

Emerging Methods For Early Detection Of Forest Fires

Team ID PNT2022TMID25532

Import Libraries

```
In [1]: from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
from keras.models import Sequential, load_model
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
import matplotlib.pyplot as plt
import numpy as np
```

Image Augmentation

```
In [2]: train_datagen = ImageDataGenerator(
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

```
In [3]: val_datagen = ImageDataGenerator(
    rescale=1./255
)
```

Load Datasets

```
In [4]: train_generator = train_datagen.flow_from_directory(  
        'C:/Users/santh/Videos/ForestFireDetection/Dataset/train_set',  
        target_size=(150, 150),  
        batch_size=16,  
        class_mode='binary'  
    )
```

Found 436 images belonging to 2 classes.

```
In [5]: val_generator = val_datagen.flow_from_directory(  
        'C:/Users/santh/Videos/ForestFireDetection/Dataset/test_set/',  
        target_size=(150, 150),  
        batch_size=16,  
        class_mode='binary'  
    )
```

Found 121 images belonging to 2 classes.

Create Model

```
In [6]: model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=(150, 150, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))
```

Compile the model

```
In [7]: model.compile(
    loss='binary_crossentropy',
    optimizer='rmsprop',
    metrics=['acc']
)
```

Fit the model

```
In [8]: model.fit(
        train_generator,
        epochs=10,
        validation_data=val_generator
    )
```

Epoch 1/10

28/28 [=====] - 33s 1s/step - loss: 0.8366 - acc: 0.7064 - val_loss: 0.1606 - val_acc: 0.9587

Epoch 2/10

28/28 [=====] - 30s 1s/step - loss: 0.3363 - acc: 0.8601 - val_loss: 0.1064 - val_acc: 0.9669

Epoch 3/10

28/28 [=====] - 30s 1s/step - loss: 0.2862 - acc: 0.8716 - val_loss: 0.1042 - val_acc: 0.9669

Epoch 4/10

28/28 [=====] - 30s 1s/step - loss: 0.2339 - acc: 0.9083 - val_loss: 0.2153 - val_acc: 0.9256

Epoch 5/10

28/28 [=====] - 30s 1s/step - loss: 0.2835 - acc: 0.9037 - val_loss: 0.0282 - val_acc: 1.0000

Epoch 6/10

28/28 [=====] - 30s 1s/step - loss: 0.2237 - acc: 0.9128 - val_loss: 0.0233 - val_acc: 1.0000

Epoch 7/10

28/28 [=====] - 25s 883ms/step - loss: 0.2114 - acc: 0.9083 - val_loss: 0.0483 - val_acc: 0.9917

Epoch 8/10

28/28 [=====] - 16s 575ms/step - loss: 0.2124 - acc: 0.9289 - val_loss: 0.0739 - val_acc: 0.9669

Epoch 9/10

28/28 [=====] - 16s 576ms/step - loss: 0.2598 - acc: 0.9128 - val_loss: 0.0339 - val_acc: 1.0000

Epoch 10/10

28/28 [=====] - 15s 536ms/step - loss: 0.1722 - acc: 0.9404 - val_loss: 0.0222 - val_acc: 1.0000

Out[8]: <keras.callbacks.History at 0x1fd33f65d80>

Save the Model

```
In [9]: model.save("ForestDetectionModel.h5")
```

Functions for testing

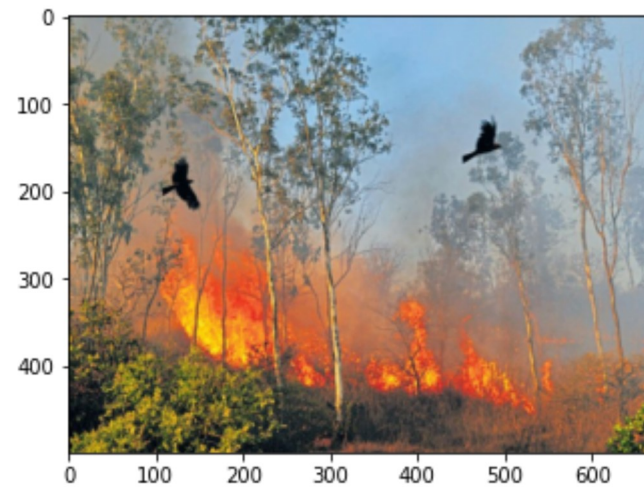
```
In [10]: li = ['Not Fire', 'Fire']  
def detect(pred):  
    print(li[pred])  
  
def show_img(path):  
    img = plt.imread(path)  
    plt.imshow(img)
```

Testing

```
In [11]: model = load_model("ForestDetectionModel.h5")
```

```
In [12]: img_path = "C:/Users/santh/Videos/ForestFireDetection/Dataset/test_set/with fire/599857.jpg"
img = image.load_img(img_path,target_size=(150,150))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = model.predict(x)
detect(pred)
show_img(img_path)
```

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



```
In [13]: img_path = "C:/Users/santh/Videos/ForestFireDetection/Dataset/test_set/forest/_101542074_gettyimages_956391468.jpg"
img = image.load_img(img_path,target_size=(150,150))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = model.predict(x)
detect(pred)
show_img(img_path)
```

1/1 [=====] - 0s 27ms/step
Not Fire

