```
! # ls Datasets/scrapped/All
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
!ls "/content/drive/My Drive/Training Dataset"
```

```
Fire  NoFire
```

```
import os
import cv2
import numpy as np
from tqdm import tqdm

DATADIR = '/content/drive/My Drive/Training Dataset'
CATEGORIES = ['Fire', 'NoFire']

IMG_SIZE = 64
def create_training_data():
    training_data = []
    for category in CATEGORIES:

        path = os.path.join(DATADIR,category)
        class_num = CATEGORIES.index(category)  # get the
classification  (0 or a 1). 0=C 1=O

        for img in tqdm(os.listdir(path)):  # iterate over each image
            try:
                img_array = cv2.imread(os.path.join(path,img))  #
convert to array
                new_array = cv2.resize(img_array, (IMG_SIZE,
IMG_SIZE))  # resize to normalize data size
                training_data.append([new_array, class_num])  # add
this to our training_data
            except Exception as e:  # in the interest in keeping the
output clean...
                pass

    return training_data

training_data = create_training_data()
```

```
100%|██████████| 1124/1124 [00:45<00:00, 24.72it/s]
100%|██████████| 1301/1301 [00:51<00:00, 25.49it/s]
```

```
import random

print(len(training_data))
random.shuffle(training_data)
```

```
for sample in training_data[:10]:
    print(sample[1])

2423
1
0
0
1
1
1
1
1
1
1

X = []
Y = []

for features,label in training_data:
    X.append(features)
    Y.append(label)

X = np.array(X).reshape(-1, IMG_SIZE, IMG_SIZE, 3)
X = X/255.0
X.shape[1:]

Y = np.array(Y)

# # set up image augmentation
# from keras.preprocessing.image import ImageDataGenerator

# datagen = ImageDataGenerator(
#     rotation_range=15,
#     horizontal_flip=True,
#     width_shift_range=0.1,
#     height_shift_range=0.1
#     #zoom_range=0.3
#     )
# datagen.fit(X)

import tensorflow as tf
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation,
Flatten
from tensorflow.keras.layers import Conv2D, AveragePooling2D

model = Sequential()
```

```
model.add(Conv2D(filters=16, kernel_size=(3, 3), activation='relu',
input_shape=X.shape[1:]))
model.add(AveragePooling2D())
model.add(Dropout(0.5))

model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='relu'))
model.add(AveragePooling2D())
model.add(Dropout(0.5))

model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(AveragePooling2D())
model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(units=256, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(units=128, activation='relu'))

model.add(Dense(units=2, activation = 'softmax'))

