.add(Dense(units=300,kernel_initializer='uniform',activation='relu'))
```


Vegetable Training Model
File Edit View Insert Runtime Tools Help Last saved at November 18
Comment
Share
K
+ Code + Text
Connect
Editing

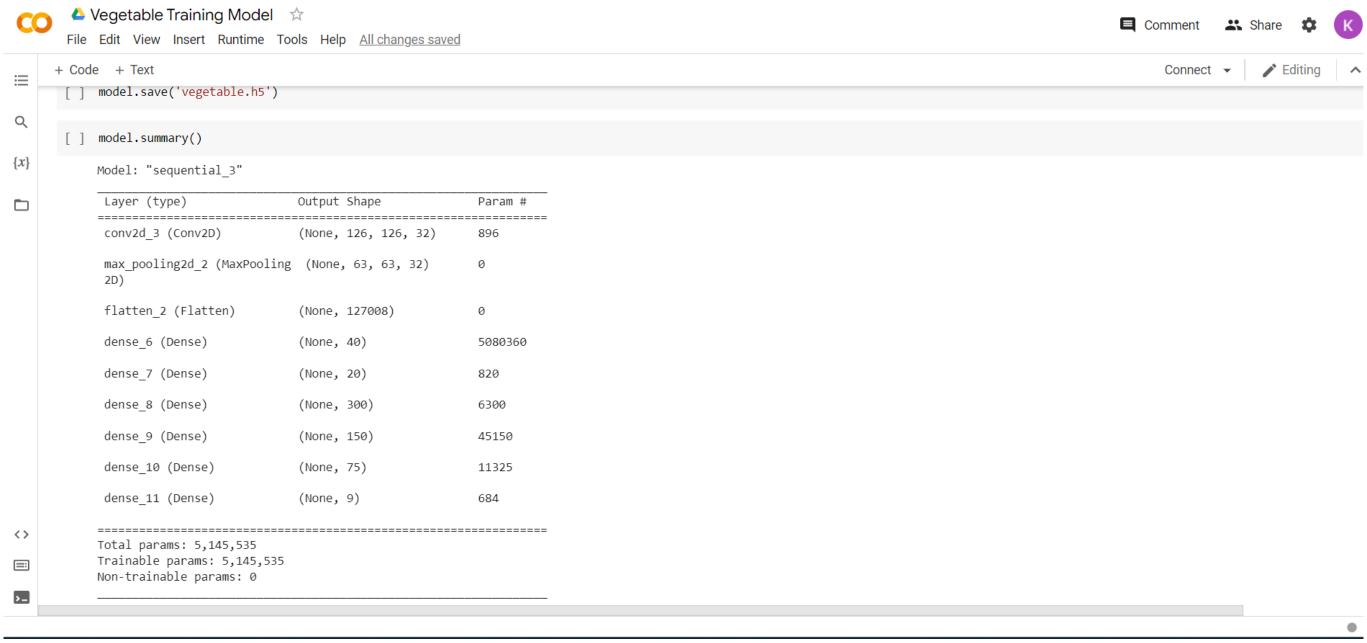
```
[ ] model.add(Dense(units=150,kernel_initializer='random_uniform',activation='relu'))

[ ] model.add(Dense(units=75,kernel_initializer='uniform',activation='relu'))

[ ] model.add(Dense(units=9,kernel_initializer='uniform',activation='softmax'))

[ ] model.compile(loss='categorical_crossentropy',optimizer="adam",metrics=["accuracy"])

[ ] model.fit(x_train,steps_per_epoch=89,epochs=20,validation_data=x_test,validation_steps=27)

