PREPROCESSING

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator


traindata = ImageDataGenerator(
    rescale=1./255,
    zoom_range=0.2,
    horizontal_flip=True,
    rotation_range=260,
    vertical_flip=True,
    fill_mode='reflect',
    validation_split=0.2
)


testdata = ImageDataGenerator(rescale=1./255)


x_train = traindata.flow_from_directory(
    '/content/Dataset/Dataset/train_set',
    target_size = (64,64),
    class_mode ='categorical',
    batch_size = 100,
    shuffle=True
)


x_test = testdata.flow_from_directory(
    "/content/Dataset/Dataset/test_set",
    target_size=(64,64),
    class_mode = 'categorical',
    batch_size = 100,
    shuffle=True
)
```

MODEL BUILDING

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,Dense,MaxPooling2D,Flatten


model = Sequential()


model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())


model.add(Dense(300,activation='relu'))
model.add(Dense(450,activation='relu'))
model.add(Dense(500,activation='relu'))
model.add(Dense(2,activation='softmax'))
```

```
#compile
model.compile(optimizer='Adam',loss="categorical_crossentropy",metrics=['accuracy'
```

```
#train
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=100,
                    validation_data=x_test,
                    validation_steps=len(x_test))
```

```
#save the model
model.save('FFD.h5')
```

```
from keras.models import load_model

model1 = load_model('//content/drive/MyDrive/ibm/FFD.h5')
```

```
import numpy as np
from tensorflow.keras.preprocessing import image
```

```
img1 = image.load_img("/content/Dataset/Dataset/test_set/with fire/Wild_fires.jpg"
```

```
img1
```

```
x = image.img_to_array(img1)
```

```
x = np.expand_dims(x,axis=0)
x
```

```
model1.predict(x)
```

```
x_train.class_indices
```

```
op = ['No fire','fire']
pred = np.argmax(model1.predict(x))
op[pred]
```

VIDEO ANALYSIS

```
import cv2
import numpy as np
```

```
from keras.models import load_model
from twilio.rest import Client
from playsound import playsound
from tensorflow.keras.utils import load_img, img_to_array

model = load_model('FFD.h5')

video = cv2.VideoCapture(0)

name = ["with fire", "forest"]


i = 1
while(i <= 100):
  success,frame = video.read()
  cv2.imwrite("image.jpg",frame)
  img = load_img("image.jpg",target_size = (64, 64))
  x = img_to_array(img)
  x = np.expand_dims(x, axis = 0)
  pred = np.argmax(model.predict(x))
  print(pred)
  P = name[pred]
  print (P)
  cv2.putText(frame,"predicted class = "+P, (100, 100),cv2.FONT_HERSHEY_SIMPLEX, 1

  if pred==0:
    account_sid = 'ACc4260bf733e55db7efe6fa2cf2a570ba'
    auth_token = '3c6e567ba504f0c040dcfc76355af5e9'
    client = Client(account_sid,auth_token)
    message = client.messages \
    .create(
        body = 'Forest Fire is detected, stay alert',
        from_ = '+14793393874',
        to = '+91 78713 35390'
      )
    print(message.sid)
    print('Fire Detected')
    print('SMS sent!')
    playsound(r'C:\Users\Dhinesh\Downloads\Message Alert.mp3')
    break
  else:
    print("No Danger")

  cv2.imshow("image",frame)
  if cv2.waitKey(1) & 0xFF == ord('a'):
        break

video.release()
cv2.destroyAllWindows()
```