

**Final Deliverables**  
**Coding And Solutioning**

Team ID	PNT2022TMID17719
Project Name	Project - Emerging methods for the early detection of forest fires

**CODING & SOLUTIONING**

**Feature 1**

Sequential model is created and pooling layers are added to the model. And model is flatten.

**Coding:**

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,zoom_range=0.2,horizontal_flip=True) test_datagen=ImageDataGenerator(rescale=1./255)
x_train=train_datagen.flow_from_directory(r'C:\Users\USER\Documents\Sem7\Naalaiyathiran\Dataset\Dataset\train_set', target_size=(128,128),
batch_size=32,
class_mode='binary')
x_test=train_datagen.flow_from_directory(r'C:\Users\USER\Documents\Sem7\Naalaiyathiran\Dataset\Dataset\test_set', target_size=(128,128),
```

```

batch_size=32,
class_mode='binary')
x_train.class_indices
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D,MaxPooling2D, Flatten
import warnings
warnings.filterwarnings('ignore')
model=Sequential()
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.summary()

```

## Feature 2

Model is build by adding dense layers with activation function relu and sigmoid. Then the model is compiled with binary cross entropy with adam optimizer and accuracy as performance metric.And then model is trained using training data and predicted using test data.

### Model Building:

```

model.add(Dense(150,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
len(x_train)
len(x_test)
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,
validation_data=x_test,validation_steps=len(x_test))

import tensorflow as tf
from keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
import cv2
model.save('forestfire.h5')
model=load_model('forestfire.h5')
testImg =
image.load_img(r'C:\Users\win\Desktop\Project_NT\test_set\forest\_101542074_g ettyimages_956391468.jpg')
testImgarrayImg = image.img_to_array(testImg)
arrayImg
x = np.expand_dims(arrayImg , axis = 0)X
images = np.vstack([x])

```

```
pred=model.predict(images)
```

```
Pred
```

```
x_train.class_indicesif
```

```
(pred[0] > 0.5):
```

```
print("forest with fire")
```

```
else:
```

```
print("forest without fire")
```

### Feature 3

Video analysis is done on the trained model and alert notification is delivered to the user.

To get the alert notifications we use twilio platform and create account in that platform.

### Video Analysis:

```
import cv2
```

```
import numpy as np
```

```
from keras.preprocessing import image
```

```
from keras.models import load_model
```

```
from twilio.rest import Client
```

```
!pip install twilio
```

```
model=load_model('forestfire.h5')
```

```
video=cv2.VideoCapture(r'C:\Users\win\Desktop\Project_NT\video.mp4')
```

```
name=['forest','with fire']
```

```
#predict=model.predict(x)import
```

```
keras
```

```
from tensorflow.keras.utils import load_img, img_to_array
```

```
while(1):
```

```
success,frame=video.read()
```

```
cv2.imwrite("image.jpg",frame)
```

```
img=keras.utils.load_img("image.jpg")img=
```

```
cv2.resize(frame, (128,128))
```

```
x=keras.utils.img_to_array(img)
```

```
x=np.expand_dims(x,axis=0) dim=(128,128)
```

```
# x=x.reshape(128, 128, 3)
```

```
# x= cv2.resize(x, (128,128)) pred
```

```
= model.predict(x)
```

```
#pred=model.predict_classes(x)
```

```
p=pred[0]
```

```
print(pred)
```

```
# cv2.putText(frame,"predicted
class="+str(name[p]),(100,100),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),1)if
pred[0]==1:
account_sid='AC63518ea
0e5f8e919ee2a4dc4dc17c
db6'
auth_token='e5413a0fd6c
65647ca88e8cb0cd33fac'
client=Client(account_sid,
auth_token)
message=client.messages.create(body='Forest Fire is
detected,stay alert',from_='+1 989 7621639',
to='+91 93394743')
print(message.sid)
print("Fire detected")
print("Sms sent")
else:
print("No danger")
cv2.imshow("image",frame)
video.release()
cv2.destroyAllWindows()
```