    Epoch 1/20
    89/89 [==============================] - 35s 385ms/step - loss: 2.1847 - accuracy: 0.1404 - val_loss: 106.8484 - val_accuracy: 0.2037
    Epoch 2/20
    89/89 [==============================] - 33s 377ms/step - loss: 2.1765 - accuracy: 0.1573 - val_loss: 11.9824 - val_accuracy: 0.0926
    Epoch 3/20
    89/89 [==============================] - 32s 360ms/step - loss: 2.1300 - accuracy: 0.1742 - val_loss: 15.3685 - val_accuracy: 0.2222
    Epoch 4/20
    89/89 [==============================] - 31s 350ms/step - loss: 2.1420 - accuracy: 0.1854 - val_loss: 5.1603 - val_accuracy: 0.1852
    Epoch 5/20
    89/89 [==============================] - 26s 298ms/step - loss: 2.0488 - accuracy: 0.2528 - val_loss: 74.2569 - val_accuracy: 0.2778
    Epoch 6/20
    89/89 [==============================] - 28s 309ms/step - loss: 2.1077 - accuracy: 0.1742 - val_loss: 63.1243 - val_accuracy: 0.2037
    Epoch 7/20
    89/89 [==============================] - 29s 326ms/step - loss: 2.1061 - accuracy: 0.2022 - val_loss: 16.3873 - val_accuracy: 0.1296
    Epoch 8/20
    89/89 [==============================] - 27s 303ms/step - loss: 2.0402 - accuracy: 0.2079 - val_loss: 38.0985 - val_accuracy: 0.2037
    Epoch 9/20
    89/89 [==============================] - 25s 286ms/step - loss: 2.0548 - accuracy: 0.2191 - val_loss: 3.3657 - val_accuracy: 0.1852
    Epoch 10/20
```



```
]   89/89 [==============================] - 25s 278ms/step - loss: 2.0607 - accuracy: 0.1629 - val_loss: 40.5847 - val_accuracy: 0.1852
    Epoch 11/20
    89/89 [==============================] - 25s 284ms/step - loss: 2.0919 - accuracy: 0.1685 - val_loss: 2.5276 - val_accuracy: 0.2037
    Epoch 12/20
    89/89 [==============================] - 23s 257ms/step - loss: 2.0382 - accuracy: 0.1638 - val_loss: 2.0362 - val_accuracy: 0.1296
    Epoch 13/20
    89/89 [==============================] - 24s 265ms/step - loss: 2.0836 - accuracy: 0.1921 - val_loss: 17.4306 - val_accuracy: 0.1852
    Epoch 14/20
    89/89 [==============================] - 24s 268ms/step - loss: 2.0536 - accuracy: 0.2135 - val_loss: 89.3357 - val_accuracy: 0.1296
    Epoch 15/20
    89/89 [==============================] - 22s 242ms/step - loss: 2.0168 - accuracy: 0.2135 - val_loss: 2.0947 - val_accuracy: 0.1667
    Epoch 16/20
    89/89 [==============================] - 23s 252ms/step - loss: 2.0462 - accuracy: 0.2416 - val_loss: 9.1029 - val_accuracy: 0.2778
    Epoch 17/20
    89/89 [==============================] - 21s 235ms/step - loss: 2.0865 - accuracy: 0.1966 - val_loss: 2.5007 - val_accuracy: 0.2593
    Epoch 18/20
    89/89 [==============================] - 21s 237ms/step - loss: 2.0585 - accuracy: 0.2191 - val_loss: 52.0719 - val_accuracy: 0.1481
    Epoch 19/20
    89/89 [==============================] - 19s 213ms/step - loss: 2.0058 - accuracy: 0.2191 - val_loss: 7.0255 - val_accuracy: 0.1667
    Epoch 20/20
    89/89 [==============================] - 21s 232ms/step - loss: 1.9033 - accuracy: 0.2978 - val_loss: 6.2714 - val_accuracy: 0.2407
    <keras.callbacks.History at 0x7f9c213ac550>
```

ave the Model

```
] model.save('vegetable.h5')

