```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 1,
   "metadata": {},
   "outputs": [],
   "source": [
    "import keras\n",
    "import tensorflow\n",
    "\n",
    "from tensorflow.keras.preprocessing.image import ImageDataGenerator"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {},
   "outputs": [],
   "source": [
    "train_datagen = ImageDataGenerator(rescale=1./255,\n",
    "                                   shear_range=0.2,\n",
    "                                   rotation_range=180,\n",
    "                                   zoom_range=0.2,\n",
    "                                   horizontal_flip=True)\n",
    "\n",
    "test_datagen = ImageDataGenerator(rescale=1./255)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Found 436 images belonging to 2 classes.\n"
     ]
    }
   ],
   "source": [
    "x_train = train_datagen.flow_from_directory(r'./Dataset/train_set/',\n",
    "                                            target_size=(128, 128),\n",
    "                                            batch_size=32,\n",
```

```
    "                                                  class_mode='binary')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 4,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Found 121 images belonging to 2 classes.\n"
     ]
    }
   ],
   "source": [
    "x_test = train_datagen.flow_from_directory(r'./Dataset/test_set/',\n",
    "                                           target_size=(128, 128),\n",
    "                                           batch_size=32,\n",
    "                                           class_mode='binary')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 5,
   "metadata": {},
   "outputs": [],
   "source": [
    "from tensorflow.keras.models import Sequential\n",
    "from tensorflow.keras.layers import Dense, Convolution2D, MaxPooling2D,
Flatten\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 6,
   "metadata": {},
   "outputs": [],
   "source": [
    "model = Sequential()\n",
    "model.add(Convolution2D(32, (3,3), input_shape=(128, 128, 3),
activation=\"relu\"))\n",
    "model.add(MaxPooling2D(pool_size=(2,2)))\n",
    "model.add(Flatten())\n",
    "model.add(Dense(150,activation=\"relu\"))\n",
```

```
    "model.add(Dense(1, activation=\"sigmoid\"))\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 7,
   "metadata": {},
   "outputs": [],
   "source": [
    "model.compile(loss=\"binary_crossentropy\",\n",
    "              optimizer=\"adam\",\n",
    "              metrics=[\"accuracy\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 8,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Epoch 1/10\n",
      "14/14 [==============================] - 64s 4s/step - loss: 3.5440 -
accuracy: 0.5665 - val_loss: 0.4052 - val_accuracy: 0.8430\n",
      "Epoch 2/10\n",
      "14/14 [==============================] - 23s 2s/step - loss: 0.5222 -
accuracy: 0.7431 - val_loss: 0.2283 - val_accuracy: 0.9669\n",
      "Epoch 3/10\n",
      "14/14 [==============================] - 23s 2s/step - loss: 0.3097 -
accuracy: 0.8647 - val_loss: 0.1622 - val_accuracy: 0.9504\n",
      "Epoch 4/10\n",
      "14/14 [==============================] - 22s 2s/step - loss: 0.2392 -
accuracy: 0.8945 - val_loss: 0.1137 - val_accuracy: 0.9669\n",
      "Epoch 5/10\n",
      "14/14 [==============================] - 23s 2s/step - loss: 0.2125 -
accuracy: 0.8968 - val_loss: 0.1337 - val_accuracy: 0.9504\n",
      "Epoch 6/10\n",
      "14/14 [==============================] - 23s 2s/step - loss: 0.1922 -
accuracy: 0.9243 - val_loss: 0.0887 - val_accuracy: 0.9669\n",
      "Epoch 7/10\n",
      "14/14 [==============================] - 23s 2s/step - loss: 0.1773 -
accuracy: 0.9266 - val_loss: 0.1454 - val_accuracy: 0.9339\n",
      "Epoch 8/10\n",
      "14/14 [==============================] - 21s 2s/step - loss: 0.1678 -
```

```
accuracy: 0.9427 - val_loss: 0.0835 - val_accuracy: 0.9752\n",
      "Epoch 9/10\n",
      "14/14 [==============================] - 24s 2s/step - loss: 0.1733 -
accuracy: 0.9243 - val_loss: 0.1079 - val_accuracy: 0.9669\n",
      "Epoch 10/10\n",
      "14/14 [==============================] - 25s 2s/step - loss: 0.1647 -
accuracy: 0.9335 - val_loss: 0.0716 - val_accuracy: 0.9752\n"
     ]
    },
    {
     "data": {
      "text/plain": [
       "<keras.callbacks.History at 0x1920c974be0>"
      ]
     },
     "execution_count": 8,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "model.fit(x_train, steps_per_epoch=14, epochs=10,
validation_data=x_test, validation_steps=4)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "Save the model"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 9,
   "metadata": {},
   "outputs": [],
   "source": [
    "model.save(\"model.h5\")"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "Prediction"
```

```
 ]
},
{
 "cell_type": "code",
 "execution_count": 10,
 "metadata": {},
 "outputs": [],
 "source": [
  "from tensorflow.keras.models import load_model\n",
  "from tensorflow.keras.preprocessing import image\n",
  "import numpy as np\n",
  "import cv2"
 ]
},
{
 "cell_type": "code",
 "execution_count": 11,
 "metadata": {},
 "outputs": [],
 "source": [
  "model = load_model(\"model.h5\")"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {},
 "source": [
  "Reviewing the model "
 ]
},
{
 "cell_type": "code",
 "execution_count": 13,
 "metadata": {},
 "outputs": [],
 "source": [
  "img = image.load_img(\"forest-fire.jpg\")\n",
  "x = image.img_to_array(img)\n",
  "res = cv2.resize(x, dsize=(128, 128), interpolation=cv2.INTER_CUBIC)\n",
  "x = np.expand_dims(res, axis=0)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 14,
 "metadata": {},
```

```
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "1/1 [==============================] - 1s 524ms/step\n"
    ]
   },
   {
    "data": {
     "text/plain": [
      "0"
     ]
    },
    "execution_count": 14,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "pred = model.predict(x)\n",
   "pred = int(pred[0][0])\n",
   "pred"
  ]
 }
],
"metadata": {
 "kernelspec": {
  "display_name": "Python 3.10.6 64-bit",
  "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": "3.10.6"
 },
 "orig_nbformat": 4,
 "vscode": {
```

```
   "interpreter": {
    "hash":
"a9cff5a362bc38ef45d817ae74b1af54d6a076e3d773891282bce078b815ba34"
   }
  }
 },
 "nbformat": 4,
 "nbformat_minor": 2
}
```