

{	2
"nbformat": 4,	3
"nbformat_minor": 0,	4
"metadata": {	5
"colab": {	6
"provenance": []	7
},	8
"kernel_spec": {	9
"name": "python3",	10
"display_name": "Python 3"	11
},	12
"language_info": {	13
"name": "python"	14
}	15

```
},
16
"cells": [
17
{
18
"cell_type": "code",
19
"execution_count": 1,
20
"metadata": {
21
"id": "M9TDYjHFMfBL"
22
},
23
"outputs": [],
24
"source": [
25
"import keras\n",
26
"from keras.preprocessing.image import ImageDataGenerator"
27
]
28
},
29
```

```
{  
  
    "cell_type": "code",  
  
    "source": [  
  
        "#Define the parameters/arguments for ImageDataGenerator class\n",  
  
        "train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, rotation_range=180, zoom_range=0.2, horizontal_flip=True)\n",  
  
        "\n",  
  
        "test_datagen=ImageDataGenerator(rescale=1./255)"  
  
    ],  
  
    "metadata": {  
  
        "id": "KEFqxyqwMqMu"  
  
    },  
  
    "execution_count": 2,  
  
    "outputs": []  
  
},
```

```
43
{
44
"cell_type": "code",
45
"source": [
46
"#Applying ImageDataGenerator functionality to trainset\n",
47
"x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/Dataset/Dataset/train_set',target_size=(128,128),batch_
size=32,class_mode='binary')"
48
],
49
"metadata": {
50
"colab": {
51
"base_uri": "https://localhost:8080/"
52
},
53
"id": "D5S4LCT-Mu2L",
54
"outputId": "a4113be0-c0b3-4a95-bf3e-ea77de8b744d"
55
},
56
```

```
"execution_count": 4,
57
"outputs": [
58
{
59
"output_type": "stream",
60
"name": "stdout",
61
"text": [
62
"Found 436 images belonging to 2 classes.\n"
63
]
64
}
65
]
66
},
67
{
68
"cell_type": "code",
69
"source": [
70
```

```
"#Applying ImageDataGenerator functionality to testset\n",
```

71

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

72

```
],
```

73

```
"metadata": {
```

74

```
"colab": {
```

75

```
"base_uri": "https://localhost:8080/"
```

76

```
},
```

77

```
"id": "rQakYllkMxQG",
```

78

```
"outputId": "5d4f7c20-34c2-42ca-e0d8-a8fbfedf01c4"
```

79

```
},
```

80

```
"execution_count": 5,
```

81

```
"outputs": [
```

82

```
{
```

83

```
"output_type": "stream",
```

	84
"name": "stdout",	
	85
"text": [	
	86
"Found 121 images belonging to 2 classes.\n"	
	87
]	
	88
}	
	89
]	
	90
},	
	91
{	
	92
"cell_type": "code",	
	93
"source": [	
	94
"#import model building libraries\n",	
	95
"\n",	
	96
"#To define Linear initialisation import Sequential\n",	
	97
"from keras.models import Sequential\n",	

	98
"#To add layers import Dense\n",	
	99
"from keras.layers import Dense\n",	
	100
"#To create Convolution kernel import Convolution2D\n",	
	101
"from keras.layers import Convolution2D\n",	
	102
"#import Maxpooling layer\n",	
	103
"from keras.layers import MaxPooling2D\n",	
	104
"#import flatten layer\n",	
	105
"from keras.layers import Flatten\n",	
	106
"import warnings\n",	
	107
"warnings.filterwarnings('ignore')"	
	108
],	
	109
"metadata": {	
	110
"id": "tbh9PKR1M9H8"	
	111
},	



	112
"execution_count": 6,	
	113
"outputs": []	
	114
},	
	115
{	
	116
"cell_type": "code",	
	117
"source": [	
	118
"#initializing the model\n",	
	119
"model=Sequential()"	
	120
],	
	121
"metadata": {	
	122
"id": "vINVG-FhM_gF"	
	123
},	
	124
"execution_count": 7,	
	125
"outputs": []	

```
126
},
127
{
128
"cell_type": "code",
129
"source": [
130
"#add convolutional layer\n",
131
"model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))\n",
132
"#add maxpooling layer\n",
133
"model.add(MaxPooling2D(pool_size=(2,2)))\n",
134
"#add flatten layer \n",
135
"model.add(Flatten())"
136
],
137
"metadata": {
138
"id": "UI-pbzGNNCQ5"
139
},
```

```
140
"execution_count": 8,
141
"outputs": []
142
},
143
{
144
"cell_type": "code",
145
"source": [
146
"#add hidden layer\n",
147
"model.add(Dense(150,activation='relu'))\n",
148
"#add output layer\n",
149
"model.add(Dense(1,activation='sigmoid'))"
150
],
151
"metadata": {
152
"id": "CHkc6tNmND_Q"
153
},
```

```
154
"execution_count": 10,
155
"outputs": []
156
},
157
{
158
"cell_type": "code",
159
"source": [
160
"#configure the learning process\n",
161
"model.compile(loss='binary_crossentropy',optimizer=\"adam\",metrics=[\"accuracy\"])"
162
],
163
"metadata": {
164
"id": "T71QZpAVNH7P"
165
},
166
"execution_count": 11,
167
"outputs": []
```

```
168
},
169
{
170
  "cell_type": "code",
171
  "source": [
172
    "#Training the model\n",
173
    "model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_test,validation_steps=4)"
174
  ],
175
  "metadata": {
176
    "colab": {
177
      "base_uri": "https://localhost:8080/"
178
    },
179
    "id": "aTbX1lgCN8Md",
180
    "outputId": "68023839-025c-4295-d02b-88409de396ba"
181
  },

