```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 1,
   "id": "505cf664",
   "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_flip=True)\n",
    "test_datagen=ImageDataGenerator(rescale=1)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "id": "4cca39b6",
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Found 11386 images belonging to 9 classes.\n",
      "Found 3416 images belonging to 9 classes.\n"
     ]
    }
   ],
   "source": [
```

"x_train=train_datagen.flow_from_directory(r'C:\\Users\\uma25\\project\\Dataset Plant Disease\\Veg-dataset\\Veg-dataset\\train_set',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\\Veg-dataset\\Veg-dataset\\test_set',target_size=(128,128),batch_size=2,class_mode='categorical')"

]

},

{

"cell_type": "code",

"execution_count": 3,

"id": "15120f24",

"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": "ec7b5878",

"metadata": {},

"outputs": [],

"source": [

"model=Sequential()"

]

},

```
{
 "cell_type": "code",
 "execution_count": 5,
 "id": "fe21b4e6",
 "metadata": {},
 "outputs": [],
 "source": [
  "model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))"
 ]
},
{
 "cell_type": "code",
 "execution_count": 6,
 "id": "f9721cca",
 "metadata": {},
 "outputs": [],
 "source": [
  "model.add(MaxPooling2D(pool_size=(2,2)))"
 ]
},
{
 "cell_type": "code",
 "execution_count": 7,
 "id": "54769702",
 "metadata": {},
 "outputs": [],
 "source": [
  "model.add(Flatten())"
 ]
},
{
```

```
 "cell_type": "code",
 "execution_count": 8,
 "id": "00b3ab14",
 "metadata": {},
 "outputs": [],
 "source": [
  "model.add(Dense(units=300,kernel_initializer='uniform',activation='relu'))"
 ]
},
{
 "cell_type": "code",
 "execution_count": 9,
 "id": "790bddf6",
 "metadata": {},
 "outputs": [],
 "source": [
  "model.add(Dense(units=150,kernel_initializer='uniform',activation='relu'))"
 ]
},
{
 "cell_type": "code",
 "execution_count": 10,
 "id": "d3f34fe4",
 "metadata": {},
 "outputs": [],
 "source": [
  "model.add(Dense(units=75,kernel_initializer='uniform',activation='relu'))"
 ]
},
{
 "cell_type": "code",
```

```
  "execution_count": 11,
  "id": "fedefb79",
  "metadata": {},
  "outputs": [],
  "source": [
   "model.add(Dense(units=9,kernel_initializer='uniform',activation='softmax'))"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 12,
  "id": "80d841b8",
  "metadata": {},
  "outputs": [],
  "source": [
   "model.compile(loss='categorical_crossentropy',optimizer=\"adam\",metrics=[\"accuracy\"])"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 13,
  "id": "35d47cf9",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "Epoch 1/20\n",
     "89/89 [==============================] - 95s 1s/step - loss: 2.1765 - accuracy: 0.1404 - val_loss: 107.0669 - val_accuracy: 0.2407\n",
```

"Epoch 2/20\n",

"89/89 [==============================] - 61s 679ms/step - loss: 2.1010 - accuracy: 0.2303 - val_loss: 73.7251 - val_accuracy: 0.0741\n",

"Epoch 3/20\n",

"89/89 [==============================] - 67s 755ms/step - loss: 2.1514 - accuracy: 0.1348 - val_loss: 56.0996 - val_accuracy: 0.1111\n",

"Epoch 4/20\n",

"89/89 [==============================] - 64s 717ms/step - loss: 2.0868 - accuracy: 0.1573 - val_loss: 23.7097 - val_accuracy: 0.3148\n",

"Epoch 5/20\n",

"89/89 [==============================] - 60s 671ms/step - loss: 2.0239 - accuracy: 0.3090 - val_loss: 99.1493 - val_accuracy: 0.2222\n",

"Epoch 6/20\n",

"89/89 [==============================] - 72s 807ms/step - loss: 1.9236 - accuracy: 0.2753 - val_loss: 172.7210 - val_accuracy: 0.1296\n",

