## PROJECT DEVELOPMENT PHASE

## SPRINT 2

|  |  |
|---|---|
| **Date** | **08 November 2022** |
| **Team ID** | **PNT2022TMID50335** |
| **Project Name** | **Emerging Methods For Early Detection Of Forest fire** |
| **Marks** | **8 Marks** |

### Importing The ImageDataGenerator Library

In [ ]:

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

### Define the parameters/arguments for ImageDataGenerator class

In [ ]:

```
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)
```

### Applying ImageDataGenerator functionality to trainset

In [ ]:

```
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/train_set',target_size=(128,128),batch_size=32, class_mode='binary')
```

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

### Applying ImageDataGenerator functionality to testset

In [ ]:

```
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set',target_size=(128,128),batch_size=32, class_mode='binary')
```

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

### Import model building libraries

In [ ]:

```
#To define Linear initialisation 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 Convolution2D
#import Maxpooling layer
from keras.layers import MaxPooling2D
#import flatten layer
from keras.layers import Flatten
```

```
import warnings
warnings.filterwarnings('ignore')
```

### Initializing the model

In [ ]:

```
model=Sequential()
```

### Add CNN Layer

In [ ]:

```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
#add maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
#add flatten layer
model.add(Flatten())
```

### Add Hidden Layer

In [ ]:

```
#add hidden layer
model.add(Dense(150,activation='relu'))
#add output layer
model.add(Dense(1,activation='sigmoid'))
```

### Configure the learning process

In [ ]:

```
model.compile(loss='binary_crossentropy',optimizer="adam",metrics=["accuracy"
])
```

### Train the model

In [ ]:

```
model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_te
st,validation_steps=4)
```

```
Epoch 1/10
14/14 [==============================] - 205s 15s/step - loss: 2.7344 -
accuracy: 0.7454 - val_loss: 0.2016 - val_accuracy: 0.9256
Epoch 2/10
14/14 [==============================] - 20s 1s/step - loss: 0.3267 -
accuracy: 0.8945 - val_loss: 0.2290 - val_accuracy: 0.9339
Epoch 3/10
14/14 [==============================] - 20s 1s/step - loss: 0.2991 -
accuracy: 0.8922 - val_loss: 0.0524 - val_accuracy: 0.9835
Epoch 4/10
14/14 [==============================] - 20s 1s/step - loss: 0.2418 -
accuracy: 0.9174 - val_loss: 0.1570 - val_accuracy: 0.9421
Epoch 5/10
14/14 [==============================] - 20s 1s/step - loss: 0.1984 -
accuracy: 0.9083 - val_loss: 0.0767 - val_accuracy: 0.9752
Epoch 6/10
14/14 [==============================] - 20s 1s/step - loss: 0.1643 -
accuracy: 0.9335 - val_loss: 0.0749 - val_accuracy: 0.9752
Epoch 7/10
14/14 [==============================] - 20s 1s/step - loss: 0.1538 -
accuracy: 0.9312 - val_loss: 0.1264 - val_accuracy: 0.9421
```

```
Epoch 8/10
14/14 [==============================] - 20s 1s/step - loss: 0.1732 -
accuracy: 0.9266 - val_loss: 0.0652 - val_accuracy: 0.9835
Epoch 9/10
14/14 [==============================] - 20s 1s/step - loss: 0.1514 -
accuracy: 0.9358 - val_loss: 0.0567 - val_accuracy: 0.9835
Epoch 10/10
14/14 [==============================] - 20s 1s/step - loss: 0.1445 -
accuracy: 0.9404 - val_loss: 0.0448 - val_accuracy: 0.9917
```

Out[ ]:

### Save The Model

In [ ]:

```
model.save("forest1.h5")
```

### Predictions

In [ ]:

```
#import load_model from keras.model
from keras.models import load_model
#import image class from keras
from tensorflow.keras.preprocessing import image #import numpy
import numpy as np
#import cv2
import cv2
```

In [ ]:

```
#load the saved model
model = load_model("forest1.h5")
```

In [ ]:

```
img=image.load_img(r'/content/drive/MyDrive/Dataset/test_set/forest/0.4800720
0_1530881924_final_forest.jpg')
x=image.img_to_array(img)
res = cv2.resize(x, dsize=(128, 128), interpolation=cv2.INTER_CUBIC)
#expand the image shape
x=np.expand_dims(res,axis=0)
```

In [ ]:

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

```
1/1 [==============================] - 0s 94ms/step
```

In [ ]:

```
pred
```

Out[ ]:

```
array([[0.]], dtype=float32)
```