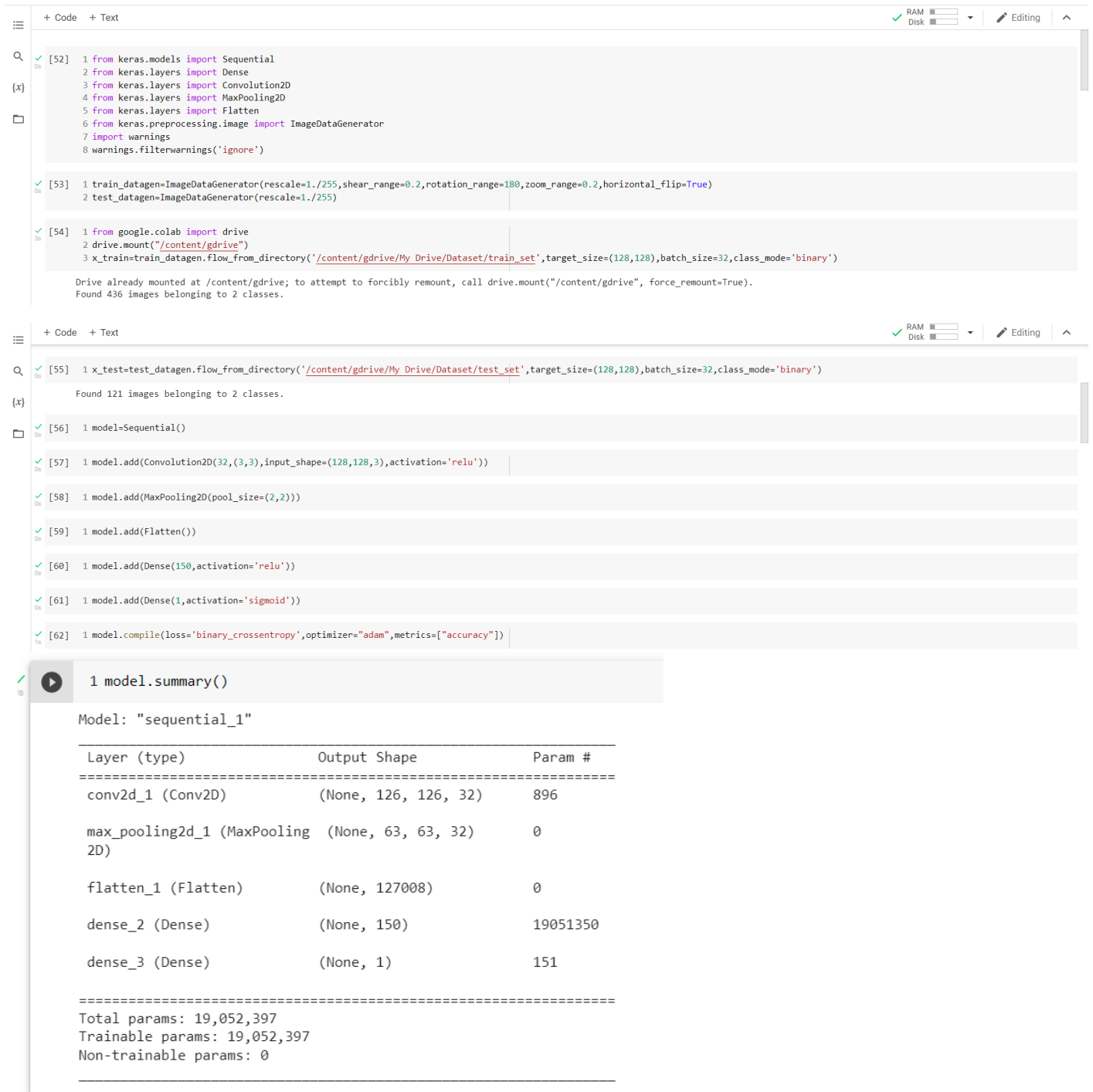


## Project Development Phase SPRINT-1

Date	03 October 2022
Team ID	PNT2022TMID21552
Project Name	EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

