```json
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
```
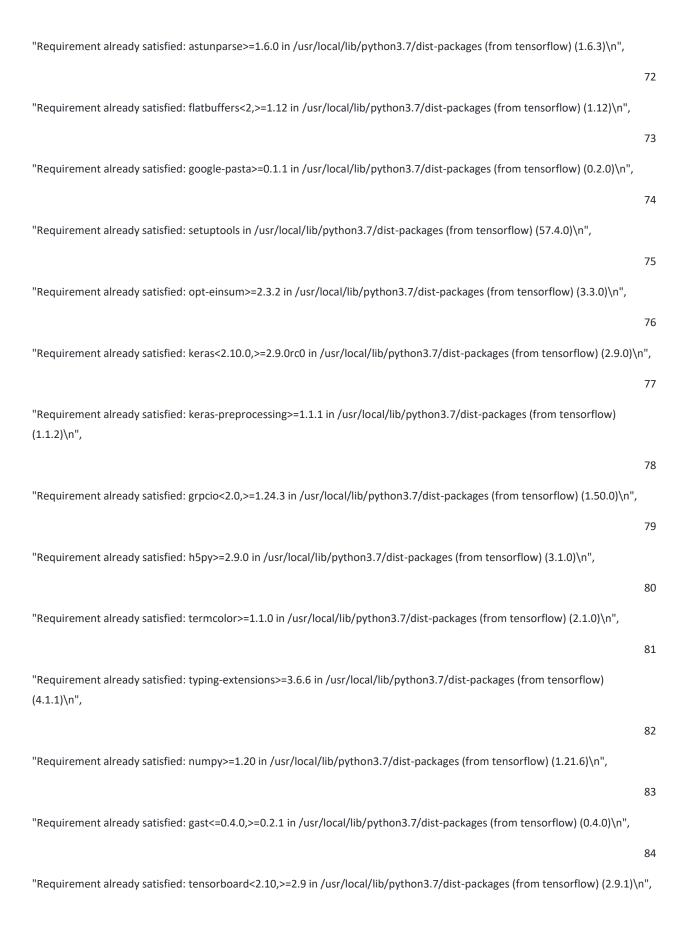
```json
        },
16
    "cells": [
17
        {
18
            "cell_type": "code",
19
            "execution_count": 1,
20
            "metadata": {
21
                "colab": {
22
                    "base_uri": "https://localhost:8080/"
23
                },
24
                "id": "VnfDHSL-jhQX",
25
                "outputId": "c6d7f8ac-6855-47e5-8cdf-c3107c48eb4d"
26
            },
27
            "outputs": [
28
                {
29
```

```
"output_type": "stream",

"name": "stdout",

"text": [

"Mounted at /content/drive\n"

]

}

],

"source": [

"from google.colab import drive\n",

"drive.mount('/content/drive')"

]

},

{

"cell_type": "code",
```

"source": [

"!pip install tensorflow\n",

"!pip install opencv-python\n",

"!pip install opencv-contrib-python\n",

"import tensorflow as tf\n",

"import numpy as np\n",

"from tensorflow import keras\n",

"import os\n",

"import cv2\n",

"from tensorflow.keras.preprocessing.image import ImageDataGenerator\n",

"from tensorflow.keras.preprocessing import image"

],

"metadata": {

"colab": {

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

},

"id": "uwei0k6EjzxD",

"outputId": "cb49595d-e084-4289-9981-0de67da1c1ec"

},

"execution_count": 3,

"outputs": [

{

"output_type": "stream",

"name": "stdout",

"text": [

"Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/\n",

"Requirement already satisfied: tensorflow in /usr/local/lib/python3.7/dist-packages (2.9.2)\n",

"Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.14.1)\n",

"Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.6.3)\n",

72

"Requirement already satisfied: flatbuffers<2,>=1.12 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.12)\n",

