

```
In [8]: import keras
        from keras.preprocessing.image import ImageDataGenerator
```

```
In [14]: from keras.models import load_model
        from keras.layers import Lambda
        import tensorflow as tf
```

```
In [22]: tf.keras.preprocessing.image_dataset_from_directory(
        directory="C:\\Users\\Akash\\Downloads\\Dataset",
        labels="inferred",
        label_mode="int",
        class_names=None,
        color_mode="rgb",
        batch_size=32,
        image_size=(256, 256),
        shuffle=True,
        seed=None,
        validation_split=None,
        subset=None,
        interpolation="bilinear",
        follow_links=False,
        crop_to_aspect_ratio=False,
        )
```

Found 558 files belonging to 1 classes.

```
Out[22]: <BatchDataset element_spec=(TensorSpec(shape=(None, 256, 256, 3), dtype=tf.float32, name=None), TensorSpec(shape=(None,), dtype=
tf.int32, name=None))>
```

```
In [58]: tf.keras.preprocessing.image.load_img(
        path="C:\\Users\\Akash\\Downloads\\Dataset\\Dataset\\train_set\\forest\\with_fire (1).gif", grayscale=False, color_mode="rgb"
        )
```

```
In [21]: #: Applying ImageDataGenerator functionality to trainset.
        x_train = train_datagen.flow_from_directory(r"C:\\Users\\Akash\\Downloads\\Dataset\\Dataset\\train_set",
        target_size = (128,128),
        batch_size = 32,
        class_mode= 'binary')
```

Found 436 images belonging to 2 classes.

```
In [22]: x_test = test_datagen.flow_from_directory(r"C:\\Users\\Akash\\Downloads\\Dataset\\Dataset\\test_set",
        target_size = (128,128),
        batch_size = 32,
        class_mode= 'binary')
```

Found 121 images belonging to 2 classes.

```
In [23]: from keras.models import Sequential
        from keras.layers import Convolution2D,MaxPooling2D,Dense,Flatten
        import warnings
        warnings.filterwarnings('ignore')
```

```
In [24]: model = Sequential()
        model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
        model.add(MaxPooling2D(pool_size=(2,2)))
        model.add(Flatten())
```