```
!pip install kaggle
```

```
Requirement already satisfied: kaggle in
/usr/local/lib/python3.11/dist-packages (1.7.4.5)
Requirement already satisfied: bleach in
/usr/local/lib/python3.11/dist-packages (from kaggle) (6.2.0)
Requirement already satisfied: certifi>=14.05.14 in
/usr/local/lib/python3.11/dist-packages (from kaggle) (2025.4.26)
Requirement already satisfied: charset-normalizer in
/usr/local/lib/python3.11/dist-packages (from kaggle) (3.4.2)
Requirement already satisfied: idna in /usr/local/lib/python3.11/dist-
packages (from kaggle) (3.10)
Requirement already satisfied: protobuf in
/usr/local/lib/python3.11/dist-packages (from kaggle) (5.29.5)
Requirement already satisfied: python-dateutil>=2.5.3 in
/usr/local/lib/python3.11/dist-packages (from kaggle) (2.9.0.post0)
Requirement already satisfied: python-slugify in
/usr/local/lib/python3.11/dist-packages (from kaggle) (8.0.4)
Requirement already satisfied: requests in
/usr/local/lib/python3.11/dist-packages (from kaggle) (2.32.3)
Requirement already satisfied: setuptools>=21.0.0 in
/usr/local/lib/python3.11/dist-packages (from kaggle) (75.2.0)
Requirement already satisfied: six>=1.10 in
/usr/local/lib/python3.11/dist-packages (from kaggle) (1.17.0)
Requirement already satisfied: text-unidecode in
/usr/local/lib/python3.11/dist-packages (from kaggle) (1.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-
packages (from kaggle) (4.67.1)
Requirement already satisfied: urllib3>=1.15.1 in
/usr/local/lib/python3.11/dist-packages (from kaggle) (2.4.0)
Requirement already satisfied: webencodings in
/usr/local/lib/python3.11/dist-packages (from kaggle) (0.5.1)
```

```
from google.colab import files
files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving kaggle.json to kaggle.json
```

```
{'kaggle.json':
b'{"username":"samaviot7","key":"b6582ac5631ecea828f527a451b4a1c7"}'}
```

```
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
```

```
!kaggle datasets download -d phylake1337/fire-dataset -p
/content/fire-dataset --unzip
```

```
Dataset URL: https://www.kaggle.com/datasets/phylake1337/fire-dataset
License(s): CC0-1.0
Downloading fire-dataset.zip to /content/fire-dataset
 99% 383M/387M [00:01<00:00, 317MB/s]
100% 387M/387M [00:01<00:00, 389MB/s]
```

```
!unzip students-performance-in-exams.zip
```

```
unzip:  cannot find or open students-performance-in-exams.zip,
students-performance-in-exams.zip.zip or students-performance-in-
exams.zip.ZIP.
```

```
!ls
```

```
fire-dataset  kaggle.json  sample_data
```

```
!ls Fire-Detection-Image-Dataset
```

```
ls: cannot access 'Fire-Detection-Image-Dataset': No such file or
directory
```

```
!ls /content/fire-dataset/Fire_images
```

```
ls: cannot access '/content/fire-dataset/Fire_images': No such file or
directory
```

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator

data_dir = '/content/Fire-Detection-Image-Dataset'

 IMG_SIZE=224
 BATCH_SIZE=32

train_datagen=ImageDataGenerator(rescale=1./255,validation_split=0.2)

train_generator=train_datagen.flow_from_directory(
    '/content/fire-dataset',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='training'
 )
```

```
Found 800 images belonging to 1 classes.
```

```
 val_generator=train_datagen.flow_from_directory(
    '/content/fire-dataset',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='validation'
 )
```

```
Found 199 images belonging to 1 classes.
```

```
 model=keras.Sequential([
    layers.Conv2D(32,(3,3),activation='relu',
input_shape=(IMG_SIZE,IMG_SIZE,3)),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,(3,3),activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(128,(3,3),activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Flatten(),
    layers.Dense(128,activation='relu'),
    layers.Dense(1,activation='sigmoid')
 ])
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/
convolutional/base_conv.py:107: UserWarning: Do not pass an
`input_shape`/`input_dim` argument to a layer. When using Sequential
models, prefer using an `Input(shape)` object as the first layer in
the model instead.
  super().__init__(activity_regularizer=activity_regularizer,
**kwargs)
```