"Epoch 7/20\n",

"89/89 [==============================] - 86s 971ms/step - loss: 1.9143 - accuracy: 0.2753 - val_loss: 107.2718 - val_accuracy: 0.2778\n",

"Epoch 8/20\n",

"89/89 [==============================] - 58s 646ms/step - loss: 1.7796 - accuracy: 0.3034 - val_loss: 64.2221 - val_accuracy: 0.3148\n",

"Epoch 9/20\n",

"89/89 [==============================] - 60s 676ms/step - loss: 1.7756 - accuracy: 0.3427 - val_loss: 182.9076 - val_accuracy: 0.3519\n",

"Epoch 10/20\n",

"89/89 [==============================] - 59s 665ms/step - loss: 1.8444 - accuracy: 0.2978 - val_loss: 138.7072 - val_accuracy: 0.2407\n",

"Epoch 11/20\n",

"89/89 [==============================] - 53s 598ms/step - loss: 1.7811 - accuracy: 0.2640 - val_loss: 111.3470 - val_accuracy: 0.3333\n",

"Epoch 12/20\n",

"89/89 [==============================] - 57s 637ms/step - loss: 1.8700 - accuracy: 0.2809 - val_loss: 104.8549 - val_accuracy: 0.2778\n",

"Epoch 13/20\n",

"89/89 [==============================] - 53s 593ms/step - loss: 1.8179 - accuracy: 0.3371 - val_loss: 88.9790 - val_accuracy: 0.3519\n",

"Epoch 14/20\n",

"89/89 [==============================] - 53s 590ms/step - loss: 1.7108 - accuracy: 0.2921 - val_loss: 79.7810 - val_accuracy: 0.4074\n",

"Epoch 15/20\n",

"89/89 [==============================] - 57s 644ms/step - loss: 1.8212 - accuracy: 0.2416 - val_loss: 187.6725 - val_accuracy: 0.2222\n",

"Epoch 16/20\n",

"89/89 [==============================] - 53s 589ms/step - loss: 1.7251 - accuracy: 0.3483 - val_loss: 148.7835 - val_accuracy: 0.4259\n",

"Epoch 17/20\n",

"89/89 [==============================] - 56s 624ms/step - loss: 1.6795 - accuracy: 0.3146 - val_loss: 109.7393 - val_accuracy: 0.2593\n",

"Epoch 18/20\n",

"89/89 [==============================] - 54s 608ms/step - loss: 1.8275 - accuracy: 0.3202 - val_loss: 90.4495 - val_accuracy: 0.3148\n",

"Epoch 19/20\n",

"89/89 [==============================] - 53s 591ms/step - loss: 1.7737 - accuracy: 0.3427 - val_loss: 141.6376 - val_accuracy: 0.2593\n",

"Epoch 20/20\n",

"89/89 [==============================] - 53s 592ms/step - loss: 1.7060 - accuracy: 0.4213 - val_loss: 96.3340 - val_accuracy: 0.3889\n"

]

},

{

"data": {

"text/plain": [

"<keras.callbacks.History at 0x25aa4cdb7f0>"

]

},

"execution_count": 13,

"metadata": {},

"output_type": "execute_result"

```
 }
 ],
 "source": [
  "model.fit(x_train,steps_per_epoch=89,epochs=20,validation_data=x_test,validation_steps=27)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 14,
 "id": "2cb00ef6",
 "metadata": {},
 "outputs": [],
 "source": [
  "model.save(r'C:\\Users\\uma25\\project\\flask\\uploads\\vegetable.h5')          "
 ]
},
{
 "cell_type": "code",
 "execution_count": 15,
 "id": "abce82ea",
 "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, 300)             38102700  \n",
    "                                                \n",
    " dense_1 (Dense)            (None, 150)             45150     \n",
    "                                                \n",
    " dense_2 (Dense)            (None, 75)              11325     \n",
    "                                                \n",
    " dense_3 (Dense)            (None, 9)               684       \n",
    "                                                \n",
    "=================================================================\n",
    "Total params: 38,160,755\n",
    "Trainable params: 38,160,755\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
}
```