model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])



history = model.fit(X, Y, batch_size=32,
epochs=100,validation_split=0.3)
# model.fit_generator(datagen.flow(X, Y, batch_size=32),
#                     epochs=100,
#                     verbose=1)

Epoch 1/100
53/53 [==============================] - 5s 72ms/step - loss: 0.5977 -
accuracy: 0.6745 - val_loss: 0.5569 - val_accuracy: 0.7276
Epoch 2/100
53/53 [==============================] - 5s 93ms/step - loss: 0.5263 -
accuracy: 0.7500 - val_loss: 0.4774 - val_accuracy: 0.7730
Epoch 3/100
53/53 [==============================] - 4s 68ms/step - loss: 0.4824 -
accuracy: 0.7801 - val_loss: 0.5205 - val_accuracy: 0.7469
Epoch 4/100
53/53 [==============================] - 3s 64ms/step - loss: 0.4576 -
accuracy: 0.7930 - val_loss: 0.4921 - val_accuracy: 0.7620
Epoch 5/100
53/53 [==============================] - 4s 78ms/step - loss: 0.4379 -
accuracy: 0.8113 - val_loss: 0.4527 - val_accuracy: 0.7758
Epoch 6/100
53/53 [==============================] - 4s 78ms/step - loss: 0.4114 -
```

```
accuracy: 0.8237 - val_loss: 0.4093 - val_accuracy: 0.8033
Epoch 7/100
53/53 [==============================] - 3s 63ms/step - loss: 0.4153 -
accuracy: 0.8184 - val_loss: 0.4842 - val_accuracy: 0.7524
Epoch 8/100
53/53 [==============================] - 3s 63ms/step - loss: 0.4092 -
accuracy: 0.8137 - val_loss: 0.3926 - val_accuracy: 0.8171
Epoch 9/100
53/53 [==============================] - 4s 84ms/step - loss: 0.3540 -
accuracy: 0.8432 - val_loss: 0.3961 - val_accuracy: 0.8061
Epoch 10/100
53/53 [==============================] - 4s 74ms/step - loss: 0.3396 -
accuracy: 0.8520 - val_loss: 0.3592 - val_accuracy: 0.8514
Epoch 11/100
53/53 [==============================] - 5s 93ms/step - loss: 0.3348 -
accuracy: 0.8567 - val_loss: 0.3303 - val_accuracy: 0.8514
Epoch 12/100
53/53 [==============================] - 4s 70ms/step - loss: 0.3097 -
accuracy: 0.8738 - val_loss: 0.3000 - val_accuracy: 0.8721
Epoch 13/100
53/53 [==============================] - 5s 88ms/step - loss: 0.3352 -
accuracy: 0.8550 - val_loss: 0.3006 - val_accuracy: 0.8652
Epoch 14/100
53/53 [==============================] - 3s 64ms/step - loss: 0.2897 -
accuracy: 0.8791 - val_loss: 0.2990 - val_accuracy: 0.8638
Epoch 15/100
53/53 [==============================] - 3s 65ms/step - loss: 0.2613 -
accuracy: 0.8945 - val_loss: 0.2633 - val_accuracy: 0.8996
Epoch 16/100
53/53 [==============================] - 4s 76ms/step - loss: 0.2329 -
accuracy: 0.9027 - val_loss: 0.3124 - val_accuracy: 0.8721
