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\\Datas
et 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,valid
ation_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
}