73

"Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.2.0)\n",

74

"Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from tensorflow) (57.4.0)\n",

75

"Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.3.0)\n",

76

"Requirement already satisfied: keras<2.10.0,>=2.9.0rc0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.9.0)\n",

77

"Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.1.2)\n",

78

"Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.50.0)\n",

79

"Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.1.0)\n",

80

"Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.1.0)\n",

81

"Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (4.1.1)\n",

82

"Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.21.6)\n",

83

"Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.4.0)\n",

84

"Requirement already satisfied: tensorboard<2.10,>=2.9 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.9.1)\n",

"Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from tensorflow) (21.3)\n",

"Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (14.0.6)\n",

"Requirement already satisfied: protobuf<3.20,>=3.9.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.17.3)\n",

"Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.27.0)\n",

"Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.15.0)\n",

"Requirement already satisfied: tensorflow-estimator<2.10.0,>=2.9.0rc0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.9.0)\n",

"Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.3.0)\n",

"Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.7/dist-packages (from astunparse>=1.6.0->tensorflow) (0.38.1)\n",

"Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-packages (from h5py>=2.9.0->tensorflow) (1.5.2)\n",

"Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.10,>=2.9->tensorflow) (1.35.0)\n",

"Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.10,>=2.9->tensorflow) (2.23.0)\n",

"Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.10,>=2.9->tensorflow) (0.6.1)\n",

"Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.10,>=2.9->tensorflow) (1.0.1)\n",

"Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.10,>=2.9->tensorflow) (3.4.1)\n",

"Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.10,>=2.9->tensorflow) (1.8.1)\n",

"Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.10,>=2.9->tensorflow) (0.4.6)\n",

"Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (0.2.8)\n",

"Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (4.2.4)\n",

"Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (4.9)\n",

"Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow) (1.3.1)\n",

"Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist-packages (from markdown>=2.6.8->tensorboard<2.10,>=2.9->tensorflow) (4.13.0)\n",

"Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata>=4.4->markdown>=2.6.8->tensorboard<2.10,>=2.9->tensorflow) (3.10.0)\n",

"Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (0.4.8)\n",

"Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (1.24.3)\n",

"Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (2.10)\n",

"Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (3.0.4)\n",

"Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (2022.9.24)\n",

"Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow) (3.2.2)\n",

"Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->tensorflow) (3.0.9)\n",

"Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/\n",

"Requirement already satisfied: opencv-python in /usr/local/lib/python3.7/dist-packages (4.6.0.66)\n",

"Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-packages (from opencv-python) (1.21.6)\n",

"Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/\n",

"Requirement already satisfied: opencv-contrib-python in /usr/local/lib/python3.7/dist-packages (4.6.0.66)\n",

119

"Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-packages (from opencv-contrib-python) (1.21.6)\n"

120

]

121

}

122

]

123

},

124

{

125

"cell_type": "code",

126

"source": [

127

"train=ImageDataGenerator(rescale=1./255,\n",

128

" shear_range=0.2,\n",

129

" rotation_range=180,\n",

130

" zoom_range=0.2,\n",

131

" horizontal_flip=True)\n",

132

"train = ImageDataGenerator(rescale=1/255)\n",

133

"test = ImageDataGenerator(rescale=1/255)"

134

],

135

"metadata": {

136

"id": "4mOdBR1BkCOZ"

137

},

138

"execution_count": 4,

139

"outputs": []

140

},

141

{

142

"cell_type": "code",

143

"source": [

144

"train_dataset = train.flow_from_directory(\"/content/drive/MyDrive/Dataset/Dataset/train_set\",\n",

145

" target_size=(128,128),\n",

" batch_size = 32,\n",

" class_mode = 'binary')"

],

"metadata": {

"colab": {

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

},

"id": "Mv7vksB9kIYK",

"outputId": "e216fc79-b711-46a0-8ff0-3b8fc39b77e9"

},

"execution_count": 5,

"outputs": [

{

"output_type": "stream",

160

"name": "stdout",

161

"text": [

162

"Found 436 images belonging to 2 classes.\n"