Epoch 17/100
53/53 [==============================] - 4s 82ms/step - loss: 0.2467 -
accuracy: 0.8998 - val_loss: 0.2643 - val_accuracy: 0.8858
Epoch 18/100
53/53 [==============================] - 3s 64ms/step - loss: 0.2489 -
accuracy: 0.8974 - val_loss: 0.2785 - val_accuracy: 0.8872
Epoch 19/100
53/53 [==============================] - 4s 67ms/step - loss: 0.2070 -
accuracy: 0.9163 - val_loss: 0.2619 - val_accuracy: 0.8913
Epoch 20/100
53/53 [==============================] - 6s 116ms/step - loss: 0.2066
- accuracy: 0.9092 - val_loss: 0.2212 - val_accuracy: 0.9037
Epoch 21/100
53/53 [==============================] - 4s 72ms/step - loss: 0.2089 -
accuracy: 0.9228 - val_loss: 0.2172 - val_accuracy: 0.9216
Epoch 22/100
53/53 [==============================] - 3s 63ms/step - loss: 0.1908 -
accuracy: 0.9328 - val_loss: 0.2363 - val_accuracy: 0.9120
```

```
Epoch 23/100
53/53 [==============================] - 3s 64ms/step - loss: 0.1678 -
accuracy: 0.9334 - val_loss: 0.2162 - val_accuracy: 0.9216
Epoch 24/100
53/53 [==============================] - 5s 89ms/step - loss: 0.1707 -
accuracy: 0.9292 - val_loss: 0.2106 - val_accuracy: 0.9216
Epoch 25/100
53/53 [==============================] - 4s 68ms/step - loss: 0.1529 -
accuracy: 0.9387 - val_loss: 0.2440 - val_accuracy: 0.9051
Epoch 26/100
53/53 [==============================] - 3s 64ms/step - loss: 0.1484 -
accuracy: 0.9399 - val_loss: 0.2279 - val_accuracy: 0.9188
Epoch 27/100
53/53 [==============================] - 3s 64ms/step - loss: 0.1449 -
accuracy: 0.9440 - val_loss: 0.2385 - val_accuracy: 0.9188
Epoch 28/100
53/53 [==============================] - 5s 94ms/step - loss: 0.1476 -
accuracy: 0.9428 - val_loss: 0.2255 - val_accuracy: 0.9367
Epoch 29/100
53/53 [==============================] - 3s 64ms/step - loss: 0.1738 -
accuracy: 0.9322 - val_loss: 0.2552 - val_accuracy: 0.9051
Epoch 30/100
53/53 [==============================] - 3s 65ms/step - loss: 0.1314 -
accuracy: 0.9475 - val_loss: 0.2171 - val_accuracy: 0.9285
Epoch 31/100
53/53 [==============================] - 4s 67ms/step - loss: 0.1346 -
accuracy: 0.9499 - val_loss: 0.2159 - val_accuracy: 0.9257
Epoch 32/100
53/53 [==============================] - 5s 91ms/step - loss: 0.0973 -
accuracy: 0.9640 - val_loss: 0.2510 - val_accuracy: 0.9257
Epoch 33/100
53/53 [==============================] - 3s 65ms/step - loss: 0.1059 -
accuracy: 0.9587 - val_loss: 0.2576 - val_accuracy: 0.9161
Epoch 34/100
53/53 [==============================] - 3s 65ms/step - loss: 0.1049 -
accuracy: 0.9581 - val_loss: 0.2632 - val_accuracy: 0.9298