```
model.summary()
```

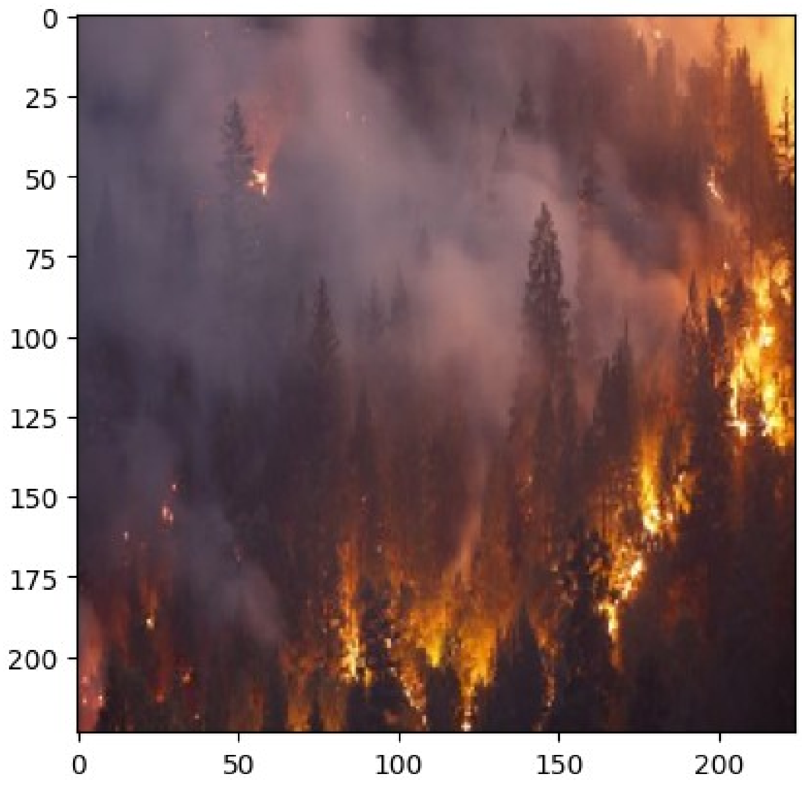
```
Model: "sequential"
```

```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━
━━━━━┓
┃ Layer (type)                    ┃ Output Shape           ┃
Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━
━━━━━┩
│ conv2d (Conv2D)                 │ (None, 222, 222, 32)   │
896 │
├─────────────────────────────────┼────────────────────────┼──────────
─────┤
│ max_pooling2d (MaxPooling2D)    │ (None, 111, 111, 32)   │
0 │
├─────────────────────────────────┼────────────────────────┼──────────
─────┤
│ conv2d_1 (Conv2D)               │ (None, 109, 109, 64)   │
18,496 │
├─────────────────────────────────┼────────────────────────┼──────────
─────┤
│ max_pooling2d_1 (MaxPooling2D)  │ (None, 54, 54, 64)     │
0 │
├─────────────────────────────────┼────────────────────────┼──────────
```

```
──────┤
│ conv2d_2 (Conv2D)                 │ (None, 52, 52, 128)      │
73,856 │
├───────────────────────────────────┼──────────────────────────┼──────────
──────┤
│ max_pooling2d_2 (MaxPooling2D)    │ (None, 26, 26, 128)      │
0 │
├───────────────────────────────────┼──────────────────────────┼──────────
──────┤
│ flatten (Flatten)                 │ (None, 86528)            │
0 │
├───────────────────────────────────┼──────────────────────────┼──────────
──────┤
│ dense (Dense)                     │ (None, 128)              │
11,075,712 │
├───────────────────────────────────┼──────────────────────────┼──────────
──────┤
│ dense_1 (Dense)                   │ (None, 1)                │
129 │
└───────────────────────────────────┴──────────────────────────┴──────────
──────┘

 Total params: 11,169,089 (42.61 MB)

 Trainable params: 11,169,089 (42.61 MB)

 Non-trainable params: 0 (0.00 B)
```

```
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['ac
curacy'])


model.fit(train_generator,epochs=3,validation_data=val_generator,batch
_size=BATCH_SIZE)
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/
data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset`
class should call `super().__init__(**kwargs)` in its constructor.
`**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will
be ignored.
  self._warn_if_super_not_called()

Epoch 1/3
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5s/step - accuracy: 0.9952 - loss:
0.1018

/usr/local/lib/python3.11/dist-packages/keras/src/trainers/
data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset`
class should call `super().__init__(**kwargs)` in its constructor.
`**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will
be ignored.
  self._warn_if_super_not_called()

 25/25 ━━━━━━━━━━━━━━━━━━━━ 131s 5s/step - accuracy: 0.9954 - loss:
0.0989 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 2/3
25/25 ━━━━━━━━━━━━━━━━━━━━ 116s 5s/step - accuracy: 1.0000 - loss:
0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 3/3
25/25 ━━━━━━━━━━━━━━━━━━━━ 117s 5s/step - accuracy: 1.0000 - loss:
0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00

<keras.src.callbacks.history.History at 0x7fc810eab8d0>

 model.save('/content/fire-datasetmodel.h5')

WARNING:absl:You are saving your model as an HDF5 file via
`model.save()` or `keras.saving.save_model(model)`. This file format
is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')` or
`keras.saving.save_model(model, 'my_model.keras')`.

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
import numpy as np
model=load_model('/content/fire-datasetmodel.h5')
print('Model Loaded Sucessfully')

WARNING:absl:Compiled the loaded model, but the compiled metrics have
yet to be built. `model.compile_metrics` will be empty until you train
or evaluate the model.

Model Loaded Sucessfully

test_image_path="/content/fire-dataset/fire_dataset/fire_images/
fire.102.png"
img=image.load_img(test_image_path,target_size=(224,224))
plt.imshow(img)
plt.axis()
plt.show()
```



```
img_array=image.img_to_array(img)
img_array=np.expand_dims(img_array,axis=0)
img_array/=255.0
prediction=model.predict(img_array)
print(prediction)
if prediction>=0.5:
 print("no fire Detected")
else:
 print("fire Detected")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step
[[0.]]
fire Detected
```