163

]

164

}

165

]

166

},

167

{

168

"cell_type": "code",

169

"source": [

170

"test_dataset = test.flow_from_directory(\"/content/drive/MyDrive/Dataset/Dataset/test_set\",\n",

171

" target_size=(128,128),\n",

172

" batch_size = 32,\n",

173

" class_mode = 'binary')"

174

```
  ],
```

175

```
  "metadata": {
```

176

```
    "colab": {
```

177

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

178

```
    },
```

179

```
    "id": "r3gWVNLgkM4F",
```

180

```
    "outputId": "4b24776a-84af-40be-fe9a-f6b772cb1329"
```

181

```
  },
```

182

```
  "execution_count": 6,
```

183

```
  "outputs": [
```

184

```
    {
```

185

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

186

```
      "name": "stdout",
```

187

```
      "text": [
```

188

"Found 121 images belonging to 2 classes.\n"

189

]

190

}

191

]

192

},

193

{

194

"cell_type": "code",

195

"source": [

196

"test_dataset.class_indices"

197

],

198

"metadata": {

199

"colab": {

200

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

201

},

"id": "K7kRkC4YkPgI",

"outputId": "929549b5-fcf8-44e0-d0c2-8ecd415b6ae6"

},

"execution_count": 7,

"outputs": [

{

"output_type": "execute_result",

"data": {

"text/plain": [

"{'forest': 0, 'with fire': 1}"

]

},

"metadata": {},

"execution_count": 7

```
}

]

},

{

"cell_type": "code",

"source": [

"#to define linear initialisation import sequential\n",

"from keras.models import Sequential\n",

"#to add layer import Dense\n",

"from keras.layers import Dense\n",

"#to create convolution kernel import convolution2D\n",

"from keras.layers import Convolution2D\n",

"#import Maxpooling layer\n",

"from keras.layers import MaxPooling2D\n",
```

"#import flatten layer\n",

"from keras.layers import Flatten\n",

"import warnings\n",

"warnings.filterwarnings('ignore')"

],

"metadata": {

"id": "F9eyEUqKkRBR"

},

"execution_count": 8,

"outputs": []

},

{

"cell_type": "code",

"source": [

244

"model = keras.Sequential()\n",

245

"model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))\n",

246

"model.add(MaxPooling2D(pool_size=(2,2)))\n",

247

"model.add(Convolution2D(32,(3,3),activation='relu'))\n",

248

"model.add(MaxPooling2D(pool_size=(2,2)))\n",

249

"model.add(Convolution2D(32,(3,3),activation='relu'))\n",

250

"model.add(MaxPooling2D(pool_size=(2,2)))\n",

251

"model.add(Convolution2D(32,(3,3),activation='relu'))\n",

252

"model.add(MaxPooling2D(pool_size=(2,2)))\n",

253

"model.add(Flatten())"

254

],

255

"metadata": {

256

"id": "_fy9tvO0kTWo"

257

},

"execution_count": 9,

"outputs": []

},

{

"cell_type": "code",

"source": [

"model.add(Dense(150,activation='relu'))\n",

"\n",

"model.add(Dense(1,activation='sigmoid'))"

],

"metadata": {

"id": "SBGnl4iMkVAy"

},

"execution_count": 10,

"outputs": []

},

{

"cell_type": "code",

"source": [

"model.compile(loss = 'binary_crossentropy',\n",

" optimizer = \"adam\",\n",

" metrics = [\"accuracy\"])"

],

"metadata": {

"id": "9S22QmvIkXjo"

},

"execution_count": 11,

"outputs": []

```
286        },

287        {

288            "cell_type": "code",

289            "source": [

290                "r = model.fit(train_dataset, epochs = 5, validation_data = test_dataset)"

291            ],

292            "metadata": {

293                "colab": {

294                    "base_uri": "https://localhost:8080/"

295                },

296                "id": "RD4bBxo-kZUb",

297                "outputId": "e2f81de2-5c45-4c76-97fa-f8244101e194"

298            },

299            "execution_count": 12,
```

