

VIDEO ANALYSIS

TWILIO SERVICE

Date	04 November 2022
Team ID	PNT2022TMID13480
Project Name	Emerging methods for the early detection of forest fires

Code:

```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": [],
      "collapsed_sections": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
```

```

{
  "cell_type": "code",
  "execution_count": 1,
  "metadata": {
    "id": "W8sBRTFDx9vh"
  },
  "outputs": [],
  "source": [
    "import keras\n",
    "from keras.preprocessing.image import ImageDataGenerator "
  ]
},
{
  "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": "7gvCpfDH0MOD"
  },
  "execution_count": 2,

```

```

    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "#Applying ImageDataGenerator functionality to trainset\n",

"x_train=train_datagen.flow_from_directory('/content/Dataset/Dataset/train_set',target_size=(128,128),batch_size=32,class_mode='binary')"

    ],
    "metadata": {
      "id": "zTMlFh9K3XfS",
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "outputId": "63b623a6-ea8d-476f-e035-199a3991d566"
    },
    "execution_count": 4,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "Found 436 images belonging to 2 classes.\n"
        ]
      }
    ]
  }

```

```

    ]
  },
  {
    "cell_type": "code",
    "source": [
      "#Applying ImageDataGenerator functionality to testset\n",

      "x_test=test_datagen.flow_from_directory('/content/Dataset/Dataset/test_set',target
      _size=(128,128),batch_size=32,class_mode='binary')"

    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "FyiVLj49OynB",
      "outputId": "cf37d6ba-5c44-4638-a712-c17fd073ae97"
    },
    "execution_count": 5,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "Found 121 images belonging to 2 classes.\n"
        ]
      }
    ]
  }

```

```

]
},
{
  "cell_type": "code",
  "source": [
    "#import model building libraries\n",
    "\n",
    "#To define Linear initialisation import Sequential\n",
    "from keras.models import Sequential\n",
    "#To add layers 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": "mNBt4boEQ_EO"
  },
  "execution_count": 6,
  "outputs": []
}

```

```

},
{
  "cell_type": "code",
  "source": [
    "#initializing the model\n",
    "model=Sequential()"
  ],
  "metadata": {
    "id": "WOxj9bU9-goY"
  },
  "execution_count": 7,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "#add convolutional layer\n",

"model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))\n",
    "#add maxpooling layer\n",
    "model.add(MaxPooling2D(pool_size=(2,2)))\n",
    "#add flatten layer \n",
    "model.add(Flatten()) "
  ],
  "metadata": {

```

```

    "id": "cADyODoXBAUI"
  },
  "execution_count": 8,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "#add hidden layer\n",
    "model.add(Dense(150,activation='relu'))\n",
    "#add output layer\n",
    "model.add(Dense(1,activation='sigmoid'))"
  ],
  "metadata": {
    "id": "C2mrC6T6Bk8u"
  },
  "execution_count": 9,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "#configure the learning process\n",
    "model.compile(loss='binary_crossentropy',optimizer=\"adam\",metrics=[\"accuracy\"])"]

```

```

],
"metadata": {
  "id": "pa8MgIjFGrVp"
},
"execution_count": 10,
"outputs": []
},
{
  "cell_type": "code",
  "source": [
    "#Training the model\n",

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

  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "-TmR_z4-IGlo",
    "outputId": "776ee68c-1c77-4737-80c9-e86b57b51c37"
  },
  "execution_count": 12,
  "outputs": [
    {
      "output_type": "stream",

```



```
"name": "stdout",
"text": [
  "Epoch 1/10\n",
  "14/14 [=====] - 26s 2s/step - loss:
0.1884 - accuracy: 0.9128 - val_loss: 0.0690 - val_accuracy: 0.9669\n",
  "Epoch 2/10\n",
  "14/14 [=====] - 25s 2s/step - loss:
0.2131 - accuracy: 0.8830 - val_loss: 0.0923 - val_accuracy: 0.9504\n",
  "Epoch 3/10\n",
  "14/14 [=====] - 25s 2s/step - loss:
0.1947 - accuracy: 0.9151 - val_loss: 0.0740 - val_accuracy: 0.9587\n",
  "Epoch 4/10\n",
  "14/14 [=====] - 25s 2s/step - loss:
0.1663 - accuracy: 0.9312 - val_loss: 0.0698 - val_accuracy: 0.9752\n",
  "Epoch 5/10\n",
  "14/14 [=====] - 26s 2s/step - loss:
0.1668 - accuracy: 0.9404 - val_loss: 0.0611 - val_accuracy: 0.9835\n",
  "Epoch 6/10\n",
  "14/14 [=====] - 25s 2s/step - loss:
0.1840 - accuracy: 0.9151 - val_loss: 0.0641 - val_accuracy: 0.9752\n",
  "Epoch 7/10\n",
  "14/14 [=====] - 25s 2s/step - loss:
0.2018 - accuracy: 0.9128 - val_loss: 0.0846 - val_accuracy: 0.9752\n",
  "Epoch 8/10\n",
  "14/14 [=====] - 25s 2s/step - loss:
0.1943 - accuracy: 0.9106 - val_loss: 0.0665 - val_accuracy: 0.9752\n",
  "Epoch 9/10\n",
```

```
    "14/14 [=====] - 25s 2s/step - loss: 0.1984 - accuracy: 0.9151 - val_loss: 0.0715 - val_accuracy: 0.9669\n",
```

```
    "Epoch 10/10\n",
```

```
    "14/14 [=====] - 26s 2s/step - loss: 0.1742 - accuracy: 0.9243 - val_loss: 0.0627 - val_accuracy: 0.9752\n"
```

```
    ]
```

```
  },
```

```
  {
```

```
    "output_type": "execute_result",
```

```
    "data": {
```

```
      "text/plain": [
```

```
        "<keras.callbacks.History at 0x7f04f32388d0>"
```

```
      ]
```

```
    },
```

```
    "metadata": {},
```

```
    "execution_count": 12
```

```
  }
```

```
]
```

```
},
```

```
{
```