In this sprint, we have developed a machine-learning model using CNN to predict forest fires from static images and tested the accuracy of the model.



```
[52] 1 from keras.models import Sequential
2 from keras.layers import Dense
3 from keras.layers import Convolution2D
4 from keras.layers import MaxPooling2D
5 from keras.layers import Flatten
6 from keras.preprocessing.image import ImageDataGenerator
7 import warnings
8 warnings.filterwarnings('ignore')

[53] 1 train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, rotation_range=180, zoom_range=0.2, horizontal_flip=True)
2 test_datagen=ImageDataGenerator(rescale=1./255)

[54] 1 from google.colab import drive
2 drive.mount("/content/gdrive")
3 x_train=train_datagen.flow_from_directory('/content/gdrive/My Drive/Dataset/train_set', target_size=(128,128), batch_size=32, class_mode='binary')

Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
Found 436 images belonging to 2 classes.

[55] 1 x_test=test_datagen.flow_from_directory('/content/gdrive/My Drive/Dataset/test_set', target_size=(128,128), batch_size=32, class_mode='binary')

Found 121 images belonging to 2 classes.

[56] 1 model=Sequential()

[57] 1 model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

[58] 1 model.add(MaxPooling2D(pool_size=(2,2)))

[59] 1 model.add(Flatten())

[60] 1 model.add(Dense(150,activation='relu'))

[61] 1 model.add(Dense(1,activation='sigmoid'))

[62] 1 model.compile(loss='binary_crossentropy',optimizer='adam',metrics=["accuracy"])

1 model.summary()
```

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d_1 (MaxPooling 2D)	(None, 63, 63, 32)	0
flatten_1 (Flatten)	(None, 127008)	0
dense_2 (Dense)	(None, 150)	19051350
dense_3 (Dense)	(None, 1)	151

=====  
Total params: 19,052,397  
Trainable params: 19,052,397  
Non-trainable params: 0

```

✓ [63] 1 y=model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_test,validation_steps=4)

Epoch 1/10
14/14 [=====] - 29s 2s/step - loss: 1.7624 - accuracy: 0.7867 - val_loss: 0.1010 - val_accuracy: 0.9421
Epoch 2/10
14/14 [=====] - 26s 2s/step - loss: 0.2860 - accuracy: 0.8739 - val_loss: 0.1861 - val_accuracy: 0.9256
Epoch 3/10
14/14 [=====] - 25s 2s/step - loss: 0.3511 - accuracy: 0.8601 - val_loss: 0.0831 - val_accuracy: 0.9669
Epoch 4/10
14/14 [=====] - 27s 2s/step - loss: 0.1878 - accuracy: 0.9197 - val_loss: 0.0596 - val_accuracy: 0.9835
Epoch 5/10
14/14 [=====] - 27s 2s/step - loss: 0.1704 - accuracy: 0.9381 - val_loss: 0.0746 - val_accuracy: 0.9752
Epoch 6/10
14/14 [=====] - 26s 2s/step - loss: 0.1592 - accuracy: 0.9312 - val_loss: 0.0791 - val_accuracy: 0.9752
Epoch 7/10
14/14 [=====] - 27s 2s/step - loss: 0.1689 - accuracy: 0.9243 - val_loss: 0.0508 - val_accuracy: 0.9835
Epoch 8/10
14/14 [=====] - 26s 2s/step - loss: 0.1581 - accuracy: 0.9358 - val_loss: 0.0623 - val_accuracy: 0.9669
Epoch 9/10
9/14 [=====>.....] - ETA: 7s - loss: 0.1517 - accuracy: 0.9384

```

```

✓ [65] 1 model.save("forests.h5")