"outputs": [

{

"output_type": "stream",

"name": "stdout",

"text": [

"Epoch 1/5\n",

"14/14 [==============================] - 92s 7s/step - loss: 0.5798 - accuracy: 0.6628 - val_loss: 0.3330 - val_accuracy: 0.8595\n",

"Epoch 2/5\n",

"14/14 [==============================] - 26s 2s/step - loss: 0.4139 - accuracy: 0.8280 - val_loss: 0.1400 - val_accuracy: 0.9504\n",

"Epoch 3/5\n",

"14/14 [==============================] - 26s 2s/step - loss: 0.2800 - accuracy: 0.8922 - val_loss: 0.1375 - val_accuracy: 0.9587\n",

"Epoch 4/5\n",

"14/14 [==============================] - 27s 2s/step - loss: 0.2440 - accuracy: 0.9014 - val_loss: 0.1224 - val_accuracy: 0.9669\n",

313

"Epoch 5/5\n",

314

"14/14 [==============================] - 26s 2s/step - loss: 0.1856 - accuracy: 0.9243 - val_loss: 0.0586 - val_accuracy: 0.9752\n"

315

]

316

}

317

]

318

},

319

{

320

"cell_type": "code",

321

"source": [

322

"predictions = model.predict(test_dataset)\n",

323

"predictions = np.round(predictions)"

324

],

325

"metadata": {

```
326      "colab": {

327      "base_uri": "https://localhost:8080/"

328      },

329      "id": "UTwKaugikbAM",

330      "outputId": "61103b72-ae2a-4560-b227-92425bde2a34"

331      },

332      "execution_count": 13,

333      "outputs": [

334      {

335      "output_type": "stream",

336      "name": "stdout",

337      "text": [

338      "4/4 [==============================] - 5s 1s/step\n"

339      ]
```

```
}

]

},

{

"cell_type": "code",

"source": [

"predictions"

],

"metadata": {

"colab": {

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

},

"id": "ix0ycxeClr1O",

"outputId": "a8e61ef5-753d-4878-b23a-1e14670dd267"
```

354

        },

355

        "execution_count": 14,

356

        "outputs": [

357

        {

358

        "output_type": "execute_result",

359

        "data": {

360

        "text/plain": [

361

        "array([[0.],\n",

362

        " [1.],\n",

363

        " [1.],\n",

364

        " [0.],\n",

365

        " [1.],\n",

366

        " [0.],\n",

367

        " [1.],\n",

```
    " [0.],\n",
```

```
    " [1.],\n",
```

```
    " [0.],\n",
```

```
    " [1.],\n",
```

```
    " [0.],\n",
```

```
    " [1.],\n",
```

```
    " [1.],\n",
```

```
    " [1.],\n",
```

```
    " [1.],\n",
```

```
    " [0.],\n",
```

```
    " [1.],\n",
```

```
    " [1.],\n",
```

```
    " [0.],\n",
```

```
    " [1.],\n",
```

" [0.],\n",

" [0.],\n",

" [1.],\n",

" [1.],\n",

" [0.],\n",

" [1.],\n",

" [1.],\n",

" [0.],\n",

" [0.],\n",

" [0.],\n",

" [0.],\n",

" [0.],\n",

" [0.],\n",

" [1.],\n",

```
396     " [0.],\n",

397     " [1.],\n",

398     " [0.],\n",

399     " [0.],\n",

400     " [0.],\n",

401     " [0.],\n",

402     " [0.],\n",

403     " [0.],\n",

404     " [0.],\n",

405     " [0.],\n",

406     " [1.],\n",

407     " [0.],\n",

408     " [0.],\n",

409     " [0.],\n",
```

```
410
    " [1.],\n",
411
    " [0.],\n",
412
    " [0.],\n",
413
    " [1.],\n",
414
    " [0.],\n",
415
    " [1.],\n",
416
    " [1.],\n",
417
    " [0.],\n",
418
    " [0.],\n",
419
    " [1.],\n",
420
    " [1.],\n",
421
    " [1.],\n",
422
    " [0.],\n",
423
    " [0.],\n",
```

