**Train and save the Model**

|  |  |
| --- | --- |
| Date | 11 November 2022 |
| Team ID | PNT2022TMID18696 |
| Project Name | Fertilizers Recommendation System For Disease Prediction. |

{

"cells": [

{

"cell\_type": "code",

"execution\_count": 1,

"id": "16856fd4",

"metadata": {},

"outputs": [],

"source": [

"from keras.preprocessing.image import ImageDataGenerator\n",

"train\_datagen=ImageDataGenerator(rescale=1./255,shear\_range=0.2,zoom\_range=0.2,horizontal\_f lip=True)\n",

"test\_datagen=ImageDataGenerator(rescale=1)"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"id": "60e48d97",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Found 5384 images belonging to 6 classes.\n",

"Found 1686 images belonging to 6 classes.\n"

]

}

],

"source": [

"x\_train=train\_datagen.flow\_from\_directory(r'C:\\Users\\uma25\\project\\Dataset Plant Disease\\fruit-dataset\\fruit-

dataset\\train',target\_size=(128,128),batch\_size=2,class\_mode='categorical')\n",

"x\_test=test\_datagen.flow\_from\_directory(r'C:\\Users\\uma25\\project\\Dataset Plant Disease\\fruit-dataset\\fruit-

dataset\\test',target\_size=(128,128),batch\_size=2,class\_mode='categorical')"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"id": "5b49fb49",

"metadata": {},

"outputs": [],

"source": [

"from keras.models import Sequential\n",

"from keras.layers import Dense\n",

"from keras.layers import Convolution2D\n",

"from keras.layers import MaxPooling2D\n",

"from keras.layers import Flatten"

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"id": "c9f97db4",

"metadata": {},

"outputs": [],

"source": [

"model=Sequential()"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"id": "469c271d",

"metadata": {},

"outputs": [],

"source": [

"model.add(Convolution2D(32,(3,3),input\_shape=(128,128,3),activation='relu'))"

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"id": "b5d53825",

"metadata": {},

"outputs": [],

"source": [

"model.add(MaxPooling2D(pool\_size=(2,2)))"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"id": "e34afddf",

"metadata": {},

"outputs": [],

"source": [

"model.add(Flatten())"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"id": "28a70d32",

"metadata": {},

"outputs": [],

"source": [

"model.add(Dense(units=40,kernel\_initializer='uniform',activation='relu'))\n",

"model.add(Dense(units=70,kernel\_initializer='random\_uniform',activation='relu'))\n",

"model.add(Dense(units=6,kernel\_initializer='random\_uniform',activation='softmax'))"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"id": "9aed627b",

"metadata": {},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"id": "c99d2ba8",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Epoch 1/3\n",

"168/168 [==============================] - 45s 229ms/step - loss: 1.4802 - accuracy:

0.4315 - val\_loss: 119.8421 - val\_accuracy: 0.5577\n",

"Epoch 2/3\n",

"168/168 [==============================] - 38s 223ms/step - loss: 1.0562 - accuracy:

0.5982 - val\_loss: 107.7073 - val\_accuracy: 0.5288\n",

"Epoch 3/3\n",

"168/168 [==============================] - 36s 216ms/step - loss: 0.8406 - accuracy:

0.6905 - val\_loss: 97.8494 - val\_accuracy: 0.8173\n"

]

},

{

"data": {

"text/plain": [

"<keras.callbacks.History at 0x1e34c9b7310>"

]

},

"execution\_count": 10,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"model.fit(x\_train,steps\_per\_epoch=168,epochs=3,validation\_data=x\_test,validation\_steps=52)"

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"id": "38a2791b",

"metadata": {},

"outputs": [],

"source": [

"model.save(r'C:\\Users\\uma25\\project\\flask\\uploads\\fruit.h5')"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"id": "e8fcccb8",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Model: \"sequential\"\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

" Layer (type) Output Shape Param # \n",

"=================================================================\n",

" conv2d (Conv2D) (None, 126, 126, 32) 896 \n",

" \n",

" max\_pooling2d (MaxPooling2D (None, 63, 63, 32) 0 \n",

" ) \n",

" \n",

" flatten (Flatten) (None, 127008) 0 \n",

" \n",

" dense (Dense) (None, 40) 5080360 \n",

" \n",

" dense\_1 (Dense) (None, 70) 2870 \n",

" \n",

" dense\_2 (Dense) (None, 6) 426 \n",

" \n", "=================================================================\n",

"Total params: 5,084,552\n",

"Trainable params: 5,084,552\n",

"Non-trainable params: 0\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"

]

}

],

"source": [

"model.summary()"

]

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3 (ipykernel)",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.9.12"

}

},

"nbformat": 4,

"nbformat\_minor": 5

}