Project Development Phase
SPRINT 1

Date	29.10.2022
Team ID	**PNT2022TMID48059**
Project Name	Emerging methods for Early Detection of forest fires

Executable Program

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,
z oom_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\Naalaiyathir
an\Dataset\Dataset\train_set', target_size=(128,128),
                    batch_size=32,
                    class_mode='binar
                    y')
x_test=train_datagen.flow_from_directory(r'C:\Users\USER\Documents\Sem7\Naalaiyathir
a n\Dataset\Dataset\test_set', target_size=(128,128),
                    batch_size=32,
                    class_mode='binar
                    y')
x_train.class_indices                from
tensorflow.keras.models          import
Sequentialfrom tensorflow.keras.layers
import Dense
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,
Flattenimport warnings warnings.filterwarnings('ignore')
model=Sequential()
model.add(MaxPooling2D(pool_size=(2,2))
) model.add(Flatten()) model.summary()
```

In [1]:
```
#import keras library
import keras
#import ImageDataGenerator class from keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [2]:
```
#Define the parameters /arguments for 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)
```

In [3]:
```
#: Applying ImageDataGenerator functionality to trainset.
x_train=train_datagen.flow_from_directory(r'C:\Users\dhine\Downloads\archive\Dataset\Dataset\train_set',
                                          target_size=(128,128),
                                          batch_size=32,
                                          class_mode='binary')
```

```
Found 436 images belonging to 2 classes.
```

In [4]:
```
#: Applying ImageDataGenerator functionality to testset.
x_test=train_datagen.flow_from_directory(r'C:\Users\dhine\Downloads\archive\Dataset\Dataset\test_set',
                                         target_size=(128,128),
                                         batch_size=32,
                                         class_mode='binary')
```

```
Found 121 images belonging to 2 classes.
```

In [5]:
```
x_train.class_indices
```

Out[5]:
```
{'forest': 0, 'with fire': 1}
```

In [58]:
```
#To define Linear inialisation import Sequential
from keras.models import Sequential
#To add Layers import Dense
from keras.layers import Dense
#To create Convolution kernel import Convolution2D
```

In [5]:
```
x_train.class_indices
```

Out[5]:
```
{'forest': 0, 'with fire': 1}
```

In [58]:
```
#To define Linear inialisation import Sequential
from keras.models import Sequential
#To add Layers import Dense
from keras.layers import Dense
#To create Convolution kernel import Convolution2D
from keras.layers import Conv2D
#import Maxpooling layer
from keras.layers import MaxPooling2D
#import  Flatten Layer
from keras.layers import Flatten
from tensorflow.keras import activations
import warnings
warnings.filterwarnings('ignore')
```

In [74]:
```
#intializating the model
model =Sequential()
#add convolutional layer
model.add(Conv2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

model.add(MaxPooling2D())

model.add(Flatten())
model.summary()
```

```
Model: "sequential_25"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_29 (Conv2D)          (None, 126, 126, 32)      896

 max_pooling2d_12 (MaxPoolin  (None, 63, 63, 32)       0
 g2D)

 flatten_10 (Flatten)        (None, 127008)            0

=================================================================
```

```
#import  Flatten layer
from keras.layers import Flatten
from tensorflow.keras import activations
import warnings
warnings.filterwarnings('ignore')
```

In [74]:
```
#intializating the model
model =Sequential()
#add convolutional layer
model.add(Conv2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

model.add(MaxPooling2D())

model.add(Flatten())
model.summary()
```

```
Model: "sequential_25"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_29 (Conv2D)          (None, 126, 126, 32)      896

 max_pooling2d_12 (MaxPoolin  (None, 63, 63, 32)       0
 g2D)

 flatten_10 (Flatten)        (None, 127008)            0

=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:
```
model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

In [68]:
```
#configure the learning process
model.compile(loss='categorical_crossentropy',
              optimizer='adadelta',
              metrics=['accuracy'])
```

```
1
```