" [0.],\n",

" [0.],\n",

" [0.],\n",

" [1.],\n",

" [1.],\n",

" [0.],\n",

" [0.],\n",

" [1.],\n",

" [1.],\n",

" [0.],\n",

" [0.],\n",

" [1.],\n",

" [0.],\n",

" [1.],\n",

438

" [1.],\n",

439

" [1.],\n",

440

" [1.],\n",

441

" [0.],\n",

442

" [0.],\n",

443

" [0.],\n",

444

" [0.],\n",

445

" [0.],\n",

446

" [0.],\n",

447

" [0.],\n",

448

" [1.],\n",

449

" [0.],\n",

450

" [0.],\n",

451

" [1.],\n",

```
452    " [0.],\n",

453    " [1.],\n",

454    " [1.],\n",

455    " [1.],\n",

456    " [0.],\n",

457    " [0.],\n",

458    " [1.],\n",

459    " [0.],\n",

460    " [0.],\n",

461    " [1.],\n",

462    " [1.],\n",

463    " [0.],\n",

464    " [0.],\n",

465    " [0.],\n",
```

466

" [0.],\n",

467

" [1.],\n",

468

" [1.],\n",

469

" [1.],\n",

470

" [0.],\n",

471

" [0.],\n",

472

" [1.],\n",

473

" [0.],\n",

474

" [0.],\n",

475

" [1.],\n",

476

" [0.],\n",

477

" [1.],\n",

478

" [1.],\n",

479

" [0.],\n",

```
    " [1.],\n",
    " [0.]], dtype=float32)"
   ]
  },
  "metadata": {},
  "execution_count": 14
  }
 ]
 },
 {
  "cell_type": "code",
  "source": [
   "print(len(predictions))"
  ],
```

```json
"metadata": {

    "colab": {

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

    },

    "id": "bEnOs5ewlvd7",

    "outputId": "f58e5ce5-db13-4b04-c541-0cd7ababdb5e"

},

"execution_count": 15,

"outputs": [

    {

        "output_type": "stream",

        "name": "stdout",

        "text": [

            "121\n"
```

```
      ]

    }

  ]

},

{

  "cell_type": "code",

  "source": [

    "model.save(\"/content/drive/MyDrive/archive (1)/forest1.h5\")"

  ],

  "metadata": {

    "id": "6QLmVFqFlx9Q"

  },

  "execution_count": 17,

  "outputs": []
```

508
509
510
511
512
513
514
515
516
517
518
519
520
521

```
        },

        {

            "cell_type": "code",

            "source": [

                "#import load_model from keras.model\n",

                "from keras.models import load_model\n",

                "#import image class from keras\n",

                "import tensorflow as tf\n",

                "from tensorflow.keras.preprocessing import image\n",

                "#import numpy\n",

                "import numpy as np\n",

                "#import cv2\n",

                "import cv2"

            ],
```

```
"metadata": {

    "id": "Jyc3MylSlztP"

},

"execution_count": 18,

"outputs": []

},

{

"cell_type": "code",

"source": [

"model = load_model(\"/content/drive/MyDrive/archive (1)/forest1.h5\")"

],

"metadata": {

    "id": "C9eUcTWXmIgP"

},
```