Epoch 35/100
53/53 [==============================] - 4s 74ms/step - loss: 0.1023 -
accuracy: 0.9676 - val_loss: 0.2452 - val_accuracy: 0.9175
Epoch 36/100
53/53 [==============================] - 5s 85ms/step - loss: 0.0939 -
accuracy: 0.9682 - val_loss: 0.2433 - val_accuracy: 0.9312
Epoch 37/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0975 -
accuracy: 0.9676 - val_loss: 0.2400 - val_accuracy: 0.9381
Epoch 38/100
53/53 [==============================] - 3s 64ms/step - loss: 0.1051 -
accuracy: 0.9599 - val_loss: 0.2557 - val_accuracy: 0.9092
Epoch 39/100
```

```
53/53 [==============================] - 4s 81ms/step - loss: 0.0906 -
accuracy: 0.9658 - val_loss: 0.2123 - val_accuracy: 0.9422
Epoch 40/100
53/53 [==============================] - 4s 79ms/step - loss: 0.0588 -
accuracy: 0.9788 - val_loss: 0.2252 - val_accuracy: 0.9367
Epoch 41/100
53/53 [==============================] - 3s 64ms/step - loss: 0.0677 -
accuracy: 0.9717 - val_loss: 0.2851 - val_accuracy: 0.9188
Epoch 42/100
53/53 [==============================] - 3s 64ms/step - loss: 0.0710 -
accuracy: 0.9705 - val_loss: 0.2303 - val_accuracy: 0.9312
Epoch 43/100
53/53 [==============================] - 5s 88ms/step - loss: 0.0832 -
accuracy: 0.9670 - val_loss: 0.2414 - val_accuracy: 0.9381
Epoch 44/100
53/53 [==============================] - 4s 71ms/step - loss: 0.0669 -
accuracy: 0.9735 - val_loss: 0.2414 - val_accuracy: 0.9271
Epoch 45/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0829 -
accuracy: 0.9723 - val_loss: 0.2193 - val_accuracy: 0.9257
Epoch 46/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0569 -
accuracy: 0.9794 - val_loss: 0.2870 - val_accuracy: 0.9285
Epoch 47/100
53/53 [==============================] - 5s 93ms/step - loss: 0.0446 -
accuracy: 0.9841 - val_loss: 0.2724 - val_accuracy: 0.9312
Epoch 48/100
53/53 [==============================] - 4s 70ms/step - loss: 0.0856 -
accuracy: 0.9699 - val_loss: 0.2204 - val_accuracy: 0.9367
Epoch 49/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0532 -
accuracy: 0.9794 - val_loss: 0.2331 - val_accuracy: 0.9354
Epoch 50/100
53/53 [==============================] - 4s 67ms/step - loss: 0.0704 -
accuracy: 0.9758 - val_loss: 0.2270 - val_accuracy: 0.9354
Epoch 51/100
53/53 [==============================] - 5s 91ms/step - loss: 0.0474 -
accuracy: 0.9841 - val_loss: 0.2621 - val_accuracy: 0.9326
Epoch 52/100
53/53 [==============================] - 3s 64ms/step - loss: 0.0422 -
accuracy: 0.9858 - val_loss: 0.3039 - val_accuracy: 0.9340
Epoch 53/100