] model.summary()
```

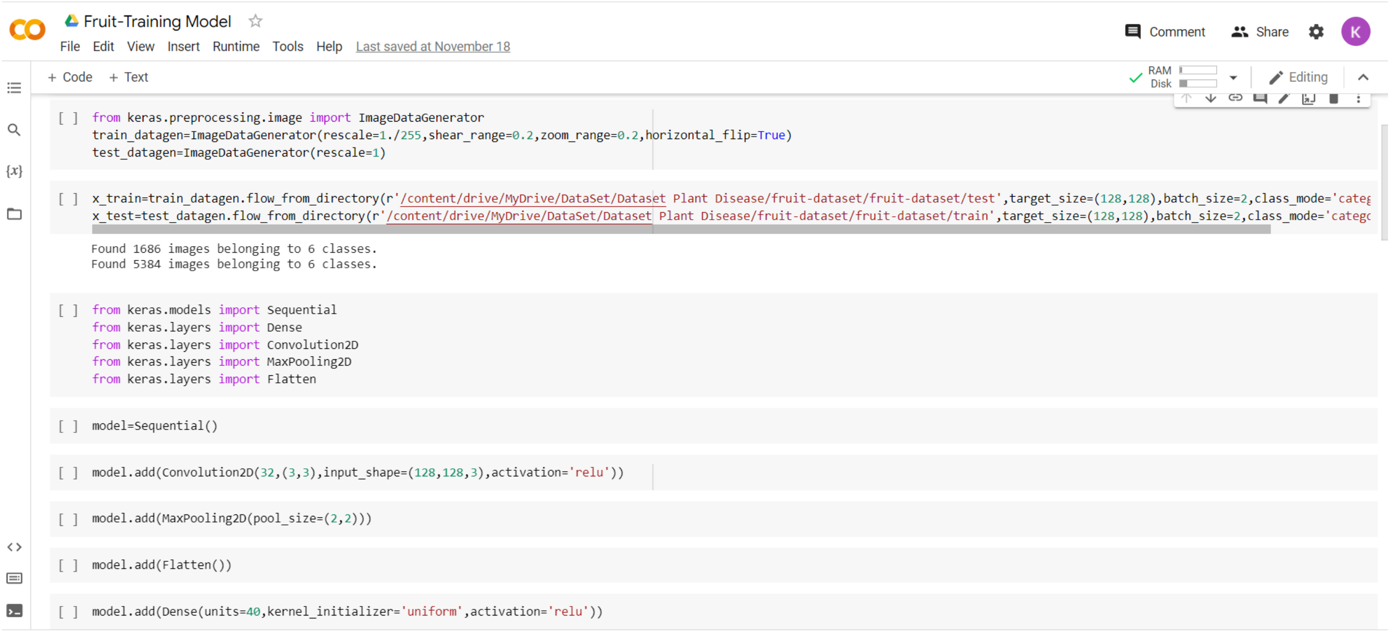


```
model.save('vegetable.h5')
```

```
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)           (None, 126, 126, 32)      896

 max_pooling2d_2 (MaxPooling  (None, 63, 63, 32)       0
 2D)

 flatten_2 (Flatten)         (None, 127008)            0

 dense_6 (Dense)             (None, 40)                5080360

 dense_7 (Dense)             (None, 20)                820

 dense_8 (Dense)             (None, 300)               6300

 dense_9 (Dense)             (None, 150)               45150

 dense_10 (Dense)            (None, 75)                11325

 dense_11 (Dense)            (None, 9)                 684

=================================================================
Total params: 5,145,535
Trainable params: 5,145,535
Non-trainable params: 0
_________________________________________________________________
```

Fruit:


Fruit-Training Model
File Edit View Insert Runtime Tools Help Last saved at November 18
Comment Share
+ Code + Text
RAM
Disk
Editing

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1)
```

```
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/fruit-dataset/test',target_size=(128,128),batch_size=2,class_mode='categ
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/fruit-dataset/train',target_size=(128,128),batch_size=2,class_mode='catego
```

```
Found 1686 images belonging to 6 classes.
Found 5384 images belonging to 6 classes.
```