```

```
182
"execution_count": 12,
183
"outputs": [
184
{
185
"output_type": "stream",
186
"name": "stdout",
187
"text": [
188
"Epoch 1/10\n",
189
"14/14 [=====] - 153s 11s/step - loss: 0.6812 - accuracy: 0.6399 - val_loss: 0.6765 - val_accuracy:
0.5950\n",
190
"Epoch 2/10\n",
191
"14/14 [=====] - 26s 2s/step - loss: 0.6577 - accuracy: 0.6445 - val_loss: 0.6765 - val_accuracy:
0.5950\n",
192
"Epoch 3/10\n",
193
"14/14 [=====] - 25s 2s/step - loss: 0.6532 - accuracy: 0.6445 - val_loss: 0.6820 - val_accuracy:
0.5950\n",
194
"Epoch 4/10\n",
```

195  
"14/14 [=====] - 26s 2s/step - loss: 0.6512 - accuracy: 0.6445 - val\_loss: 0.6794 - val\_accuracy:  
0.5950\n",

196  
"Epoch 5/10\n",

197  
"14/14 [=====] - 25s 2s/step - loss: 0.6510 - accuracy: 0.6445 - val\_loss: 0.6793 - val\_accuracy:  
0.5950\n",

198  
"Epoch 6/10\n",

199  
"14/14 [=====] - 25s 2s/step - loss: 0.6509 - accuracy: 0.6445 - val\_loss: 0.6806 - val\_accuracy:  
0.5950\n",

200  
"Epoch 7/10\n",

201  
"14/14 [=====] - 26s 2s/step - loss: 0.6509 - accuracy: 0.6445 - val\_loss: 0.6807 - val\_accuracy:  
0.5950\n",

202  
"Epoch 8/10\n",

203  
"14/14 [=====] - 25s 2s/step - loss: 0.6513 - accuracy: 0.6445 - val\_loss: 0.6815 - val\_accuracy:  
0.5950\n",

204  
"Epoch 9/10\n",

205  
"14/14 [=====] - 25s 2s/step - loss: 0.6511 - accuracy: 0.6445 - val\_loss: 0.6797 - val\_accuracy:  
0.5950\n",

206  
"Epoch 10/10\n",

208

209

210

211

212

213

214

215

216

217

218

219

220

}



]	221
,	222
{	223
"cell_type": "code",	224
"source": [	225
"model.save(\"forest1.h5\")"	226
],	227
"metadata": {	228
"id": "XijgvH8lQ9Qy"	229
,	230
"execution_count": 13,	231
"outputs": []	232
,	233
{	234

"cell_type": "code",	235
"source": [	
	236
"#import load_model from keras.model\n",	
	237
"from keras.models import load_model\n",	
	238
"#import image class from keras\n",	
	239
"from tensorflow.keras.preprocessing import image\n",	
	240
"#import numpy\n",	
	241
"import numpy as np\n",	
	242
"#import cv2\n",	
	243
"import cv2"	
	244
],	
	245
"metadata": {	
	246
"id": "IBDMxiLGQ9xY"	
	247
},	
	248

"execution_count": 14,	249
"outputs": []	250
,	251
{	252
"cell_type": "code",	253
"source": [	254
"#load the saved model\n",	255
"model = load_model(\"forest1.h5\")"	256
],	257
"metadata": {	258
"id": "Q6ywg20GRPF0"	259
},	260
"execution_count": 15,	261
"outputs": []	262

```
},
263

{
264

"cell_type": "code",
265

"source": [
266

"img=image.load_img('/content/drive/MyDrive/Dataset/Dataset/test_set/with
fire/180802_CarrFire_010_large_700x467.jpg')\n",
267

"x=image.img_to_array(img)\n",
268

"res = cv2.resize(x, dsize=(128, 128), interpolation=cv2.INTER_CUBIC)\n",
269

"#expand the image shape\n",
270

"x=np.expand_dims(res,axis=0)"
271

],
272

"metadata": {
273

"id": "_MiXkHBGRS5H"
274

},
275

"execution_count": 16,
```

```
276
"outputs": []
277
},
278
{
279
"cell_type": "code",
280
"source": [
281
"pred=model.predict(x)"
282
],
283
"metadata": {
284
"colab": {
285
"base_uri": "https://localhost:8080/"
286
},
287
"id": "V0O2R4NURmXQ",
288
"outputId": "b8065054-62dd-4d98-809d-3a3b86695914"
289
},
```

290

291

292

293

294

295

296

297

298

299

300

301

302

303

```
"pred"
304

305

],
306

"metadata": {
307

"colab": {
308

"base_uri": "https://localhost:8080/"
309

},
310

"id": "wO031JLbRpTb",
311

"outputId": "b002a3a3-800f-4cb4-f489-74357874e068"
312

},
313

"execution_count": 18,
314

"outputs": [
315

{
316

"output_type": "execute_result",
317

"data": {
```

	318
"text/plain": [	
	319
"array([[0.35252538]], dtype=float32)"	
	320
]	
	321
},	
	322
"metadata": {},	
	323
"execution_count": 18	
	324
}	
	325
]	
	326
},	
	327
{	
	328
"cell_type": "code",	
	329
"source": [],	
	330
"metadata": {	
	331
"id": "IcgNPJNbRros"	



```
},
```

```
"execution_count": null,
```

```
"outputs": []
```

```
}
```

```
]
```

```
}
```

332

333

334

335

336

337