53/53 [==============================] - 3s 64ms/step - loss: 0.0550 -
accuracy: 0.9811 - val_loss: 0.2597 - val_accuracy: 0.9381
Epoch 54/100
53/53 [==============================] - 4s 75ms/step - loss: 0.0476 -
accuracy: 0.9811 - val_loss: 0.2627 - val_accuracy: 0.9422
Epoch 55/100
53/53 [==============================] - 5s 85ms/step - loss: 0.0692 -
```

```
accuracy: 0.9741 - val_loss: 0.2836 - val_accuracy: 0.9312
Epoch 56/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0557 -
accuracy: 0.9829 - val_loss: 0.2414 - val_accuracy: 0.9395
Epoch 57/100
53/53 [==============================] - 3s 64ms/step - loss: 0.0573 -
accuracy: 0.9764 - val_loss: 0.2526 - val_accuracy: 0.9354
Epoch 58/100
53/53 [==============================] - 4s 82ms/step - loss: 0.0483 -
accuracy: 0.9823 - val_loss: 0.2414 - val_accuracy: 0.9340
Epoch 59/100
53/53 [==============================] - 4s 77ms/step - loss: 0.0678 -
accuracy: 0.9764 - val_loss: 0.2476 - val_accuracy: 0.9395
Epoch 60/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0470 -
accuracy: 0.9841 - val_loss: 0.2638 - val_accuracy: 0.9312
Epoch 61/100
53/53 [==============================] - 3s 64ms/step - loss: 0.0309 -
accuracy: 0.9876 - val_loss: 0.2621 - val_accuracy: 0.9464
Epoch 62/100
53/53 [==============================] - 5s 88ms/step - loss: 0.0264 -
accuracy: 0.9894 - val_loss: 0.2823 - val_accuracy: 0.9381
Epoch 63/100
53/53 [==============================] - 4s 72ms/step - loss: 0.0956 -
accuracy: 0.9711 - val_loss: 0.3099 - val_accuracy: 0.9092
Epoch 64/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0481 -
accuracy: 0.9782 - val_loss: 0.2430 - val_accuracy: 0.9409
Epoch 65/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0335 -
accuracy: 0.9888 - val_loss: 0.3146 - val_accuracy: 0.9271
Epoch 66/100
53/53 [==============================] - 6s 109ms/step - loss: 0.0289
- accuracy: 0.9894 - val_loss: 0.2818 - val_accuracy: 0.9298
Epoch 67/100
53/53 [==============================] - 5s 86ms/step - loss: 0.0588 -
accuracy: 0.9776 - val_loss: 0.2336 - val_accuracy: 0.9395
Epoch 68/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0479 -
accuracy: 0.9811 - val_loss: 0.2474 - val_accuracy: 0.9326
Epoch 69/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0400 -
accuracy: 0.9870 - val_loss: 0.2842 - val_accuracy: 0.9340
Epoch 70/100
53/53 [==============================] - 4s 83ms/step - loss: 0.0268 -
accuracy: 0.9935 - val_loss: 0.2884 - val_accuracy: 0.9243
Epoch 71/100
53/53 [==============================] - 4s 77ms/step - loss: 0.0318 -