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
```

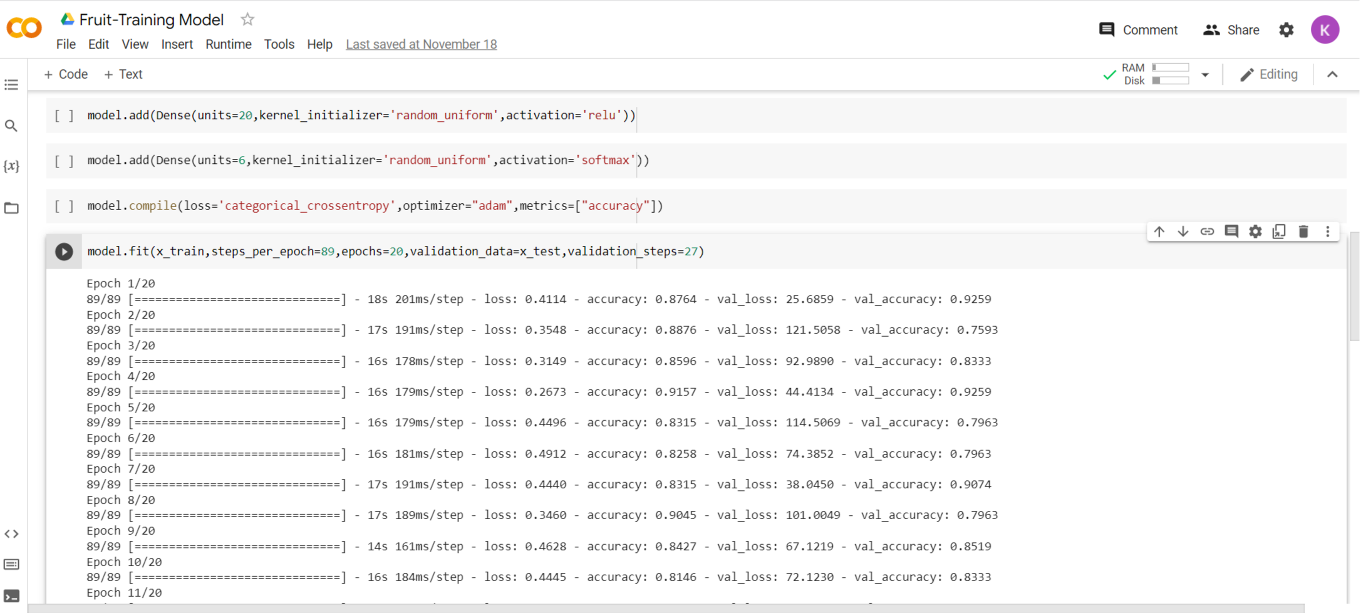
```
model=Sequential()
```

```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```

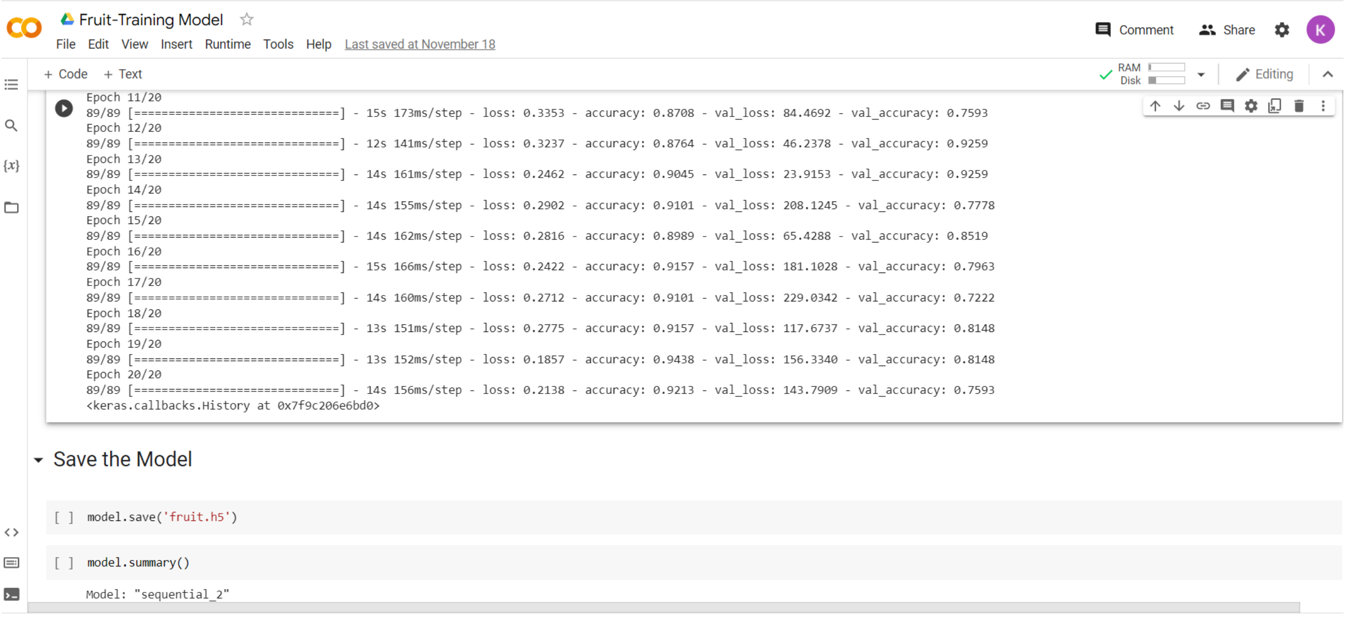
```
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.add(Flatten())
```

```
model.add(Dense(units=40,kernel_initializer='uniform',activation='relu'))
```