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

```
  "source": [
```

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

```
  ],
```

```
  "metadata": {
```

```
    "id": "wy0ybBWnL5Jb"
```

```
},
"execution_count": null,
"outputs": []
},
{
"cell_type": "code",
"source": [
"#import load_model from keras.model\n",
"from keras.models import load_model\n",
"#import image class from keras\n",
"from tensorflow.keras.preprocessing import image\n",
"#import numpy\n",
"import numpy as np\n",
"#import cv2\n",
"import cv2\n",
],
"metadata": {
"id": "d8dYcGPqoEne"
},
"execution_count": null,
"outputs": []
},
{
"cell_type": "code",
"source": [
```

```

"#load the saved model\n",
"model = load_model(\"forest1.h5\")"
],
"metadata": {
  "id": "Zkq9A29zpkml"
},
"execution_count": null,
"outputs": []
},
{
  "cell_type": "code",
  "source": [
    "img=image.load_img('/content/Dataset/Dataset/test_set/with\n",
fire/180802_CarrFire_010_large_700x467.jpg')\n",
    "x=image.img_to_array(img)\n",
    "res = cv2.resize(x, dsize=(128, 128), interpolation=cv2.INTER_CUBIC)\n",
    "#expand the image shape\n",
    "x=np.expand_dims(res,axis=0)"
  ],
  "metadata": {
    "id": "0wPII3sMps3A"
  },
  "execution_count": null,
  "outputs": []
},
{

```

```
"cell_type": "code",
"source": [
  "pred=model.predict(x)"
],
"metadata": {
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "id": "_X7xSZkHr6Ef",
  "outputId": "945c247c-1f53-446a-e05d-c1f24a4134cd"
},
"execution_count": null,
"outputs": [
  {
    "output_type": "stream",
    "name": "stdout",
    "text": [
      "1/1 [=====] - 0s 37ms/step\n"
    ]
  }
],
{
  "cell_type": "code",
  "source": [
```

```
"pred"
],
"metadata": {
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "id": "FqvlyDfrtIMk",
  "outputId": "89dd9d63-9859-436e-8de9-cd09bd6a7ebe"
},
"execution_count": null,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "array([[1.]], dtype=float32)"
      ]
    },
    "metadata": {},
    "execution_count": 72
  }
],
{
  "cell_type": "code",
```

```
"source": [  
    "#import opencv library\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",  
    "#import playsound package\n",  
    "from playsound import playsound"  
],  
"metadata": {  
    "id": "rn0FqT0YVV98"  
},  
"execution_count": null,  
"outputs": []  
},  
{  
    "cell_type": "code",  
    "source": [  
        "#load the saved model\n",  
        "model=load_model('')\n",
```

```
"#define video\n",
"video=cv2.VideoCapture(0)\n",
"#define the features\n",
"name=['forest','with fire']\n"
],
"metadata": {
  "id": "ZdDJ4NOISyUy"
},
"execution_count": null,
"outputs": []
},
{
  "cell_type": "code",
  "source": [
    "#twilio account ssid\n",
    "account_sid='ACcc04dc5c3d6c4d751207bc04285f173c'\n",
    "#twilio account authentication token\n",
    "auth_token ='d618b8b66a30f7579f6cf509c4301a08'\n",
    "client=Client(account_sid,auth_token)\n",
    "\n",
    "message=client.messages \\\n",
    ".create(\n",
    "  body='Forest Fire is detected,stay alert',\n",
    "  #use twilio free number\n",
    "  from_='+19804145862',\n
```



```
"    #to number\n",
"    to='9080590163')\n",
"print(message.sid)\n",
"print('Fire Detected')\n",
"print('SMS sent!')\n",
"playsound('/tornado-siren-in-streamwood-il-35510.mp3')\n"
],
"metadata": {
    "id": "8MFj3cXkbP2z"
},
"execution_count": null,
"outputs": []
},
{
    "cell_type": "code",
    "source": [],
    "metadata": {
        "id": "RD8a1k43Yfml"
    },
    "execution_count": null,
    "outputs": []
}
]
```