accuracy: 0.9900 - val_loss: 0.2835 - val_accuracy: 0.9409
```

```
Epoch 72/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0242 -
accuracy: 0.9929 - val_loss: 0.2844 - val_accuracy: 0.9436
Epoch 73/100
53/53 [==============================] - 3s 64ms/step - loss: 0.0266 -
accuracy: 0.9917 - val_loss: 0.3423 - val_accuracy: 0.9312
Epoch 74/100
53/53 [==============================] - 5s 105ms/step - loss: 0.0339
- accuracy: 0.9917 - val_loss: 0.2765 - val_accuracy: 0.9257
Epoch 75/100
53/53 [==============================] - 5s 90ms/step - loss: 0.0546 -
accuracy: 0.9805 - val_loss: 0.2673 - val_accuracy: 0.9326
Epoch 76/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0364 -
accuracy: 0.9876 - val_loss: 0.2602 - val_accuracy: 0.9243
Epoch 77/100
53/53 [==============================] - 3s 64ms/step - loss: 0.0439 -
accuracy: 0.9858 - val_loss: 0.2750 - val_accuracy: 0.9436
Epoch 78/100
53/53 [==============================] - 5s 96ms/step - loss: 0.0412 -
accuracy: 0.9847 - val_loss: 0.2459 - val_accuracy: 0.9312
Epoch 79/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0291 -
accuracy: 0.9882 - val_loss: 0.2938 - val_accuracy: 0.9312
Epoch 80/100
53/53 [==============================] - 3s 66ms/step - loss: 0.0332 -
accuracy: 0.9888 - val_loss: 0.3093 - val_accuracy: 0.9340
Epoch 81/100
53/53 [==============================] - 4s 70ms/step - loss: 0.0270 -
accuracy: 0.9894 - val_loss: 0.3359 - val_accuracy: 0.9216
Epoch 82/100
53/53 [==============================] - 5s 95ms/step - loss: 0.0336 -
accuracy: 0.9870 - val_loss: 0.2877 - val_accuracy: 0.9381
Epoch 83/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0384 -
accuracy: 0.9864 - val_loss: 0.2839 - val_accuracy: 0.9395
Epoch 84/100
53/53 [==============================] - 3s 64ms/step - loss: 0.0175 -
accuracy: 0.9941 - val_loss: 0.4221 - val_accuracy: 0.9354
Epoch 85/100
53/53 [==============================] - 4s 79ms/step - loss: 0.0378 -
accuracy: 0.9870 - val_loss: 0.2818 - val_accuracy: 0.9326
Epoch 86/100
53/53 [==============================] - 4s 81ms/step - loss: 0.0197 -
accuracy: 0.9935 - val_loss: 0.3306 - val_accuracy: 0.9395
Epoch 87/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0224 -
accuracy: 0.9953 - val_loss: 0.3130 - val_accuracy: 0.9381
Epoch 88/100
```

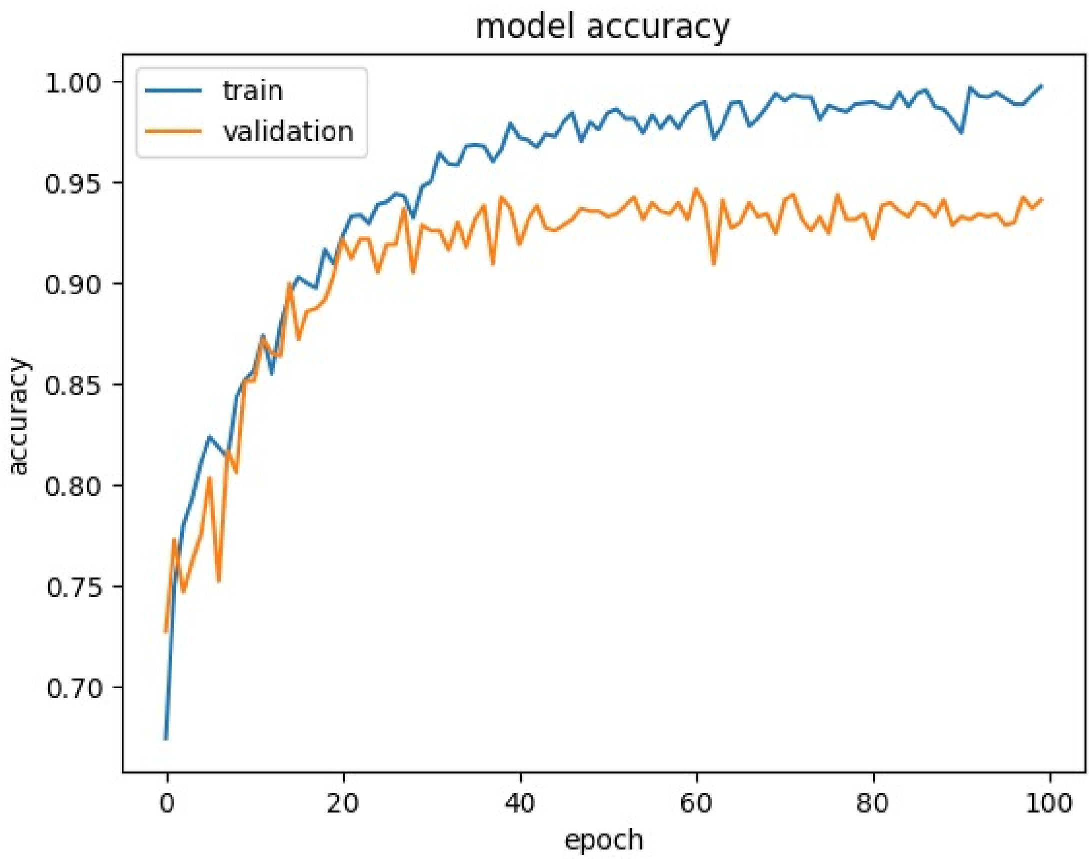
```
53/53 [==============================] - 3s 65ms/step - loss: 0.0425 - accuracy: 0.9870 - val_loss: 0.3054 - val_accuracy: 0.9326
Epoch 89/100
53/53 [==============================] - 5s 87ms/step - loss: 0.0404 - accuracy: 0.9858 - val_loss: 0.3173 - val_accuracy: 0.9409