Fruit-Training Model
File Edit View Insert Runtime Tools Help Last saved at November 18
Comment Share
+ Code + Text
RAM
Disk
Editing

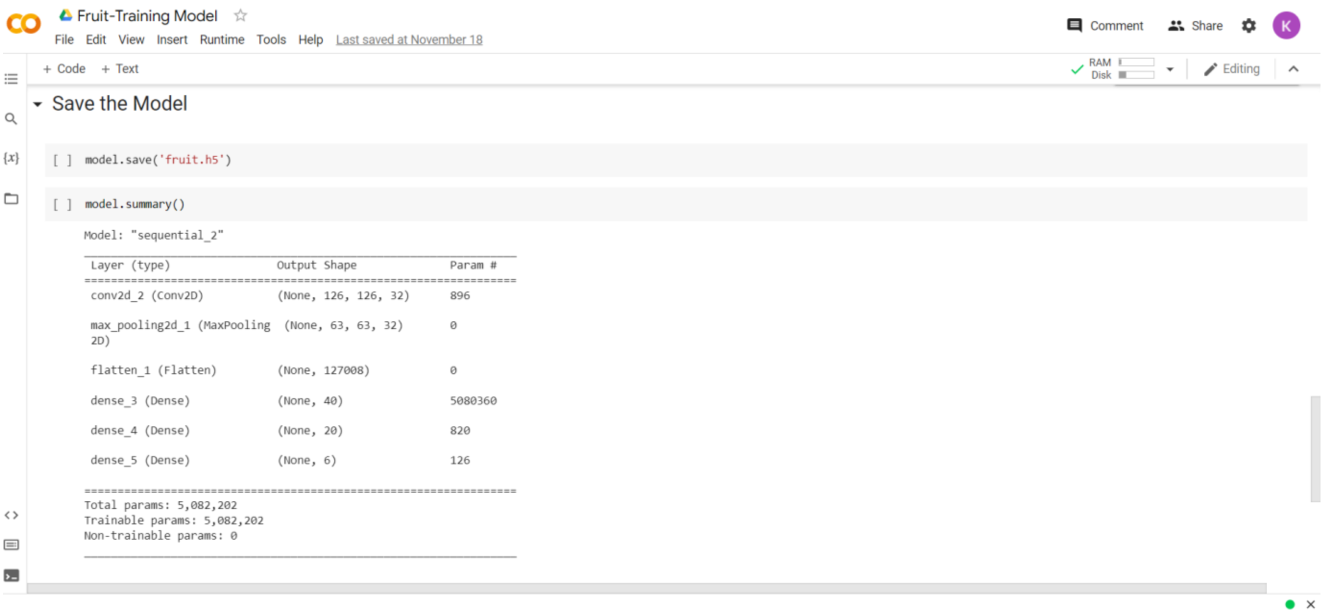
```
[ ] model.add(Dense(units=20,kernel_initializer='random_uniform',activation='relu'))

[ ] model.add(Dense(units=6,kernel_initializer='random_uniform',activation='softmax'))

[ ] model.compile(loss='categorical_crossentropy',optimizer="adam",metrics=["accuracy"])

model.fit(x_train,steps_per_epoch=89,epochs=20,validation_data=x_test,validation_steps=27)

Epoch 1/20
89/89 [==============================] - 18s 201ms/step - loss: 0.4114 - accuracy: 0.8764 - val_loss: 25.6859 - val_accuracy: 0.9259
Epoch 2/20
89/89 [==============================] - 17s 191ms/step - loss: 0.3548 - accuracy: 0.8876 - val_loss: 121.5058 - val_accuracy: 0.7593
Epoch 3/20
89/89 [==============================] - 16s 178ms/step - loss: 0.3149 - accuracy: 0.8596 - val_loss: 92.9890 - val_accuracy: 0.8333
Epoch 4/20
89/89 [==============================] - 16s 179ms/step - loss: 0.2673 - accuracy: 0.9157 - val_loss: 44.4134 - val_accuracy: 0.9259
Epoch 5/20
89/89 [==============================] - 16s 179ms/step - loss: 0.4496 - accuracy: 0.8315 - val_loss: 114.5069 - val_accuracy: 0.7963
Epoch 6/20
89/89 [==============================] - 16s 181ms/step - loss: 0.4912 - accuracy: 0.8258 - val_loss: 74.3852 - val_accuracy: 0.7963
Epoch 7/20
89/89 [==============================] - 17s 191ms/step - loss: 0.4440 - accuracy: 0.8315 - val_loss: 38.0450 - val_accuracy: 0.9074
Epoch 8/20
89/89 [==============================] - 17s 189ms/step - loss: 0.3460 - accuracy: 0.9045 - val_loss: 101.0049 - val_accuracy: 0.7963
Epoch 9/20
89/89 [==============================] - 14s 161ms/step - loss: 0.4628 - accuracy: 0.8427 - val_loss: 67.1219 - val_accuracy: 0.8519
Epoch 10/20
89/89 [==============================] - 16s 184ms/step - loss: 0.4445 - accuracy: 0.8146 - val_loss: 72.1230 - val_accuracy: 0.8333
Epoch 11/20
```