536

537

538

539

540

541

542

543

544

545

546

547

548

549

```
      "execution_count": 19,

      "outputs": []
    },
    {
      "cell_type": "code",

      "source": [

        "def predictImage(filename):\n",

        "  img1 = image.load_img(filename,target_size=(128,128))\n",

        "  Y = image.img_to_array(img1)\n",

        "  X = np.expand_dims(Y,axis=0)\n",

        "  val = model.predict(X)\n",

        "  print(val)\n",

        "  if val == 1:\n",

        "    print(\" fire\")\n",
```

550
551
552
553
554
555
556
557
558
559
560
561
562
563

564
```
" elif val == 0:\n",
```

565
```
" print(\"no fire\")"
```

566
```
],
```

567
```
"metadata": {
```

568
```
"id": "lKU2o8qxmLGO"
```

569
```
},
```

570
```
"execution_count": 20,
```

571
```
"outputs": []
```

572
```
},
```

573
```
{
```

574
```
"cell_type": "code",
```

575
```
"source": [
```

576
```
"predictImage(\"/content/drive/MyDrive/Dataset/Dataset/test_set/with fire/19464620_401.jpg\")"
```

577
```
],
```

"metadata": {

"colab": {

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

},

"id": "9Y4J0xcRmMrO",

"outputId": "9b6daec0-186c-4208-e9b6-45287fd84d9d"

},

"execution_count": 26,

"outputs": [

{

"output_type": "stream",

"name": "stdout",

"text": [

"1/1 [==============================] - 0s 125ms/step\n",

"[[1.]]\n",

" fire\n"

]

}

]

},

{

"cell_type": "code",

"source": [

"pip install twilio"

],

"metadata": {

"colab": {

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

        },

        "id": "L5ro8SE3mO86",

        "outputId": "297e1bec-44cb-49a8-dc53-73b15b46cd30"

      },

      "execution_count": 27,

      "outputs": [

        {

          "output_type": "stream",

          "name": "stdout",

          "text": [

            "Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/\n",

            "Collecting twilio\n",

            " Downloading twilio-7.15.1-py2.py3-none-any.whl (1.4 MB)\n",

            "\u001b[K |████████████████████████████████| 1.4 MB 4.9 MB/s \n",

"\u001b[?25hRequirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.23.0)\n",

"Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from twilio) (2022.6)\n",

"Collecting PyJWT<3.0.0,>=2.0.0\n",

" Downloading PyJWT-2.6.0-py3-none-any.whl (20 kB)\n",

"Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)\n",

"Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (3.0.4)\n",

"Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (1.24.3)\n",

"Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2022.9.24)\n",

"Installing collected packages: PyJWT, twilio\n",

"Successfully installed PyJWT-2.6.0 twilio-7.15.1\n"

]

}

```
    ]
  },
  {
    "cell_type": "code",
    "source": [
      "pip install playsound"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "u1GFVreJnKlT",
      "outputId": "d631c790-2ad6-4f4b-96c8-cb09084bddfb"
    },
```

"execution_count": 28,

"outputs": [

{

"output_type": "stream",

"name": "stdout",

"text": [

"Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/\n",

"Collecting playsound\n",

" Downloading playsound-1.3.0.tar.gz (7.7 kB)\n",

"Building wheels for collected packages: playsound\n",

" Building wheel for playsound (setup.py) ... \u001b[?25l\u001b[?25hdone\n",

" Created wheel for playsound: filename=playsound-1.3.0-py3-none-any.whl size=7035 sha256=6cc8a594765dc045811d54129bc5e3fbe95669eecf509234f657cb6a9be4eb0c\n",

" Stored in directory: /root/.cache/pip/wheels/ba/f8/bb/ea57c0146b664dca3a0ada4199b0ecb5f9dfcb7b7e22b65ba2\n",