Epoch 90/100
53/53 [==============================] - 4s 75ms/step - loss: 0.0520 - accuracy: 0.9805 - val_loss: 0.2885 - val_accuracy: 0.9285
Epoch 91/100
53/53 [==============================] - 3s 66ms/step - loss: 0.0676 - accuracy: 0.9741 - val_loss: 0.2638 - val_accuracy: 0.9326
Epoch 92/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0249 - accuracy: 0.9965 - val_loss: 0.3029 - val_accuracy: 0.9312
Epoch 93/100
53/53 [==============================] - 5s 95ms/step - loss: 0.0200 - accuracy: 0.9923 - val_loss: 0.2999 - val_accuracy: 0.9340
Epoch 94/100
53/53 [==============================] - 4s 67ms/step - loss: 0.0288 - accuracy: 0.9917 - val_loss: 0.3103 - val_accuracy: 0.9326
Epoch 95/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0179 - accuracy: 0.9941 - val_loss: 0.3299 - val_accuracy: 0.9340
Epoch 96/100
53/53 [==============================] - 4s 68ms/step - loss: 0.0306 - accuracy: 0.9912 - val_loss: 0.3725 - val_accuracy: 0.9285
Epoch 97/100
53/53 [==============================] - 5s 92ms/step - loss: 0.0460 - accuracy: 0.9882 - val_loss: 0.2543 - val_accuracy: 0.9298
Epoch 98/100
53/53 [==============================] - 3s 64ms/step - loss: 0.0293 - accuracy: 0.9882 - val_loss: 0.2769 - val_accuracy: 0.9422
Epoch 99/100
53/53 [==============================] - 3s 65ms/step - loss: 0.0212 - accuracy: 0.9929 - val_loss: 0.3085 - val_accuracy: 0.9367
Epoch 100/100
53/53 [==============================] - 5s 87ms/step - loss: 0.0183 - accuracy: 0.9971 - val_loss: 0.2737 - val_accuracy: 0.9409
```

```python
model.save('TrainedModels/Fire-64x64-color-v7.1-soft.h5')
```

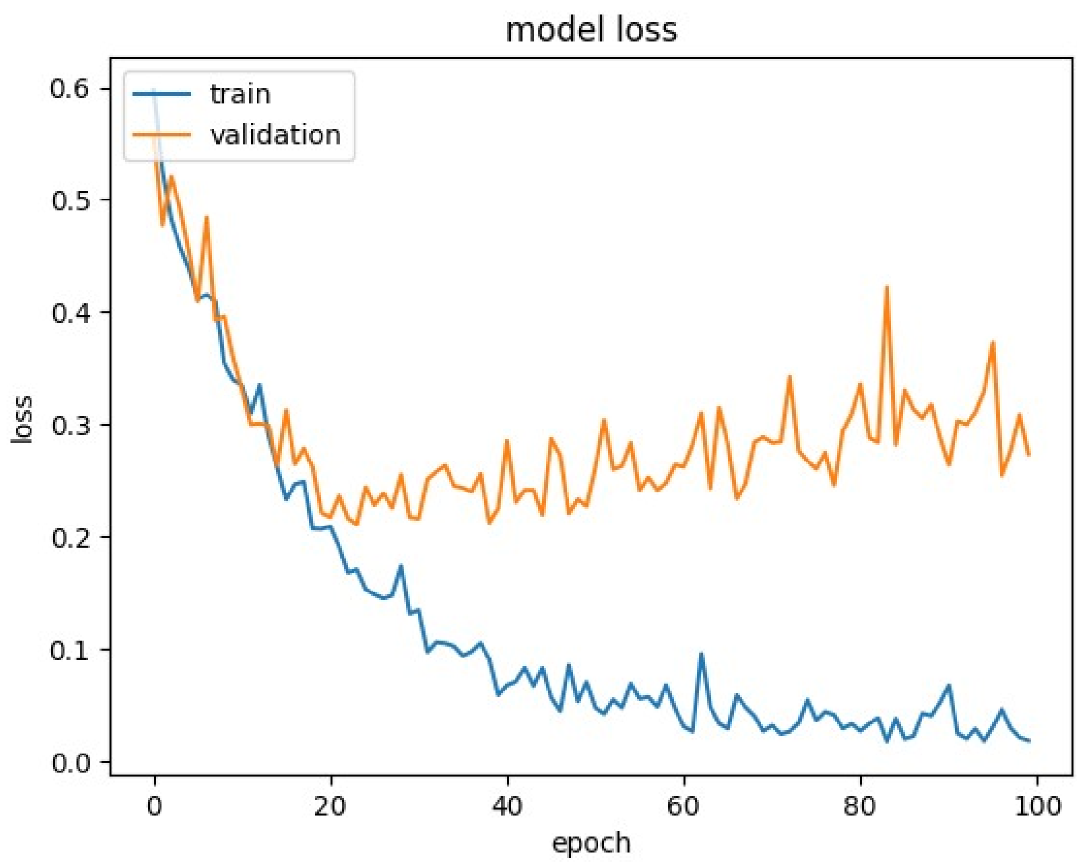
```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3079: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
```

```python
from matplotlib import pyplot as plt
plt.plot(history.history['accuracy']) # since tensorflow 2.x version,
```

```
acc -> accuracy
plt.plot(history.history['val_accuracy']) # val_acc -> val_accuracy
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper left')
plt.show()
```




```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper left')
plt.show()
```




```
import tensorflow as tf
from tensorflow.keras.utils import plot_model
model = tf.keras.models.load_model('TrainedModels/Fire-64x64-color-
v7.1-soft.h5')

# model.fit_generator(datagen.flow(X, Y, batch_size=32),
#                       epochs=100,
#                     verbose=1)

# plot_model(model, to_file='model_small.svg', show_layer_names=False,
show_shapes=True)
```