Fruit-Training Model
File Edit View Insert Runtime Tools Help Last saved at November 18
Comment Share
+ Code + Text
RAM
Disk
Editing

```
Epoch 11/20
89/89 [==============================] - 15s 173ms/step - loss: 0.3353 - accuracy: 0.8708 - val_loss: 84.4692 - val_accuracy: 0.7593
Epoch 12/20
89/89 [==============================] - 12s 141ms/step - loss: 0.3237 - accuracy: 0.8764 - val_loss: 46.2378 - val_accuracy: 0.9259
Epoch 13/20
89/89 [==============================] - 14s 161ms/step - loss: 0.2462 - accuracy: 0.9045 - val_loss: 23.9153 - val_accuracy: 0.9259
Epoch 14/20
89/89 [==============================] - 14s 155ms/step - loss: 0.2902 - accuracy: 0.9101 - val_loss: 208.1245 - val_accuracy: 0.7778
Epoch 15/20
89/89 [==============================] - 14s 162ms/step - loss: 0.2816 - accuracy: 0.8989 - val_loss: 65.4288 - val_accuracy: 0.8519
Epoch 16/20
89/89 [==============================] - 15s 166ms/step - loss: 0.2422 - accuracy: 0.9157 - val_loss: 181.1028 - val_accuracy: 0.7963
Epoch 17/20
89/89 [==============================] - 14s 160ms/step - loss: 0.2712 - accuracy: 0.9101 - val_loss: 229.0342 - val_accuracy: 0.7222
Epoch 18/20
89/89 [==============================] - 13s 151ms/step - loss: 0.2775 - accuracy: 0.9157 - val_loss: 117.6737 - val_accuracy: 0.8148
Epoch 19/20
89/89 [==============================] - 13s 152ms/step - loss: 0.1857 - accuracy: 0.9438 - val_loss: 156.3340 - val_accuracy: 0.8148
Epoch 20/20
89/89 [==============================] - 14s 156ms/step - loss: 0.2138 - accuracy: 0.9213 - val_loss: 143.7909 - val_accuracy: 0.7593
<keras.callbacks.History at 0x7f9c206e6bd0>
```

Save the Model

```
[ ] model.save('fruit.h5')

[ ] model.summary()

Model: "sequential_2"
```



Fruit-Training Model
File Edit View Insert Runtime Tools Help Last saved at November 18
Comment
Share
K
+ Code + Text
RAM
Disk
Editing

Save the Model

```
[ ] model.save('fruit.h5')
```

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

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 126, 126, 32)      896

 max_pooling2d_1 (MaxPooling  (None, 63, 63, 32)       0
 2D)

 flatten_1 (Flatten)         (None, 127008)            0

 dense_3 (Dense)             (None, 40)                5080360

 dense_4 (Dense)             (None, 20)                820

 dense_5 (Dense)             (None, 6)                 126

=================================================================
Total params: 5,082,202
Trainable params: 5,082,202
Non-trainable params: 0
_________________________________________________________________
```