```

```

✓ [66] 1 from keras.models import load_model
2 import cv2
3 import numpy as np
4 from PIL import Image
5 from keras.utils import img_to_array
6 import matplotlib.pyplot as plt
7 model = load_model('forests.h5')
8 def prediction(img_path):
9     i = cv2.imread(img_path)
10    i = cv2.cvtColor(i, cv2.COLOR_BGR2RGB)
11    img = Image.open(img_path)
12    img = img.resize((128,128))
13    x = img_to_array(img)
14    x = np.expand_dims(x,axis=0)
15    pred = model.predict(x)
16    plt.imshow(i)
17    print("%s"%("FOREST FIRE DETECTED!" if pred==[[1.]] else "NO FOREST FIRE DETECTED"))

```

```

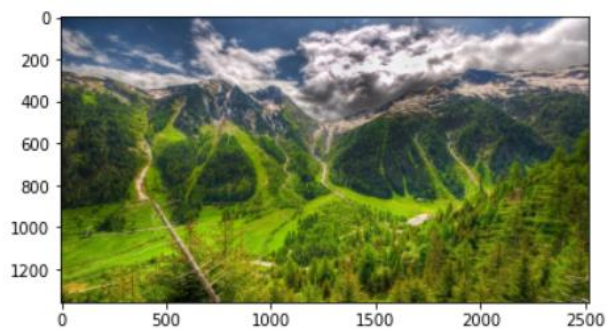
✓ [67] 1 prediction(r'/content/gdrive/My Drive/Dataset/test_set/forest/146019.jpg')

```

```

1/1 [=====] - 0s 70ms/step
NO FOREST FIRE DETECTED

```



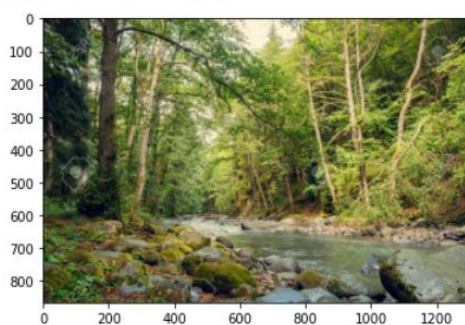
```
1 prediction(r'/content/gdrive/My Drive/Dataset/test_set/forest/1170x500_Ireland_web.jpg')
```

```
1/1 [=====] - 0s 31ms/step  
NO FOREST FIRE DETECTED
```



```
[69] 1 prediction(r'/content/gdrive/My Drive/Dataset/test_set/forest/111188170_river_in_the_mountain_forest.jpg')
```

```
1/1 [=====] - 0s 46ms/step  
NO FOREST FIRE DETECTED
```



```
[70] 1 prediction(r'/content/gdrive/My Drive/Dataset/test_set/with fire/599857.jpg')
```

```
1/1 [=====] - 0s 30ms/step  
FOREST FIRE DETECTED!
```



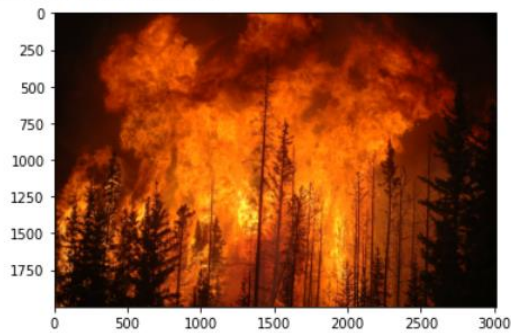
```
[71] 1 prediction(r'/content/gdrive/My Drive/Dataset/test_set/with fire/RED7_May29_1986.jpg')
```

1/1 [=====] - 0s 31ms/step  
FOREST FIRE DETECTED!



```
[72] 1 prediction(r'/content/gdrive/My Drive/Dataset/test_set/with fire/X80IR0GYSsG6RailcsXX.jpg')
```

1/1 [=====] - 0s 32ms/step  
FOREST FIRE DETECTED!



```
1 epochs = range(0,10)
2 plt.plot(epochs, y.history['accuracy'], label='Training Accuracy')
3 plt.plot(epochs, y.history['val_accuracy'], label='Validation Accuracy')
4 plt.legend()
5 plt.title('Training and Validation Accuracy')
6 plt.show()
```