In [36]:
```
#Training the model
model.fit_generator(x_train,steps_per_epoch=14,
                    epochs=10,validation_data=x_test,
                    validation_steps=4)
```

```
Epoch 1/10
14/14 [==============================] - 134s 9s/step - loss: 1.1887 - accuracy: 0.6583 - val_loss: 0.2968 - val_accuracy: 0.86
78
Epoch 2/10
14/14 [==============================] - 91s 6s/step - loss: 0.3495 - accuracy: 0.8372 - val_loss: 0.1249 - val_accuracy: 0.966
9
Epoch 3/10
14/14 [==============================] - 72s 5s/step - loss: 0.2420 - accuracy: 0.8876 - val_loss: 0.1213 - val_accuracy: 0.975
2
Epoch 4/10
14/14 [==============================] - 68s 5s/step - loss: 0.1854 - accuracy: 0.9197 - val_loss: 0.1408 - val_accuracy: 0.950
4
Epoch 5/10
14/14 [==============================] - 66s 5s/step - loss: 0.2373 - accuracy: 0.9128 - val_loss: 0.1287 - val_accuracy: 0.950
4
Epoch 6/10
14/14 [==============================] - 67s 5s/step - loss: 0.1814 - accuracy: 0.9266 - val_loss: 0.0541 - val_accuracy: 0.991
7
Epoch 7/10
14/14 [==============================] - 76s 5s/step - loss: 0.1652 - accuracy: 0.9266 - val_loss: 0.0648 - val_accuracy: 0.991
7
Epoch 8/10
14/14 [==============================] - 68s 5s/step - loss: 0.1679 - accuracy: 0.9266 - val_loss: 0.0584 - val_accuracy: 0.991
7
Epoch 9/10
14/14 [==============================] - 66s 5s/step - loss: 0.1498 - accuracy: 0.9358 - val_loss: 0.0517 - val_accuracy: 0.991
7
Epoch 10/10
14/14 [==============================] - 67s 5s/step - loss: 0.1888 - accuracy: 0.9289 - val_loss: 0.0613 - val_accuracy: 1.000
0
```

Out[36]:
```
<keras.callbacks.History at 0x18ef125c1c0>
```

In [38]:
```
#save the model
model.save("forest1.h5")
```

```
7
Epoch 7/10
14/14 [==============================] - 76s 5s/step - loss: 0.1652 - accuracy: 0.9266 - val_loss: 0.0648 - val_accuracy: 0.991
7
Epoch 8/10
14/14 [==============================] - 68s 5s/step - loss: 0.1679 - accuracy: 0.9266 - val_loss: 0.0584 - val_accuracy: 0.991
7
Epoch 9/10
14/14 [==============================] - 66s 5s/step - loss: 0.1498 - accuracy: 0.9358 - val_loss: 0.0517 - val_accuracy: 0.991
7
Epoch 10/10
14/14 [==============================] - 67s 5s/step - loss: 0.1888 - accuracy: 0.9289 - val_loss: 0.0613 - val_accuracy: 1.000
0
```

Out[36]:
```
<keras.callbacks.History at 0x18ef125c1c0>
```

In [38]:
```
#save the model
model.save("forest1.h5")
```

In [39]:
```
#import load_model from keras.mode
from keras.models import load_model
#import image class from keras
from keras.preprocessing import image
#import numpy
import numpy as np
#import cv2
import cv2
```

In [44]:
```
#Load the saved model
model = load_model("forest1.h5")
```

In [46]:
```
#give any random image path
img = image.load_img(r'"C:\Users\dhine\Downloads\archive\Dataset\Dataset\train_set\with fire\with fire (9).jpg"')
x = image.img_to_array(img)
#expand the image shape
x = np.expand_dims(x,axis = 0)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [46], in <cell line: 2>()
      1 #give any random image path
----> 2 img = image.load_img(r'"C:\Users\dhine\Downloads\archive\Dataset\Dataset\train_set\with fire\with fire (9).jpg"')
```