"Successfully built playsound\n",

```
"Installing collected packages: playsound\n",
```

```
"Successfully installed playsound-1.3.0\n"
```

```
]
```

```
}
```

```
]
```

```
},
```

```
{
```

```
"cell_type": "code",
```

```
"source": [
```

```
"#import opencv librariy\n",
```

```
"import cv2\n",
```

```
"#import numpy\n",
```

```
"import numpy as np\n",
```

```
"#import image function from keras\n",
```

```
    "from keras.preprocessing import image\n",

    "#import load_model from keras\n",

    "from keras.models import load_model\n",

    "#import client from twilio API\n",

    "from twilio.rest import Client\n",

    "#imort playsound package\n",

    "from playsound import playsound"

    ],

    "metadata": {

    "colab": {

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

    },

    "id": "lRYUXaKdnnA_",

    "outputId": "c0e95e37-90e6-4af0-ca72-c65019c4bf19"
```

674

675

676

677

678

679

680

681

682

683

684

685

686

687

},

"execution_count": 29,

"outputs": [

{

"output_type": "stream",

"name": "stderr",

"text": [

"WARNING:playsound:playsound is relying on another python subprocess. Please use `pip install pygobject` if you want playsound to run more efficiently.\n"

]

}

]

},

{

688

689

690

691

692

693

694

695

696

697

698

699

700

701

```json
"cell_type": "code",

"source": [

"#load the saved model\n",

"model = load_model(r'/content/drive/MyDrive/archive (1)/forest1.h5')\n",

"#define video\n",

"video = cv2.VideoCapture('/content/Fighting Fire with Fire _ Explained in 30 Seconds.mp4')\n",

"#define the features\n",

"name = ['forest','with forest']"

],

"metadata": {

"id": "p6I7mDNen2KO"

},

"execution_count": 30,

"outputs": []
```

```
    },
    {
      "cell_type": "code",
      "source": [
        "account_sid = 'ACde2b15dad8f6e39c32b35eaa64921cf2'\n",
        "auth_token = '1928bb64202abc74a3ff94b70d5deec4'\n",
        "client = Client(account_sid, auth_token)\n",
        "\n",
        "message = client.messages \\\n",
        " .create(\n",
        " body='Forest fire is detected , stay alert',\n",
        " from_='+16075363954',\n",
        " to='+919488200286'\n",
        " )\n",
```

```
      "\n",

      "print(message.sid)"

    ],

    "metadata": {

      "colab": {

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

      },

      "id": "C0mzy3T6oBUu",

      "outputId": "79041895-7dda-4771-9d49-f39072dd4bcb"

    },

    "execution_count": 32,

    "outputs": [

      {

        "output_type": "stream",
```

730

731

732

733

734

735

736

737

738

739

740

741

742

743

```
            "name": "stdout",

            "text": [

                "SMcd33e58fa6f60aa349ecba81dce9b48d\n"

            ]

        }

    ]

},

{

    "cell_type": "code",

    "source": [

        "#import opencv library\n",

        "import cv2\n",

        "#import numpy\n",

        "import numpy as np\n",
```

```
"#import images and load_model function from keras\n",
```

```
"from keras_preprocessing import image\n",
```

```
"from keras.models import load_model\n",
```

```
"#import client from twilio API\n",
```

```
"from twilio.rest import Client\n",
```

```
"#import playsound package\n",
```

```
"from playsound import playsound\n",
```

```
"\n",
```

```
"#load the saved model\n",
```

```
"model = load_model(r'/content/drive/MyDrive/archive (1)/forest1.h5')\n",
```

```
"video = cv2.VideoCapture('/content/Fighting Fire with Fire _ Explained in 30 Seconds.mp4')\n",
```

```
"name = ['forest','with fire']"
```

```
],
```

```
"metadata": {
```

"id": "lNprEvNEoPSy"

},

"execution_count": 33,

"outputs": []

},

{

"cell_type": "code",

"source": [],

"metadata": {

"id": "bkCT1z7PqZJE"

},

"execution_count": null,

"outputs": []

}

772

773

774

775

776

777

778

779

780

781

782

783

784

785

]

}